CHAPTER 3

INFORMATION VISUALIZATION

In this chapter, a thorough study on Information Visualization is made. Section 3.1 looks at the relationship between Information Visualization and Human Computer Interaction (HCI). Section 3.2 studies the importance and needs for information visualization for web-based applications and web sites. Section 3.3 presents the difference between information visualization and information retrieval as well as presenting the models for both of them. Section 3.4 studies the framework of information visualization. Section 3.5 discusses the examples of Information Visualization interfaces developed by other researchers. Section 3.6 summarizes and synthesizes the discussion of this chapter.

3.1 HCI AND INFORMATION VISUALIZATION

Human Computer Interaction

HCI is defined as the study of how people design, implement, and use interactive computer systems, and how computers affects individuals, organizations, and society [Greenberg, 1992]. HCI is indeed a research area of increasingly central significance to computer science, other scientific and engineering disciplines, and will be expanding in array of application domains such as in the mobile computing, mobile telephony, embedded systems for home appliances and etc.

One of the domain of study under HCI is Information Visualization. In an article entitled Strategic Directions in Human Computer Interaction [Myers et al., 1996], Information Visualization was identified as one of the strategic themes of research by HCI scholars especially on how the techniques can be applied for Internet applications as well as other communication networks.

Information Visualization

Information Visualization is historically originated from the scientific visualization but as time changes it is more prevalent in the HCI discipline. Visualization, according to Shneiderman [1998] is a method of computing. In general, visualization is a way of demonstrating properties or supersets of discrete items, whose very superabundance inhibits or precludes comprehension [Braham, 1995]. No matter what the definition of visualization, it is actually a cognitive process which enables people to use their natural tool of observation and processing that is their eyes and brain, respectively in extracting knowledge more efficiently and to find insights [Gershon, 1995].

Visualizations may be characterized on a continuum ranging from 'physically concrete to purely abstract' depending on the properties of the objects being rendered. Going back to its root in scientific visualization, the scientific one is mainly concerned with phenomena that are based in the physical world. The most concrete visualizations are renderings of objects as they exist in the world. For example, the walk-through of the National Archives of Malaysia' exhibit hall. Renderings of building plans could be placed a little further along the continuum since the building does not already exist but it is made visible based on physical properties contained in the plan made by the designer. Models that attempt to render properties that are not visible, such as forces in bridges beam or wind gusts in weather simulations, are making things that cannot ordinarily be seen visible. Maps are another example or instance of visualizations that are rooted in the physical world; they are used to magnify things like computer chips; or to shrink large objects or phenomena like geographical maps. In addition, they may be used to display attributes that are physical such as land topology or abstract such as population density in a certain location.

People have tremendous perceptual abilities for visual information. Visualizations rely on the fact that users can distinguish positions, colors, textures, and relationships. Relationships can be shown in such displays by proximity, by containment, by connected lines, by color coding, etc. Fields containing hundreds or thousands of points can be scanned rapidly and efficiently for clusters, outliers, trends and gaps [Shneiderman, 1995]. Attention can be drawn to salient items using a variety of techniques including highlighting, blinking, motion, and size. Direct manipulation of visualizations can be accomplished with a variety of methods such as pointing to select, dragging, and zooming. Feedback is immediate and intuitive in such environments. "The eye, the hand and the mind seem to work smoothly and rapidly as users perform actions on visual displays" [Shneiderman, 1995].

As information visualization is making use of the perceptual systems innate capability rather than cognitive system, thus the speed of task execution in human information assimilation can be increased by several orders of magnitude. This is due to the fact that perceptual systems operate in the range of 10 to 100 milliseconds while cognitive systems operate in a longer time range of hundreds of milliseconds to several minutes. Hence, this will make information visualization on the web's navigation is more comforting and enjoying.

3.2 THE IMPORTANCE OF INFORMATION VISUALIZATION

The importance of information visualization will increase as people have access to larger and more diverse sources of information (e.g. digital libraries, virtual archives and large databases), which are becoming universally available with the WWW. Information Visualization also expand the concept of direct manipulation

[Shneiderman, 1998] which shows the power of using computers in a more visual or graphic manner because as often said, a picture is worth thousand words. As a matter of fact, for example, a map is more easier to use and comprehend than is a textual description or a spoken report.

According to Shneiderman, Card and MacKinlay [1995] in their paper, Information Visualization is used because it will increase the assimilation of resources in terms of higher bandwidth of hierarchical interaction. This is due to the parallel perceptual processing by the eyes which offloading the work of understanding (inside the brain) cognitively to the natural perceptual system.

Information visualization is also able to reduce the need for search in terms of locality of processing. What it means here is that the user will always be in context of the site or in a topic when the technique is used. This is because the site's structure is graphically represented in any techniques deem appropriate for that particular site. Thus, browse or search action will not diverted away from the sites content itself [Jiajie, 1999]. He also consider the enhancement of recognition of patterns as they are visually represented instead of forcefully let the users to recall, and do mind abstraction and aggregation. Instead, information visualization allows visual schemata for organization of the site's collections as well as the ability of showing value, relationship and tread.

