Navigating in Hyperspace: A Structure Based Toolbox

Ehud Rivlin*† Rodrigo Botafogo‡ Ben Shneiderman*§

Abstract

Analyzing the structure of a hypertext database can give useful information to the traveler in hyperspace. We present a preliminary collection of structural tools for users of hypertext systems. These tools can suggest answers to questions like: Where am I? How can I choose and get to my destination? What else is in my current neighborhood? etc. Structure is imposed on the hypertext by using two processes: hierarchisation and cluster identification. Several metrics are presented and used in the above processes for locating landmarks and getting global information on the hypertext structure. The structural analysis is integrated with previous attempts to reduce the users' disorientation while navigating the hyperspace. An integration with fisheye views and tree-maps is presented.

1 Introduction

As more books and other documents become available in electronic form, the use of hypertext systems is becoming more common. But as the size of hypertext databases grows the "lost in hyperspace" problem may limit efficient and meaningful usage of hypertext systems [11]. Good solutions to the problem need to be holistic [12]; encompassing navigation, planning, annotation etc.

We concentrate on the navigational aspects of the problem from a structural point of view and propose a set of tools that can help hypertext users reduce the cognitive load of navigation. Nielsen [11] describes the user's disorientation while navigating as "one of the major usability problems with hypertext". Our tools can suggest answers to questions like: Where am I?[1] How do I get to any destination? What can be seen from where I am? etc. These tools are based on a structural analysis of the hypertext. The analysis, and the answers we get, reflect the input that we are using - the structure. We can not get more than what a structural analysis can give. The results can be strengthened if the analysis were extended to include textual and stylistic dimensions as well. This natural extension is beyond the scope of this paper. The results presented can be integrated nicely with previous solutions to the problem, like the ones that are based on improved user interfaces, multiple windows [10]; or maps [11].

In this paper the word "hypertext" is used to indicate hypertext systems where nodes (cards, articles, documents, ...) and links are untyped, and where the hypertext can be represented by a directed graph[2]. This paper describes results working with 3 different hypertexts created in Hyperties[3].

Section 2 introduces some concepts and metrics used for our analysis[4]. Section 3 presents some possible answers to the question: Where am I? The answers are based on the ability to find hierarchies in the hypertext, as well as the ability to create aggregates based on graph theoretic algorithms. Section 4 describes the use of different metrics in giving users some notion of the hyperspace structure. For this purpose we also show how to integrate the structural analysis with previous work such as tree-maps and

*Department of Computer Science & Human-Computer Interaction Laboratory, University of Maryland, College Park, MD 20742-3411

†e-mail: ehud@cs.umd.edu

‡C&C Information Technology Research Laboratories, NEC Corporation 4-1-1, Miyazaki, Miyamae-Ku, Kawasaki, Kanagawa 216, Japan, botafogo%joke.cl.nec.co.jp@uunet.uu.net

§Send correspondence to Ben Shneiderman, ben@cs.umd.edu

[1] In [11] Nielsen quotes previous work where 56 precent of the readers agreed (fully or partly) with the statment *I was often confused about "where I was"*.

[2] Most of the hypertext systems have a backup function that gives the user the ability to go back to the last node that was visited. This option changes the hypertext from a directed to undirected graph. Although this fact is important from a user interface point of view, an analysis of the hypertext as a directed graph (without taking into account the backup option) reveals some properties of the structure.

[3] Hyperties is a hypertext system developed by the Human Computer Interaction Laboratory, now distributed and improved by Cognetics Corporation, Princeton Junction, NJ.

[4] Most of them were introduced in [2].

fisheye views. Section 5 points out the next move, namely, how to plan a path.

2 Concepts and metrics

The converted distance matrix

A matrix that has as its entries the distances of every node (n nodes in the hypertext) to every other node is called the *distance matrix* of a graph. Let M be the distance matrix of the graph. The *converted distance matrix* is defined as follows:

$$C_{ij} = \begin{cases} M_{ij} & if M_{ij} \neq \infty, \\ K & \text{otherwise} \end{cases}$$

Where K is a finite constant (the conversion constant). Typically K = n.

