NAVIGATING IN HYPERSPACE: DESIGNING A STRUCTURE-BASED TOOLBOX

Ehud Rivlin, Rodrigo Botafogo, and Ben Shneiderman

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 [10, 12]. Good solutions to the problem need to be holistic [14]; encompassing navigation, planning, annotation, and so on.

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 [12] describes the user's disorientation while navigating as "one of the major usability problems with hypertext." Our tools can suggest answers to questions such as: Where am I?[1] How do I get to any destination? What can be seen from where I am? These tools are based on a structural analysis of the hypertext. The analysis, and the answers we get reflect the input we are using—the structure. We cannot get more than what a structural analysis can give. The results can be strengthened if the analysis is extended to include textual and stylistic dimensions as well. This natural extension is beyond the scope of this article. The results presented can be integrated nicely with previous solutions to the problem, like the ones that are based on improved user interfaces (UIs), multiple windows [11]; or maps [12].

In this article the word "hypertext" is used to indicate hypertext systems in which nodes (cards, articles, documents, . . .) and links are untyped, and where the hypertext can be represented by a directed graph.[2] We describe results from working with three different hypertexts created in Hyperties.[3] Our toolbox was a research prototype, and much work remains to refine our software, integrate it with the hypertext system, and test the UIs.

Concepts and Metrics

The metrics presented here were developed in [2] to support effective authoring. In this article they are applied to support browsing an existing hypertext.

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:

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

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

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

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

[1]In [12] Nielsen quotes previous work in which 56% of the readers agreed (fully or partly) with the statement *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.

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:

$RIC_i = CD/CID_i$

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.[4] Node c has the highest ROC and intuitively we see that this is the most central node in the graph.

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 similar to KMS require a hierarchical structure [1]. The role of hierarchy formation in reading is presented in [4, 19]. Hierarchy facilitates reading. Readers understand and learn from texts more easily when the information is set out in well-defined structures. Empirical studies of reading comprehension also suggest [4] that when readers are given the responsibility of selecting what text to read, they may sequence the information poorly or omit important information altogether. In [2] we proposed an algorithm for hierarchization of hypertexts during the authoring process. The hierarchization 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 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 point to a very large portion of the nodes in the hypertext.[5] After this process the nodes are sorted using the ROC as a key. The system can suggest the top-ranked node as a root, and the user can accept or select any alternative node.

The differentiation between hierarchical and cross-reference links is done with a variation of breadth, first searching to maintain nodes as close to the root as possible. Figure 2 shows a graph and its hierarchization 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 (viewability is also a function of the linearity of the tree). 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 the only representations shown further away are major landmarks. The importance of a node (the *a priori* interest) can be taken to be the distance from the root. This option is discussed later.

We applied the hierarchization algorithm to the CMSC hypertext. CMSC is a hypertext with 106 nodes and 402 links describing 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 in 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 = 1,838). The second-highest ROC was 1,000 and from there on the ROC dropped significantly, with the 10th-highest ROC equal to 400. This reflects that CMSC is a well-structured hypertext, with a clear choice of which node is the root.

Some other hypertexts we looked at did not have this feature. GOVA,[6] [15] for example, is a hypertext that contains information for users who want to become volunteer archaeologists. It contains information about sites, archaeology in general, accom-

[4]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].

[5]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].

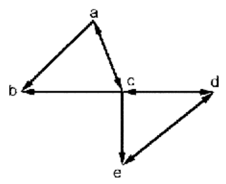
[6]GOVA stands for Guide to Opportunities in Volunteer Archaeology. It is a museum application developed by HCIL for the Smithsonian Institution. The hypertext contains 222 nodes and 1,609 links and is highly interconnected. In this application users would use the hypertext to find places around the world in which they are interested in becoming volunteer archaeologists. 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, and so forth. The system was not limited to those operations. Users had at their fingertips information about the sites: period of history, geographic information, and others that were linked forming a network of information.

modations 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 and 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. The "Graduate courses" cluster (or chapter) has six 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 users get the following answer: You are in the subcluster of "Computer systems courses," that is a subcluster of the "Graduate courses" cluster which is connected to the root ("Introduction").