Information visualization also will provide many features that the current user interfaces do not have [Brautigam, 1996], thus making the needs for such features are very much appreciated and demanded. Among the features are as listed below:

1. a means of easily seeing trends in the data
2. a means of easily seeing outliers
3. a means of seeing jumps in the data (gaps)
4. a means of easily identifying maxima and minima such as largest, smallest, most recent, oldest, etc.
5. a means of identifying boundaries
6. a means of easily identifying clusters in the data
7. a means of finding structure in heterogeneous information
8. a means of seeing an enormous amount of data on one display screen
9. a means of seeing a particular item of interest within the context of an enormous amount of contextual data

3.3 INFORMATION VISUALIZATION VERSUS INFORMATION RETRIEVAL

A very related and relevant topic to Information Visualization is the Information Retrieval topic. Even though this thesis will go more into Information Visualization, there is a temptation to get down into Information Retrieval techniques as well. Information Retrieval actually covers a discussion as big as information visualization.

The retrieval process in the traditional view is quite simple. Information is stored and later retrieved when it is needed. Online retrieval systems typically consist of a large document database. Terms that describe the document contents (also called index) are selected from manual or automatic indexing. The index terms are descriptors of the represented document. Queries are requests to process information and a search query consists of different terms combined in a structured query language. The traditional information retrieval paradigm is a matching process according to the similarity between the keyword index entries and the search query.

The problem is to find all and only the relevant documents. To evaluate the retrieval results, two statistics are used:

1. Recall, that is the percentage of relevant document found
2. Precision, the percentage of the documents found that are relevant

From the evaluation of the retrieval results, one can formulate a new query. The traditional model of information retrieval is shown in Figure 3.1. Problems can arise in this model when the number of retrieved documents is very large or when the language used to specify the query is poorly matched to the real information need of the user.

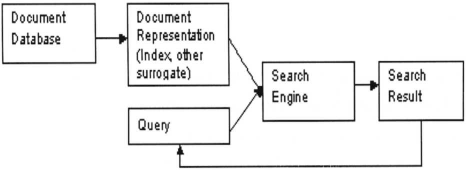

Figure 3.1 Information Retrieval Model

An information visualization as user interface could help to overcome these problems. As shown in Figure 3.2, the abstract data model represented by the index is visualized as an information space.

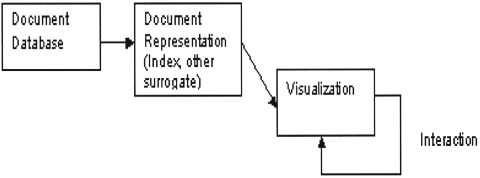

Figure 3.2 Information Visualization Model

The user interacts directly with the visualization to express their needs. The query in this model is stated implicitly in the view and it is filtered and refined through manipulation of the interface. Navigation inside the information space is helpful following a context-oriented search path to find certain domains of interest. Relevance in the visual setting is not necessarily a predetermined characteristics of a document. The interaction of the user with the interface supports browsing, creativity and constant refinement of the original statement of the goal of the search. The refinement of search is where dynamic queries will fits in.

Dynamic queries is a technique that allows user to change the query parameters and see the display update in real time. Dynamic queries continuously update search results within 100 milliseconds as users adjust sliders or tick on checkboxes or select buttons to ask simple questions of fact or to find patterns or exceptions [Shneiderman, 1993]. This technique have proved to be very effective and satisfying [Shneiderman, 1998].

3.4 INFORMATION VISUALIZATION FRAMEWORK

This thesis will take a look into works by Shneiderman [1995, 1998]. Shneiderman called the framework as the Data Type by Task Taxonomy (DTTT) Framework. He has suggested that a data type by interaction type framework could ground work in visualization as a whole. The framework suggested by Shneiderman to support research in visualization is two dimensional. The first dimension is the data-type of the objects to be represented in the interface. He lists seven types [Shneiderman, 1998] found in his textbook. The types are linear, planar, volumetric, temporal, multidimensional, tree, and network. The second dimension is a task typology and includes: overview, zoom, filter, details-on-demand, relate, history, and extract. The scope of the Shneiderman's DTTT Framework is the entire domain of visualization. Table 3.1 provides a graphical view of the framework.

The discussion on the framework introduced is covered in the following sections.

Table 3.1 DTTT Framework (Data Type by Task Taxonomy)

Data Types
Linear
2-D
3-D
Multidimensional
Temporal
Hierarchical Tree
Network
Interface Functionality
Overview
Zoom
Filter
Details-on-demand
Relate
History
Extract

Source: Shneiderman, Ben. Designing the User Interface: Strategies for Effective Human Computer Interaction, 3rd Edition. Addison-Wesley, 1998.

3.4.1 Data Types

Linear

Viewing documents as streams of words has a similarity to accessing speech. Techniques that are applied to spoken words can be adapted to analyzing written words. Speech is examined at several levels, including phonology, morphology, syntax and semantics. A contextual method that has been applied to written text is discourse analysis. Hearst and Plaunt [1993] have investigated using a statistical parser segment text into topical elements. As text is scanned from top to bottom, a sliding window can be programmed to process chunks of the text. The output is analyzed to determine when a subtopic is being introduced. The method is a motivated segmentation that reflects a text's underlying subtopic structure, which can span paragraph boundaries. TextTiling is a two step process that first compares adjacent blocks of text and assigns a similarity value. The blocks are usually 3-5 sentence units. The second step involves graphing the resulting similarity values and smoothing the generated curve. Peaks in the curve indicate regions of high subtopic coherence, whereas valleys indicate evidence for topic switching. Large expository documents were subjected to testing by asking volunteers to perform a topic identification task [Hearst and Plaunt, 1993]. The results showed that there was a high degree of correlation between the judgements of the subjects and the TextTiling algorithms. Linear data types can also be in the forms of textual documents, program source code, and alphabetical list of names. All these are normally sequential in their manner [Morse, 1997, Shneiderman, 1998,].