The converted and the relative out and in centrality metrics

The converted out distance (COD) for a node is the sum of all entries in row i in the converted distance matrix (C)

$$COD_i = \sum_j C_{ij}$$

The converted distance (CD) of a hypertext is given by the sum of all entries in the converted distance matrix

$$CD = \sum_i \sum_j C_{ij}$$

The COD is a good indication of the centrality of a node within a given hypertext. The need for normalization leads to the definition of the *relative out centrality* (ROC)

$$ROC_i = CD/COD_i$$

A high ROC indicates a node that can easily access other nodes (high out centrality). Hence, the higher the ROC of a node the more central it is. In a similar manner we can define the converted in distance (CID) for a node i as the sum of all entries in the column i in the converted distance matrix (C)

$$CID_i = \sum_j C_{ji}$$

The *relative in centrality* (RIC) is defined as:

$$RIC_i = CD/CID_i$$

Figure 1: Graph with its converted distance matrix and the associated metrics.

A high RIC indicates a node that is easily accessible (high in centrality).

Figure 1 shows a graph with the associated converted distance matrix and its COD, ROC, CID, RIC. The value of K was chosen to be the number of nodes in the hypertext[5]. Node c has the highest ROC and intuitively we see that this is the most central node in the graph.

3 Where am I? Revealing the structure

A possible way to localize a user in a hypertext is to impose a structure on the hypertext, and to identify the user's location within that structure. A natural structure is a hierarchy, and some systems like KMS require a hierarchical structure [1]. The role of hierarchy formation in reading is presented in [4, 16]. In [2] we proposed an algorithm for hierarchisation of hypertexts. The hierarchisation process is a two phase algorithm: first a root is identified and then hierarchical and cross-reference links are distinguished.

A good candidate for a root is a node with a high ROC. Intuitively this is a node that can access a large

[5] The higher the value of K, the more central is the requirement that a node needs to reach **all** the other nodes in order to have high ROC. See [2].

Figure 2: Hierarchisation of a graph.

number of nodes and the distance from it to other nodes is small. In order for the algorithm to give meaningful results a "noise removal" process should precede the ROC computation. In this process we remove index nodes, i.e. nodes that points to a very large portion of the nodes in the hypertext[6].

The differentiation between hierarchical and cross-reference links is done with a variation of breadth first search trying to maintain nodes as close to the root as possible. Figure 2 shows a graph and its hierarchisation using node c as the root. Hierarchical links are represented with normal lines, while cross-reference links are represented with dashed-lines. Note that a node which has two parents from the same level appears twice, i.e. node f. In this case redundant subtrees were created. This situation raises the option for two semantic interpretations for node f: one as part of the subtree headed by node a, the other as part of the subtree headed by e.

Once a hierarchy has been found it is important to show it to the user in a reasonable way. One option is to display the entire generated tree. This option might not be practical for a hypertext with a large number of nodes (this is of course a function of the linearity of the tree as well). Another option might be a fisheye based [5] presentation. This presentation is based on the observation that humans usually represent their own neighborhood with great detail, yet only major landmarks further away. The importance of a node (the a priori interest) can be taken to be the distance from the root. This option is discussed in the next section.

[6]If we define α to be the mean of the outdegrees of the nodes in the hypertext, an index node is defined to be a node whose outdegree is greater than α plus some predefined threshold. In order to create a "cleaner" structure we can take out reference nodes as well. Reference nodes are in a certain way the converse of index nodes, and are defined similarly on the indegree, see [2].

Figure 3: Asking: Where am I? being at the node "CMSC 412". Hierarchisation of the CMSC hypertext. Only clusters on the path from the root are presented.