An ambiguous answer is possible when asking: "Where am I?" if the user is in the node "Rosenfeld, Azriel." According to the algorithm the node is included in two subclusters,[7] 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. Rosenfeld is working in the area of computer vision, one of the fields of research in the computer science department. On the other hand, Rosenfeld is the head of the Center for Automation Research. The Center for Automation Research is supported partially by industry. These two views are equally justified, and both will be presented to the user (Figure 4). This kind of ambiguity can be resolved by using the user's history as a guide. This, of course, is a matter of 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 hypertext the number of links[8] can be quite high (and therefore confusing to the reader).



	a	b	c	d	e	COD	ROC
a	0	1	1	2	2	6	7.5
b	5	0	5	5	5	20	2.25
c	1	1	0	1	1	4	11.25
d	2	2	1	0	1	6	7.5
e	3	3	2	1	0	9	5
CID	11	7	9	9	9	45	
RIC	4.09	6.43	5	5	5		

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

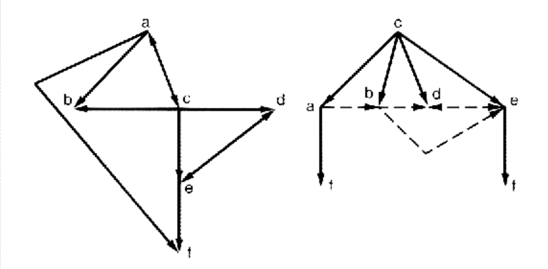


Figure 2. Hierarchization of a graph

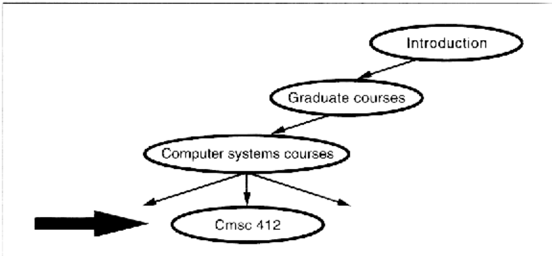
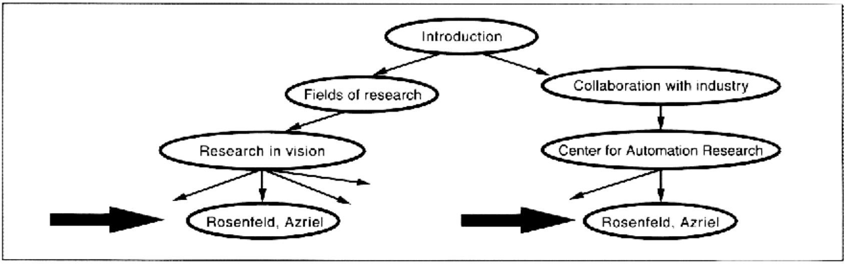


Figure 3. Asking: Where am I? when at the node "CMSC 412." Hierarchization of the CMSC hypertext. Only clusters on the path from the root are presented.

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

[8]Looking at a node and its immediate neighborhood, defined in [12] as local map.



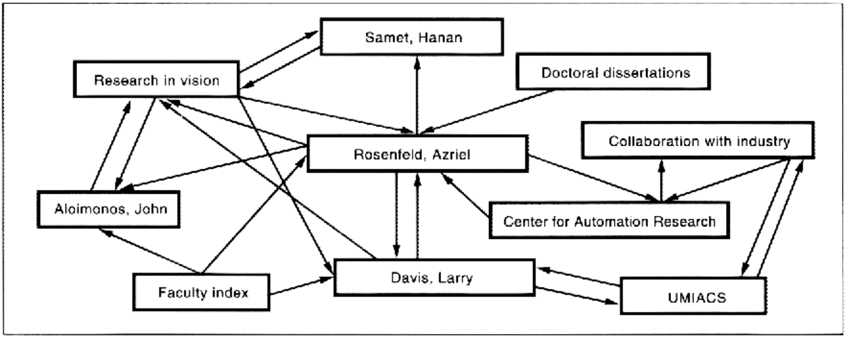


Figure 4. Asking: Where am I? when at the node "Rosenfeld Azriel." Two views that are equally justified.

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

Selecting different nodes as a root provides a different view of the same hypertext. For example, running the algorithm on GOVA resulted in 10 candidates for one root. Different perspectives resulted from taking a different root. 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 (Figure 6). When taking "Introduction" to be the root we find the node "Caesarea's harbor" under the cluster "Caesarea" which is connected to the root. This view results from the Smithsonian Institution exhibition 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. That is why it was found on level 4 (and of course it was not 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 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.[9] The algorithms are based on finding the biconnected components[10] and strongly connected components[11] 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 of breaking 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 application 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 [18] hypertext[12] (see Figure 9). Users in the node "Howard, Elman," for example, can see some different clusters that constitute the hypertext, and their semantic meaning (often the reader can conceive of a meaningful name by recognizing common themes or terms, but this is not guaranteed). A second application of the process to the cluster that contains the node "Howard, Elman" gives more information. Users can see the nodes "O'Leary, D." and "Stewart, G. W." create a subcluster within the four faculty members that constitute the main cluster (this is actually the

[9]A semantic cluster of 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 of the whole graph. This is a set of nodes appropriate for aggregation. See [3].