Two-dimensional

There are several ways that text can be viewed in two dimensions. The first way to view text as 2D is to focus on the characteristics of the text as it appears on a page. The key feature is the formatting, such as paragraphs, headings, tables, and general use of 'white space'. The 2D view of text is especially productive of metaphors. Several visualizations have been developed that build on the tangibility of printed matter. For example, people dog-eared pages to provide bookmarks and also do underlining and annotating. The printed page is familiar and provides a great deal of utility apart from its primary function; it is only reasonable and natural that graphical interface designers would borrow from this metaphor. Rather than trying to determine semantics of the page's content, implementers only show pictures of the page. This is a very simple but versatile mechanism for conveying to a user something about a text. Zooming in on a page reveals successively more and more detail and allows a user to orient himself with respect to the organization of material.

This data type is also called as planar because it must have two planes to be in a 2-dimensional view. The examples of 2D data are geographic maps, floorplans, and newspaper layouts [Morse, 1997]. Commercially, the geographic information systems (GIS) is the example of a successful visualization system.

Three Dimensional

Just as the planar or two-dimensional page can give rise to a unique view of text, a view of books as 3-dimensional objects can also serve to characterize another useful view. The tangibility of books, the feel of the pages, their location in physical space, the color of the bindings are the characteristics of 3-dimensional data or text. This view has given rise to several popular metaphors for graphically rendering documents. Those metaphors are like the desktop, piles and rooms.

Mander et al., implemented the "pile' metaphor [1992]. This is analogous to a pile or a heap of documents on a desk. The documents retain the order in which they were placed in the pile and some of their appearance such as their color. The pile of documents is displayed as a small perspective drawing, piles created by the user have a disarrayed appearance, and those created by the system appear to be neat. The design includes a gesture for spreading out the elements of the pile so they are all visible (a horizontal back and forth movement), and a gesture for starting to browse the pile elements (an up and down motion). The browsing operation uses a viewing cone, where the miniature document is displayed facing the viewer on the base of a pyramid pointing back towards the document's position in the pile [Morse, 1997].

The Room system [Henderson and Card, 1986] exchanges the idea of a single extended virtual surface for a collection of virtual screens of normal size. The reasoning is that typical work patterns are clustered into a collection of tasks between which people switch, and these tasks are not spatially related. The system also allows a window to appear in more than one room, and even to have a different location and shape depending on what room it is being seen from. An extension of this system to three dimensions is described in Robertson et al., [1991].

According to Shneiderman [1998], the three-dimensional computer graphics and computer-assisted design are many but a working information visualization in three dimensions are still rare.

Multidimensional

The use of natural language processing to generate better document vectors has been the object of intense investigation for a long time. Methods for detecting phrases [Croft et al., 1991] and for extracting names and topics [Hahn, 1990] have enriched the knowledge of information retrieval researchers [Morse, 1997]. The advent of full-text rather than mere surrogates has opened the question of whether the old methods, which were developed to handle short text pieces, would scale up to handle full-text. The evidence shows that there is some degradation of processing effectiveness [Blair and Maron, 1985]. In order to capture the meaning of these longer texts, there has been a considerable effort to detect and encode the content of sub-passages of documents.

The commonly found relational and statistical database contents are the convenient usage for multidimensional data. Examples of multidimensional systems are like the HomeFinder [Williamson and Shneiderman, 1992], Bead [Chalmers, 1993], LyberWorld [Hemmje et al., 1994] and SPIRE [Wise et al., 1995].

Temporal

Temporal data, which mostly based on time as a dimension, is the same as linear or low-dimensional data when one considers the content of a document, e.g. the timeline in a novel or news story. It is different when considered as metadata, e.g. creation date or date of last reading. Liddy [1995] has explored extracting temporal information from text in a system called CHESS. CHESS automatically creates a knowledge base which aggregates information about any named entity (people, places, events, organizations, companies or ideas) and organizes that knowledge into a timeline which covers the entire period of the knowledge base.

Documents are created and edited in time. In paper form, text is finalized and published. Although electronic text is said to be published, it is more difficult to say that it is actually finalized. There is no guarantee that the content might not be changed, more words added, sections removed, the whole reorganized or it might even disappear. Most documents do not have a version history. Computer programs have such records if they are maintained in a version control system. Similarly, legal documents and many electronically managed documents are tracked temporally. Some of the proposals for metadata standards include expanded temporal data fields [Morse, 1997, Ng et al., 1997].

The temporal data normally have start and finish time and the items in it may overlap. Tasks usually used by users in this kind of systems are like finding all events before, after, or during some time period or moment [Shneiderman, 1998]. LifeLines [Plaisant et al., 1996] is perhaps the most popular example of information visualization's temporal data prototype.

Hierarchical Trees

Many types of data lend themselves to representations as trees, including structured documents, directories, and some kinds of hypertext. Many approaches have been developed to render these spaces. Conventional methods merely draw a tree as large as it needs to be and then render an image that is controlled with scroll bars. This process has the problem that the user is prevented from seeing the overall structure and must keep most of a large space in memory rather than in view. Although by their very nature, trees can be rendered in a plane, there is no satisfactory 2D layout of a large tree [Lamping et al., 1995, Morse, 1997]. In order to make room for the leaf nodes, the nodes near the root must be placed far apart.