We applied the hierarchisation algorithm to the CMSC hypertext. CMSC is a hypertext with 106 nodes and 402 links which describes the computer science department at the University of Maryland. The hypertext contains a description of the facilities at the department, biographies of the faculty members, details about the undergraduate and the graduate programs as well as a description of the courses given at the department. In running the algorithm we identified three index nodes (using as a threshold three times the standard deviation, which gave 18 nodes above the threshold) and removed them. The node having highest ROC was the "Introduction" (ROC = 1838). The second highest ROC was 1000 and from there on the ROC droped significantly, with the $10th$ highest ROC equal to 400. This reflects the fact that CMSC is a well structured hypertext, with a clear node as the root. Some other hypertexts that we looked at did not have this feature. GOVA[7] [13] for example, is a hypertext which contains information for users who want to become volunteer archaeologists. It contains information about sites, archaeology in general, accommodations and so on. The hypertext is a collaborative work of several writers, who did not have a consistent point of view. This resulted in small differences between the candidates for the root without any dominant node. In CMSC, on the other hand, there is a coherent structure aimed to present the department. This fact is reflected in a single clear candidate for the root.

Figure 3 shows the result of asking: "Where am I?" being at the node "CMSC 412 - Operating Systems". This node describes the advanced undergraduate course (or low level graduate) in operating systems. Choosing the node "Introduction" to be the root, the algorithm found this node as one that belongs to the "Graduate Courses" cluster. This is one of the eight nodes directly connected to the root. The clusters varied from laboratory facilities to fields of research, University of Maryland, graduate courses etc. The "Graduate Courses" cluster (or chapter) has 6 subclusters, each representing a different group of courses from a different domain. The node "CMSC 412 - Operating Systems" was under the subcluster "Computer Systems Courses". Thus, the user will get the following answer: You are in the subcluster of "Computer System Courses", that is a subcluster of the "Graduate Courses" cluster which is connected to the root ("Introduction").

An ambiguous answer is possible when querying: "Where am I?" being in the node "Rosenfeld, Azriel". According to the algorithm the node is included in two subclusters[8] , namely "Research in Computer Vision" and "Center for Automation Research", which in turn are connected respectively to the clusters "Fields of Research" and "Collaboration with Industry". This situation is natural. Prof. Rosenfeld is

Figure 4: Asking: Where am I? being at the node "Rosenfeld Azriel". Two views that are equally justified.

working in the area of computer vision which is one of the fields of research in the computer science department. On the other hand Prof. Rosenfeld is the head of the Center for Automation Research, which has connection to industry. These two views are equally justified, and both will be presented to the user (figure 4) This kind of ambiguity can be solved by using the user's history as a guide. This, of course, can be the user's choice.

For comparison we present an overview map of this node (figure 5) The map presents all the neighbors of this node at distance one, and most of the neighbors at distance two. Even in a relatively small scale hypertext the number of the possible links in a small environment[9] can be quite high (and confusing).

It is interesting to note that picking different nodes as a root gives a different view of the same hypertext. For example, running the algorithm on GOVA resulted in ten candidates for a root. Different perspectives resulted from taking a different root. For example taking "Archaeological projects" to be the root we find the node "Caesarea's harbor" under the cluster "Harbor archaeology" which is connected to the root. When taking "Introduction" to be the root we find the node "Caesarea's harbor" under the clus-

[7] GOVA stands for Guide to Opportunities in Volunteer Archaeology. It is a museum application by the Smithsonian. The hypertext contains 222 nodes and 1609 links and is highly interconnected. In this application users would use the hypertext to find places around the world where they are interested in becoming volunteer archaeologist. Using a touchscreen they were be able to select in maps a region of the world or even a particular city in which they were interested. Once the place was selected the user would then read a text explaining the price, best time to go, etc. The system was not limited to only those operations, users had at their fingertips information about the sites: period of history, geographic information and others that were all linked forming a network of information.

[8] This is exactly the case as with node f in figure 2

[9] Looking at a node and its immediate neighborhood, defined in [11] as local map

Figure 5: The node "Rosenfeld Azriel", its distance-1 neighborhood and some nodes at distance 2.

Figure 6: Two different answers being at the node "Caesara's harbor".

ter "Caesarea" which is connected to the root. This view results from the Smithsonian presentation being centered around excavation in Caesarea and being called "King Herod's Dream". Viewing the hypertext from the "Introduction", Caesarea was a cluster, a central part in the presentation. Viewing from "Archaeological projects" Caesarea was only another site and that is why it was found on level 5 (and of course wasn't a main cluster).

Using aggregation

Aggregation is an operation that clusters related objects and forms a higher level object. Halasz [6] identifies aggregation as a composition mechanism. A composition mechanism is a way of dealing with a set of nodes and links as a single object. In a hypertext database the presence of a link between two articles indicates that there is a semantic relation between those two articles. The semantic relationship between the nodes is reflected in the structure.

In [3] two graph theoretic algorithms that identify semantic clusters were presented[10] . The algorithms are based on finding the biconnected components[11] and strongly connected components[12] of a graph. First outgoing edges from index nodes and incoming edges to reference nodes are removed, and then a reduced graph is built by breaking the graph into its biconnected components. The algorithms are recursive in the sense that every bicomponent found is treated as an independent graph. These algorithms give users the option to break the hypertext into semantic clusters. Users can display the reduced graph and see their location.

In figure 7 part of the reduced graph of CMSC is presented, after the first run of the biconnected components algorithm. The algorithm divided the graph into a central core and some subparts. A similar partition (central, relatively large, core) resulted when the algorithm was run on the HHO hypertext[13] (see figure 9). Users being, for example, in the

[10]A semantic cluster of a hypertext is a set of nodes and links that are a subgraph of the hypertext graph, and the compactness of the subgraph is higher than the compactness (defined in section 4) of the whole graph. This is a set of nodes appropriate for aggregation. See [3].

[11]A graph G is biconnected if it is connected and has no articulation points. A node a of a connected graph is an articulation point if the subgraph obtained by deleting a and all the edges incident on a is no longer connected. As a result biconnected components in a graph have the property that there are at least two paths between any two nodes in the component.

[12]The same definition but for directed graphs.

[13]"HyperText Hands-On!" is the first electronic book commercially published. The hypertext covers concepts of hypertext, typical hypertext applications, and currently available authoring systems. The hypertext also covers design and implementation issues such as user interface, performance and

Figure 7: Part of the reduced graph of CMSC. The user is in the node “Howard, Elman”, in the “Research in numerical analysis” cluster. The cluster is a reduced graph that contains two nodes and a subcluster.

node “Howard, Elman” can see some different clusters that constitute the hypertext, and their semantic meaning (a pleasing outcome that can not be always guaranteed). A further application of the process to the cluster that contains the node “Howard, Elman” gives more information. Users can see that the nodes “O’Leary, D.” and “Stewart, G. W.” create an appropriate subcluster because these three faculty members constitute the numerical analysis group. This is an interesting piece of semantic information that is given by the structure. The breaking of the hypertext into components, and being able to see their connections, is important, and might give users a notion of their position by using the different clusters as landmarks.

4 Where am I? Looking around

The previous section presented tools to help users toq localize themselves in the hyperspace. This section presents tools to give users useful information about their environment. The information is gained by defining and analysing metrics on the hypertext. We use global as well as local metrics. Global metrics are concerned with the hypertext as a whole, while node, or local, metrics focus on properties of individual nodes.

Local metrics

A simple metric that can give users a useful hint about their position is depth. By definition, the depth of a node in a hypertext is its distance from the root. The hierarchisation algorithm imposes a tree structure on the hypertext and by doing so gives the depth information.

Two inter-related local metrics are the *status* and *contrastatus* (first defined in [8]). The status is the sum of the finite entries in the i^{th} row of the distance matrix, and the contrastatus is the sum of the finite entries in the j^{th} column of the distance matrix. When a node has a very low status, for example, it means that it cannot reach many of the other nodes in the hypertext. When a node has very low contrastatus it means that it cannot be reached by many nodes. The *prestige* of a node is define to be the difference between its status and contrastatus. Note that the prestige might have negative values too. Figure 8 presents the graph from figure 1 with the values for the different metrics. Nodes that reach all the other nodes in the hypertext have status identical to their COD metric. There is a difference between the ROC and the status (and the prestige). Node c, for example, has the highest ROC value, but node e is more “prestigious” (its subordinates are further away).