However, it is clear that trees are useful for representing large collections of documents. Hierarchical structures like in trees are fundamentally how humans structure information and knowledge. It is no coincidence that organizational charts, tables of contents, topic indexes, outlines, product catalogs, web sites, decision trees, risk analyses, family trees, food chains, and taxonomies in general (e.g., the Dewey Decimal System, the taxonomy of the species, etc.), are arranged hierarchically.

Networks

Many views of documents can be thought as networks. Queries, semantic networks, and hypertexts can all be represented as networks. Multidimensional data discussed in the earlier section, differ qualitatively from network data in that the latter have dependencies among the parts. Multidimensional scaling methods tend to drive concepts apart, i.e., to find orthogonal dimensions, while networks assume dependencies among the concepts being manipulated.

Although paper hypertexts exist, the importance of document networks rests on the fact that the Internet is based on hypertext. Documents are connected to other documents through links and nodes. Attempts to bring order to the potential chaos of hyperlinks causing further disorientation is a great field of study at the moment.

Network displays can represent more general and more complicated structures than hierarchical tree displays. The complexity of the information spaces when expressed as networks can be difficult for users to comprehend. A major issue then is how to simplify such displays without losing critical information. One method for reducing complexity is to reduce the dimensionality of the space. This problem is also an interesting area of study to be researched.

3.4.2 Interface Functionality

Overview

For a visualization to be effective, it must provide the user with a sense of the overall composition and layout of the space. For complicated displays such as those that attempt to render large hierarchies using trees or any representation of a large document collection, this task is not as easy or obvious as it sounds.

Several issues arise when a data set is to be mapped to an interface, such as how to make the best mapping of the attributes of the data to attributes of objects in the interface. Spring and Jennings [1993] have provided a comprehensive account of the dimensions that might be mapped in an interface. They categorize each of the stimuli as to its suitability to map to data depending on whether the data is nominal, ordinal, interval or ratio. Bartram [1997] has recently raised the issues of incorporating motion as a key feature of complex display due to its easy perceptibility. Other constraints apply when deciding how to map data since different stimuli convey different perception of importance to users. For instance, it has been shown that tilted lines are more readily apparent than vertical lines. Similar observations were made with respect to curvature, color, line ends, movement, closure, contrast, and brightness [Treisman, 1986, Morse, 1997].

Another issues that arises when addressing the overview of visualizations is how to fit large spaces on the screen and still allow some appreciation of the detail that resides there. Toward encounter this, concept and prototype with the fisheye view has been developed [Furnas, 1986, Sarkar and Brown, 1994]. The space with a fish-eye lens on it is distorted so that the view is expanded under the lens. Problems can occur if large areas of the screen are distorted; many types of tasks cannot be performed under these conditions, such as comparing two points that are of different magnifications.

Zoom

Zooming is the technique for allowing a user to select a smaller region of the screen for display. Scrolling is an alternative to zooming but suffers greatly by comparison. Since only a portion of the display can ever be visible at one time, pieces of information that are at opposite ends of the display will never be subjected to some types of evaluations. Zooming includes any change in view from a larger portion to a smaller portion of a field or vice versa. As such, it is possible to implement zooming as a discrete number of intermediate views. Usually such views are available simultaneously to help user to preserve his sense of place. However, smooth zooming is increasingly available and helps users to preserve their sense of position and context [Schaffer et al., 1996].

Smooth zooming has been incorporated into many of the currently available visualizations including Pad++ [Bederson and Hollan, 1994] and the Document Lens [Robertson and MacKinlay 1993]. The availability of fast algorithms and state-of-the-art hardware that incorporates graphical routines has made rapid screen update rates possible [Morse, 1997]. Variations in zoom techniques include the capability to move in more than one plane. In Pad++, the user needs only to hold down the mouse button on a location and the view will be transformed to move that region to the center of focus. To the user, it appears that she has walked straight line toward the region [Bederson and Hollan, 1994].

Filtering

Filtering is the activity of weeding out uninteresting elements in a collection [Shneiderman, 1998]. When the user has the control towards the contents' display, then they can quickly focus on their interests by deleting unwanted items. With databases this is accomplished quite easily. Ahlberg and Shneiderman [1994a] has developed an Alphaslider, which maps an alphabetically sorted list to a slider, such that repositioning the thumb causes the list to be traversed in the expected order. The Alphaslider can be found in many visualization projects including FilmFinder [Ahlberg and Shneiderman, 1994b], Spotfire [Ahlberg, 1996] and HomeFinder [Williamson and Shneiderman 1992]. A common term for this type of filtering is dynamic query [Ionnidis, 1996].

Details-on-demand

At some point in interacting with a visualization system, the user may decide to take a closer look at one or more objects in the field of view. When the requested view provides the content of the object, then it is said that 'detail-on-demand' is provided. Most systems support this function and it is usually invoked by clicking on an item or group of items or by allowing the sprite (cursor) to dwell on an object. In the former case, a dialog pops up that contains detailed information. In the latter case, a lens might be provided. Lenses have a variety of appearances but as a group they provide what are commonly referred to as 'see-through-tools' [Bier et al., 1994]. Zooming can also provide details. When zooming magnifies a piece of a display, the view can show different information at the more detailed level.