The local metrics can be used as guidance for users. Readers might find out that after wandering about they are located only two nodes away from the root (depth 3). Being in a node with low status and high contrastatus might suggest the use of aggregation instead of hierarchisation as in the following example. In “Hypertext Hands-On!” (HHO) [14] one part of the hypertext consists of a hyper-novel, which can only be accessed through an article that introduces the novel. Since the nodes in the hyper-novel can only reach a very limited subset of the hypertext they have low status and since they can only be reached through the hyper-novel introduction they have a high contrastatus. A biconnected component analysis reveals the user’s position in a separate cluster. Figure 9 shows part of the reduced graph of HHO after the first iteration of the algorithm. For users who are brows-

networks. The hypertext has 243 nodes and 803 links.

Figure 8: The graph from figure 1 with the status, contrastatus, and the prestige.

Figure 9: Part of the reduced graph of HHO. The location of the user is indicated by an arrow.

ing through the "Interactive fiction" cluster, the analysis gives a crude notion of their whereabout in hyperspace[14]. We were pleased to find that the automated metrics revealed the intended structure created by the authors.

Global metrics

We define two metrics, *compactness* and *stratum*, that indicate how complex the hypertext is.

The compactness varies between 0 and 1, and it is 0 when the hypertext is completely disconnected and 1 when it is completely connected. Formally

$$CP = \frac{Max - \sum_i \sum_j C_{ij}}{Max - Min}$$

Where Max and Min are respectively the maximum and minimum value the converted distance can assume. C_{ij} is the converted distance between nodes i and j[15].

Stratum is a measure that indicates if there exists a "natural" order for reading the hypertext. Maximum stratum is achieved in a linear hypertext. If the stratum is zero the hypertext has no hierarchy. Basically the stratum metric indicates how much of a linear ordering there is in the hypertext. We define the *linear absolute prestige* (LAP) of a hypertext with n nodes to be the absolute prestige of a linear hypertext with n nodes. The stratum is define as the absolute prestige of the hypertext divided by the linear absolute prestige. For example, in figure the absolute prestige. The linear absolute prestige is given by (odd number of nodes, see [2]): $n^3 - n/4$, that is 30, and so the stratum is $14/30 = 0.47$.

Note that stratum and the compactness are not independent measures. for example, if the compactness is equal to 1 (the hypertext is totally connected) its stratum is 0.

Readers can use the above metrics to get useful information about the nature of the hyperspace. The stratum, for example, is a measure for the organization of the hypertext. In [2], three hypertext databases were checked: CMSC, HHO and GOVA and their stratum compared. Though HHO has more nodes than GOVA its stratum is five times as large

[14] It is interesting to note that for some nodes in the interactive fiction cluster the hierarchisation algorithm gave depth 9. This is an example were a local metric (depth) suggests using aggregation rather than hierarchisation for a crude global view.

[15] The maximum value of the converted distance is given by $Max = (n^2 - n)C$ where n is the number of nodes in the hypertext and C is the maximum value an entry in the converted distance matrix can assume. Usually C=K, where K is the conversion constant. The minimum is given by $Min = (n^2 - n)$, since the minimum distance between two nodes is 1.

(0.054 to 0.011). This reflects the fact that HHO is a well structured hypertext, where GOVA is not. Similar information can be derived from compactness measurements. These global metrics are useful when the hyperspace is broken into biconnected components (see previous section). In that case we can get information on the nature of the different clusters.

Looking around using tree maps

Being able to impose a hierarchical order on the hypertext graph creates the option to use it for presentation. Limited display space makes the use of traditional node and link diagrams impractical for large trees. A novel method for the visualization of hierarchically structured information called a tree-map is presented in [15, 9]. The tree-map visualization method maps hierarchical information to a rectangular 2-D display in a space-filling manner; 100 % of the available display window space is utilized. Tree-maps partition the display space into a collection of rectangular bounding boxes representing the tree structure. The drawing of nodes within their bounding boxes is entirely dependent on the content of the nodes. Since the display size is fixed, the drawing size of each node varies inversely with the size of the tree. The efficient use of space allows large hierarchies to be displayed.

In [9] the weight assigned to each node is used to determine its bounding box. The weight may represent a single domain property, or a combination of properties. Some optional properties that can be used for weight representation are size, the number of siblings, ROC, or prestige. We can display the whole hypertext to the user, or only portion of it (current position and down, or some pre-defined environment). In figure 10 a tree-map of CMSC hypertext is presented. As a weight measure the total number of siblings from a node down (or the size of the tree below the node) was picked. The user's location can be displayed by a directing arrow. Users can get the general position in the hypertext and some notion on how they can proceed. The information is more meaningful after index and reference nodes are removed.

Using the structure when looking around - fisheye views

As was mentioned before, as the hypertext becomes larger, it is impractical to display the entire tree. A possible solution is to use fisheye views. Furnas [5] based his model on the observation that humans usually represent their own neighborhood with great detail, yet only major landmarks further away. Furnas suggests a "degree of interest" (DOI) function which assigns a value to each node in accordance to the degree with which the user is interested in seeing that node. The DOI function for node x given that the reader is in node y is defined as

$$DOI_{fisheye}(x|y) = API(x) - D(x,y)$$

where $API(x)$ is the global a priori importance of x and $D(x,y)$ is the distance between x and y. The fisheye view is created after picking a threshold, and displaying only nodes with DOI greater than the threshold.

Several options exist for choosing our API. A natural selection might be the distance from the root. This choice gives (according to the threshold value) the main parts of the tree. Another option can be to take the ROC metric and use it as API. As we mentioned before we limited our analysis to struc-

Figure 10: A tree-map of the CMSC hypertext after the index and reference nodes were removed. The black rectangle presents the position of the user ("research in computer systems"). The node on the right side, that has 31 children, "undergraduate courses".

Figure 11: A fisheye view of the CMSC hypertext being in nodes “Shneiderman, Ben” and “Hyperties”.

tural analysis only. Better results can be achieved by combining the structural analysis with a stylistic and context analysis. This is another natural place to use semantic information in determining the API. Another option is to use the user behavior - like the number of times a node was visited.

In figure 11 we present two fisheye views, one with the user in the node “Shneiderman, Ben” in the CMSC hypertext, and the other being at the node “Hyperties”. The views were created using the prestige metric as the API function. Note the difference between the two views, although the points of view are from neighboring nodes . The user can change the threshold value to get a more/less detailed view. Another option is to change the negative weight that is given to the distance. Doubling the punishment on being far will result in showing only nodes with very high API values, and the immediate neighborhood.

5 How do I get to any destination?

Knowing where we are is a first step toward a natural goal - getting there. Getting there at once (jumping via an index to the target node) is not a hard to achieve goal. Most hypertext systems give the option to move directly to a certain node (via search, or an index). Assuming that the user has some purpose in mind while wandering about in the hyperspace we see the path that the user takes toward the goal to be as important as getting there. *** might put here the learning from the paper *** A structural analysis of the hypertext can suggest some alternatives in planning this path.[16]

Our toolbox gives users the option to impose hierarchical structure on the hypertext. Users can pick the target node on the hierarchy and get two kinds of paths: the shortest (cross reference links are allowed), or “follow the tree” path (shortest on the tree, no cross reference links are allowed). (Do we need an Example here?). Being able to present semantic clusters, the toolbox gives users the option to plan a path that will bring structure into consideration (for example bypass them). Figure 9 shows part of the reduced graph of the HHO hypertext, when users are somewhere in the “Interactive fiction” cluster. For users who consider to go directly and browse through the “Travel guide” section, the reduced graph can help in making that decision. The separation might suggest that there is not much “to lose” by bypassing the core cluster.

Another option that users have is to plan a path through landmarks. The landmarks can be viewed using a fisheye presentation. Another option is to pick the landmarks by checking some structural metrics like prestige, and plan the path through these nodes. To carry out this option the system has to be able to suggest a path to users that is constrained by the landmarks the users have picked (This can be done by a collection of local shortest paths).