Relate

The relate function seeks to make explicit the relationships between objects in a display. It can also refer to representing relationships data in multiple associated windows. This function is implemented in a variety of ways. The idea of linking graphical representations is not new. Simple linking can be found in a wide variety of programs, e.g., Bead [Chalmers, 1993], SeeSoft [Eick et al., 1992], The Table Lens [Rao and Card, 1994], and the Dynamic HouseFinder [Alhberg and Shneiderman, 1994b]. The Alphasliders found in FilmFinder and Spotfire [Ahlberg, 1996] are updated to current values each time an object is selected. When performing dynamic queries [Ahlberg and Shneiderman, 1994b], users are shown consistent views onto the data in a similar way. With this, the users can easily make connections between the pertinent relationships.

History

Maintaining histories is important for several reasons including placekeeping, supporting the ability to undo actions as well as to replay and to do refinement progressively. Exploration in visualizations is a creative process and involves many sequential user actions to arrive at a satisfactory solution. The ability to retrace steps on a particular path is important. Shneiderman [1998] suggests that "most prototypes fail to deal with requirement" and attributes this fact to the novelty of such interfaces.

Animation of steps might be a useful mechanism for providing path retracing [Gonzalez, 1996]. A problem with maintaining an adequate history is the considerable resources that are required to maintain the various kinds of information that might be considered salient by the user. In navigating visualization spaces, it might be important to keep track of the landmarks that are visible, the granularity of the zoomable display, the state of user-selectable options [Morse, 1997].

Extract

Once users have found regions or elements of interest in a visual display, they should be able to save the subsets. Not only might the user desire to save this collection as a new starting point for further study, but she might also like to print or mail it. An alternative to exporting a whole data set could be to save interface settings. This is the approach taken by VIBE [Olsen et al., 1992, 1993]. The Visage project supports a drag-and-drop feature that allows inter-application exchange of data [Morse, 1997, Roth et al., 1997].

3.5 EXAMPLES OF VISUALIZATION INTERFACES

This section will present samples of visualizations that have been developed to handle data of the various types discussed above. Table 3.2 contains a more complete listing of visualization systems that have applicability to documents.

Table 3.2 Information Visualization Interface Examples

Data Type	Interface	Reference
Linear	TileBars Document Lens	Hearst, 1995 Robertson and MacKinlay, 1993
2-Dimensional	Information Mural Pad++ Perspective Wall	Jerding and Stastko, 1995 Bederson and Hollan, 1994 MacKinlay et al, 1991
3-Dimensional	WebBook	Card et al., 1996
Multidimensional	Bead Lyberworld Themescape/SPIRE VIBE VR-VIBE	Chalmers 1993, 1996 Hemmje et al., 1994 Wise et al., 1995 Olsen et al., 1993 Benford et al., 1995
Temporal	Groupkit SeeSoft LifeLines	Greenberg and Roseman, 1996 Eick et al., 1992 Plaisant et al 1996
Hierarchical Tree	Cone Tree Hyperbolic Tree Tree Maps WebTOC	Robertson et al., 1991 Lamping et al., 1995 Johnson and Shneiderman, 1997 Nation et al., 1997
Network	Butterfly Citation Browser Multi-Trees Navigational View Builder	MacKinlay et al., 1995 Furnas and Zacks, 1994 Mukherja and Foley, 1995

Source: Shneiderman, Ben. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd Edition, Addison-Wesley, 1998

Linear : TileBars

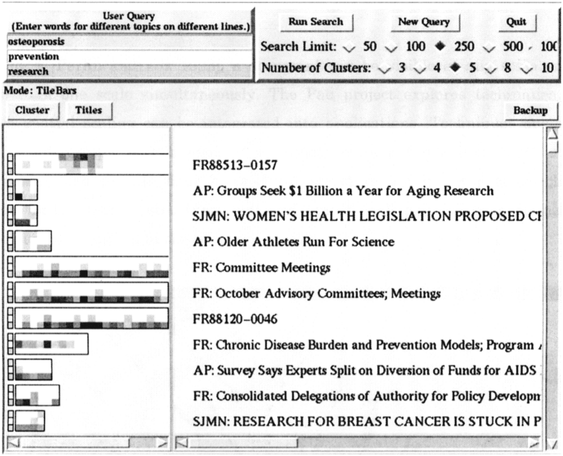

Figure 3.3 TileBars embedded in Scatter/Gather interface

TileBars [Hearst, 1995] is shown in Figure 3.3 above as part of a Scatter/Gather [Pirolli et al., 1996, Hearst et al., 1995, Hearst and Pedersen , 1996] interface. Each TileBar icon represents a single document and the length of the bar is proportional to the length of the document. Each grayscale block represents a segment of text as determined by the TextTiling method described previously. Dark blocks are segments with a high occurrence of a term or combination of terms and lighter blocks stand for pieces of text with relatively less of the topic. This particular query was composed of three sets of terms so the display for each document contains one row of blocks for each term set. This display makes some information easy to gather, such as relative size of documents, co-occurrence of term sets in a document, and absence of a particular concept from a document.

Two-Dimensional: Pad++

In Pad++ [Perlin and Fox 1993], a document can be visible at any scale or at more than one scale simultaneously. The Pad project explores techniques by which spatial scaling can be integrated into applications. Techniques such as placing microscopic text in place of a footnote marker, applications such as a calendar which reveals finer hierarchical structure as the user approaches, editors for hierarchical structured text, and a multi-scale painting program using wavelets are described in the text.

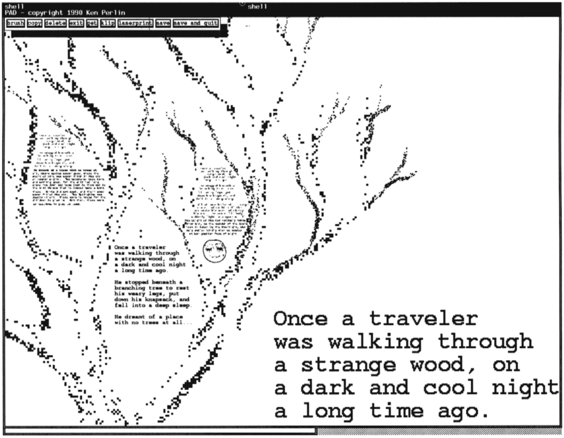

Figure 3.4 Pad++

The view of Pad++ shown in Figure 3.4 above is of a hypertext. Each node shows a text segment at a different level of detail. The developers of this system [Bederson and Hollan 1994] contend that zooming, the primary mechanism of interacting with Pad++ provides superior way-finding in many environments, including hypertext. Their study showed that subjects not only answered more questions correctly and in less time, but there was also greater subjective satisfaction.

Three-Dimensional: WebBook

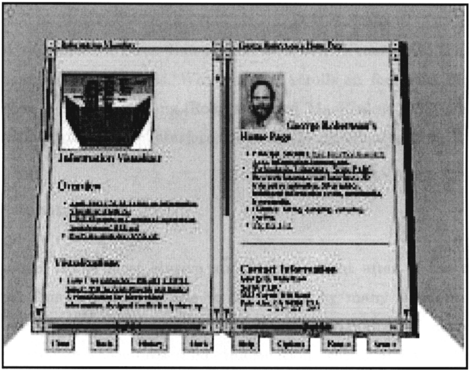

Figure 3.5 WebBook

Card et al., [1996] developed the interface shown in Figure 3.5 above in response to the observation that users have difficulty finding pages, get lost, have difficulty relocating pages, and have problems organizing material they manage to find on the Internet. Although the book metaphor has been used many times, [Yankelovich et al, 1985], the WebBook implementation is quite compelling. The WebBook is not just a static interface object but provides a variety of interactions that are typical of the way people use books in the real world. Users can riffle through pages. Other than that, the users can rip pages from a book, and can tack a page to a desk or wall in the 3D room in which the book is located. Even the use of bookmarks is more like in the real world being flat objects that are inserted between pages [Morse, 1997].

The WebBook represents a three-dimensional view of a virtual world that allows using documents at many levels. When a user scrolls in font size mode, the display becomes a Document Lens [Robertson and MacKinlay 1993] which has many similarities to the Pad++ interface shown in the previous section.

Multidimensional: SPIRE

Multidimensional is the most elusive and highly sought after of the types of visualizations. This is primarily due to the efforts by many segments of the information community, including to the database visualizers, geographical information systems specialists and information retrievalists. Even by concentrating on information retrieval visualization, it was difficult to decide from many systems such as Bead [Chalmers, 1993, 1996], VIBE [Olsen, 1992 , 1993], and LyberWorld [Hemmje et al., 1994]. SPIRE as shown in Figure 3.6 below is a project that was developed at Batelle National Labs and has gone into production by InXight (http://www.inxight.com). The goal of information retrieval projects is usually to find text that the user expects to read. SPIRE (Spatial

Paradigm for Information Retrieval and Exploration) stands in contrast to this view.

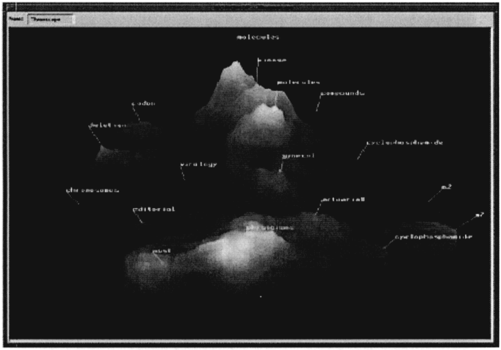

Figure 3.6 SPIRE Themescape showing topic distribution in a large document space

The Figure 3.6 shows a Themescape to which a large document collection has been mapped. The layout of the concept space has been accomplished with a variety of clustering and dimensionality reducing algorithms [York and Bohn, 1995]. The elevation of a feature in the map indicates theme strength. Time sequence animations allow a high level appraisal of the overall change in topics over a period of time.

Temporal: SeeSoft

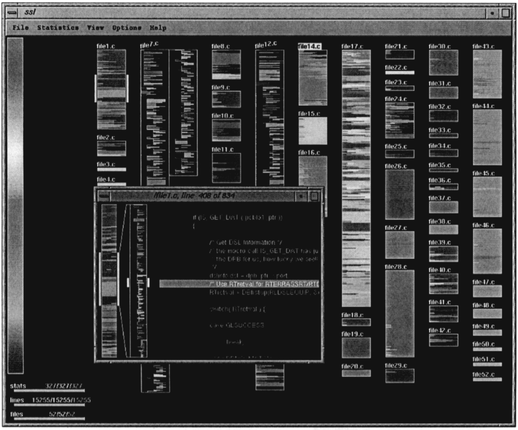

Figure 3.7 SeeSoft showing overview of a software project

Figure 3.7 shows the status of a large software project as filtered through SeeSoft [Eick et al., 1992, Morse, 1997]. The blocks in the figure represent program code modules. Color is being used to profile 'hot spots' in the code. The information in this rendering is actually a frequency count—the more often a line of code is called, the closer the color is to the red end of the color spectrum shown in the legend to the left side of the image. The smaller browser window shows a code overview and detail view. Other modes of this tool can show the time elapsed since a module was last modified.