Structural analysis is only one dimension to consider while planning a path. Adding constraints from other dimensions (content, stylistic, etc) makes the mission of helping users to plan a path a complex job. This part is left as a direction for future research.

[16] This direction suits nicely in the learning support environment [7] under orienteering as well as rambling. Hammond and Allinson present the need for tools which “not only support passive navigation around the information but also permit learners to structure the information and the learning activities in accordance with their requirements.” On top of the structure imposed on the hypertyext the system can offer the novices a meaningful (from structural point of view) path between two nodes.

6 Conclusions

Analyzing the structure of a hypertext database can give useful information to the traveler in hyperspace. We presented a preliminary collection of structural tools (processes and metrics) for users of hypertext systems. These tools can suggest answers to questions like: Where am I? How do I get to any destination? What can be seen from here looking around? etc. The crude positioning in the hyperspace was based on two processes: hierarchisation and cluster identification. Several metrics were presented and used in the above processes as well as for locating landmarks and getting global information on the hypertext structure (compactness, stratum, etc). It should be emphasized that the analysis, and the answers we get, reflect the input that we are using - the structure. We can not get more than what a structural analysis can give. The results can be much stronger if the analysis were based on the textual and the stylistic dimensions as well. This extension as well as the multi-constraints path planning is a direction for future research.

References

[1] Akscyn, R. M., McCracken, D. L. & Yoder, E. A. (1988). KMS: a distributed hypermedia system for managing knowledge in organization. *Communications of the ACM*, 31 (7), 820-835.

[2] Botafogo, R. A., Rivlin, E., & Shneiderman, B. Structural analysis of hypertext: Identifying hierarchies and useful metrics. *ACM Trans. on Information Systems.* (to appear).

[3] Botafogo, R. A., & Shneiderman, B. (1991). Identifying aggregates in hypertext structures. *Proceedings of the Hypertext 91 Conference*, Available from ACM, New York, NY.

[4] Charney, D. (1987). Comprehending non-linear text: The role of discourse cues and reading strategies. *Proceeding of the Hypertext 87 Conference* (pp. 109-120), Chapel Hill, North Carolina.

[5] Furnas, G. W. (1986). Generalized fisheye views. *Proceeding of the CHI 86 Conference*, (pp. 16-23). Available from ACM, New York, NY.

[6] Halasz, F. G. (1988). Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31 (7), 836-852.

[7] Hammond, N. & Allinson, L. (1988). Travels around a learning support environment: rambling, orienteering or touring? *Proceeding of the CHI 88 Conference*, (pp. 269-273). Available from ACM, New York, NY.

[8] Harary, F. (1959). Status and contrastatus. *Sociometry*, 22, 23-43.

[9] Johnson, B. & Shneiderman, B. (1991). Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. *Proc. IEEE Visualization 91 Conference.*

[10] Marshall, C. C. (1989). Guided tours and on-line presentations: How authors make existing hypertext intelligible for readers. *Proceedings of the Hypertext 89 Conference*, 15-26. Pittsburgh, Pennsylvania.

[11] Nielsen, J. (1990). The art of navigating through hypertext. *Communications of the ACM*, 33 (3), 296-310.

[12] Norman, K. L. (1991). Personal communication.

[13] Plaisant, C. (1991). Guide to opportunities in volunteer archaeology. Case study on the use of hypertext system in a museum exhibit. in *Hypertext/Hypermedia Handbook* Berk, E. & Devlin, J. editors. San Diego, CA. McGraw-Hill Publishing.

[14] Shneiderman, B., & Kearsley, G. (1989). *Hypertext Hands-On!*. Reeding, MA. Addison-Wesley Publishing.

[15] Shneiderman, B. (1992). Tree visualization with Tree-Maps: A 2-d space-filling approach. *ACM Transactions on Graphics*, January 1992.

[16] van Dijk, T., & Kintsch, W. (1983). *Strategies of discourse comprehension.* New York: Academic Press.