Other visualizations depict time in a variety of ways. SAGE [Roth et al., 1994] has been used to code timelines of information extracted from historical documents. LifeLines [Plaisant et al., 1996] is used to show a youth's history keyed to the needs of the Maryland Department of Juvenile Justice but also is applicable to present individual medical histories (see Figure 3.8) as a compact overview with selectable items that allows uses to get details-on-demand [Shneiderman, 1998].

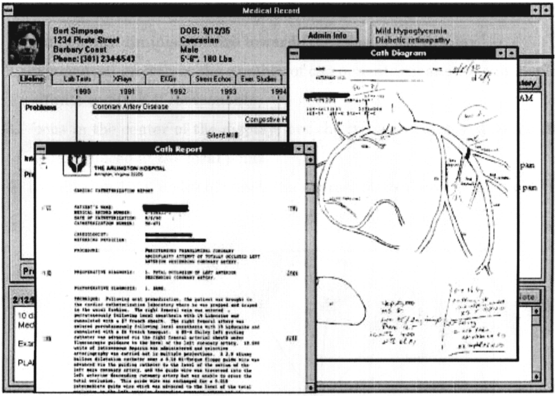

Figure 3.8 LifeLines in medical history records

Hierarchical Tree

a. Hyperbolic Tree

The hyperbolic tree [Lamping et al., 1995] shown in Figure 3.9 is representative of the types of renderings that are possible of hierarchical data. The essence of this scheme is to lay out the hierarchy in a uniform way on a hyperbolic plane and map this plane onto a circular display region. Central objects are larger than more peripheral ones. Interaction with the display is accomplished through a click and drag which cause an apparent rotation of the surface pulling peripheral parts into more central focus.

Hyperbolic Tree has features such as:

1. components diminish in size towards the edge of the display
2. exponential growth in the number of components means we can display an infinite number of components
3. focus in the center of the display, but the tree can be moved so a different node is placed at the focal point
4. can display 1000 nodes of which the nearest 50 can have text labels

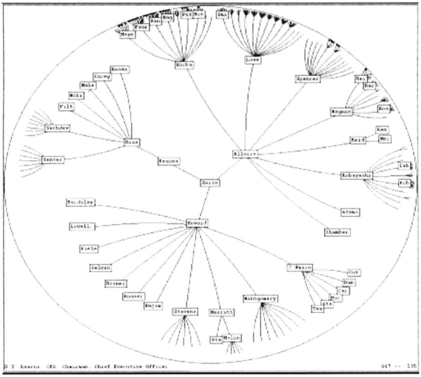

Figure 3.9 Hyperbolic Tree

The circumference of a circle on the hyperbolic plane grows exponentially with its radius, which means that exponentially more space is available with increasing distance. Thus, hierarchies which tend to expand exponentially with depth can be laid out in hyperbolic space in a uniform way, so that the distance (as measured in the hyperbolic geometry) between parents, children, and siblings is approximately the same everywhere in the hierarchy.

The advantage of Hyperbolic Tree is that it can display a similar number of nodes as can the Cone Tree but it does not require 3D animation support. Relatively, it just requires modest computation needs because it is in 2D. Thus, this makes it useful and can be used on a wider variety of platforms.

Other implementations of the hyperbolic tree include Cone Trees [Robertson et al., 1991] which project a tree into a semi-transparent 3-dimensional display. Trees of modest size can be rendered using this approach but very large trees are still difficult to manage. TreeMaps [Johnson and Shneiderman, 1991] are another way to maximize the use of screen real estate. They are a space-filling technique that uses alternating directions, icon size and texture to render large numbers of objects in a hierarchy. These displays require a considerable learning time and the hierarchy is often lost in the remapping.

b. Hierarchical Tree: WebTOC

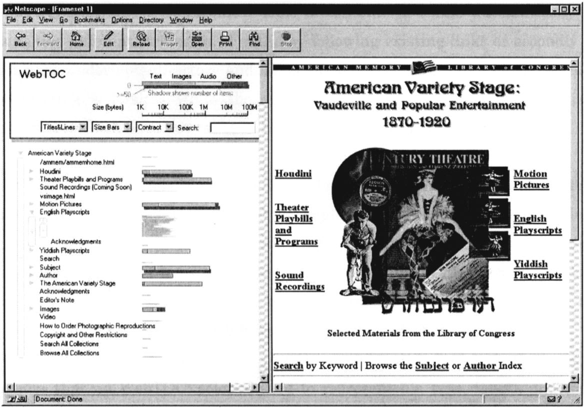

Figure 3.10 WebTOC showing it being used in the Library of Congress Web Site – American Variety Stage Collection

WebTOC or Web Table of Contents [Nation et al., 1997] as shown above is an individual collection browser tool to help webmasters and users in representing the quantity of information and its distribution within a set of linked documents. Digital librarians must manage huge directories and need tools to visual directory structures and data types. Knowledge of the quantity and types of documents available can be helpful to users as they decide whether a web site may be interesting or useful. To browse large numbers of documents in the growing collections is not an easy task since the size and extent of such large collections makes them hard to understand. WebTOC is a tool that can help both end users and library staff develop and organize collections.

WebTOC generates a hierarchical table of contents of a site automatically by using two different strategies. First is by following existing links or secondly by using the underlying directory and file structure. Following links us appropriate for existing web sites while the second option is good for newly digitized collections, which have not yet been linked, indexed or annotated.

WebTOC is a two-frame layout which on the left frame is the table of contents and on the right is the page selected for display. The user can make the table of contents in either contract or expand mode. WebTOC summarizes individual pages with individual lines while bars aggregate groups of files behind a link or directory to represent the total size of the included documents. A legend on the control panel frame (on the top left frame) is showing all these.

Besides that, in WebTOC, color is used to represent file type, length of overall size and the shadow below the bar is proportional to the number of documents included. The number of documents and size represented are displayed in the browser status area when the user's cursor is over the bar. WebTOC also provides a text search capability for the table of contents and display results in the context of the table of contents hierarchy. When a search is done in WebTOC, the results are shown by opening all (and only) the tree branches leading to result hits – therefore showing results in context. This is illustrated in the Figure 3.11 which a search is of keyword "Shakespeare".

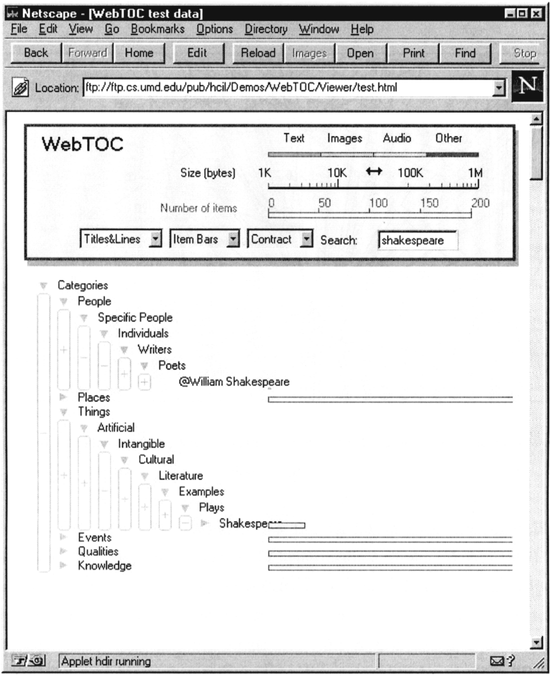

Figure 3.11 Searching for “Shakespeare” using WebTOC

Networks: Navigational View Builder

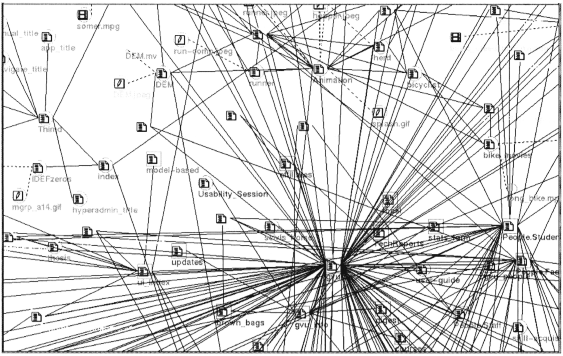

Figure 3.12 Navigational View Builder showing relationships among a set of documents

The figure above shows Navigational View Builder [Mukherjea and Foley, 1995], a tool for designing overview diagrams of hypermedia systems [Morse, 1997]. The rendering done by the tool are the results of a series of operations, including binding (mapping data attributes to visual display attributes), clustering (coalescing nearby objects into a single icon), filtering based on content, links, and structure, and hierarchization (reduce dimensionality by viewing 3D trees instead of graphs). The authors admit that the problem is difficult and that their solution can be markedly improved. The three main outstanding issues that they identify are:

1. the system has not been subjected to usability testing
2. the algorithms that they use have not proven to be scalable

3. the metadata that is currently available is too limited to provide interesting views to be built and the content is not captured in the data that they do collect.

3.6 SUMMARY

From the analysis of the information visualization techniques in this chapter, several synthesizes have been made. The thesis outcomes were manifested in the following specific design goals:

1. minimize disorientation by reducing navigation (e.g., minimize scrolling and jumping) and anchoring users in a consistent context. Thus, a hierarchical tree structure for the new National Archives of Malaysia web site will be implemented. The concepts of the new site are based from the discussion on hierarchical tree structure in Section 3.4.1;
2. provide primary information at the earliest point in the interaction as possible which will be achieved using a table of contents metaphor and implemented in the CRIVE web site. This concept is taken from WebTOC example (see Section 3.5) but new features are added to the proposed new interface.

The user interface (CRIVE) is designed to help users and archivists to examine and use different digital documents. The user interface developed consists of Java applets, one in the form of Table of Contents and another one as a control panel allowing users to do the following:

1. To set their preference on the Table of Contents (TOC) structure – in expand or contract mode
2. To let the user view the Legend used in the TOC
3. To provide basic information about the tool in About tab
4. To find the exact document or collection size
5. To do search and filtering on the site's collections