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

[11]The same definition but for directed graphs.

[12]"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 networks. It has 243 nodes and 803 links.

Figure 6. Two different answers being at the node "Caesarea"

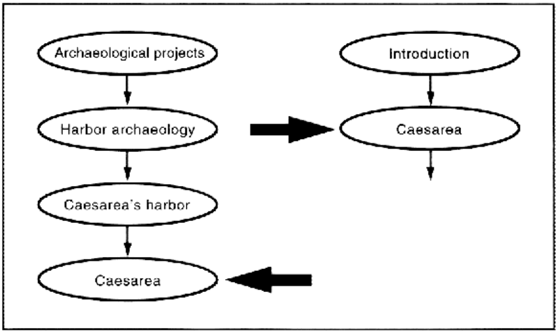
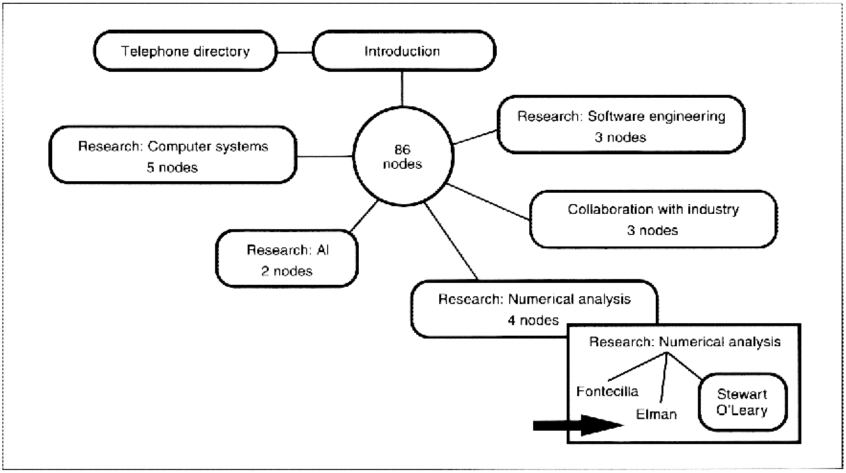


Figure 7. Part of the reduced graph of CMSC (similar to Nielsen global maps [12]). The user is in the node "Howard Elman," in the "Research in numerical analysis" cluster. The cluster is a reduced graph that contains two nodes and a subcluster.



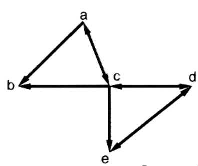
numerical analysis group). The subcluster suggests existence of some closer relationship between these two nodes: the only full professors in the numerical analysis group. This is an interesting piece of semantic information that is given by the structure. Breaking the hypertext into clusters, and being able to see their connections, is not only important, it might give users a notion of their position by using the different clusters as landmarks.

Where am I? Looking Around

Here, we present tools to give users useful information about their environment. The information is gained by defining and analyzing 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 [2].

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

Two interrelated 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 it cannot reach many of the other nodes in the hypertext. When a node has very low contrastatus it means it cannot be reached by many nodes. The *prestige* of a node is defined 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).



	a	b	c	d	e	Status	Prestige
a	0	1	1	2	2	6	5
b	inf	0	inf	inf	inf	0	–2
c	1	1	0	1	1	4	3
d	inf	inf	inf	0	1	1	–3
e	inf	inf	inf	1	0	1	–3
Constrastatus	1	2	1	4	4	12	

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

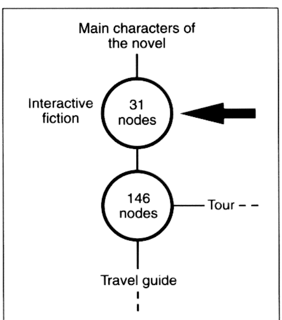


Figure 9. Part of the reduced graph of HHO. The location of the user is indicated by an arrow.

The local metrics can be used as guidance for readers who 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 hierarchization as in the following example. In "Hypertext Hands-On!" (HHO) [18] one part of the hypertext consists of a hypernovel, which can only be accessed through an article that introduces the novel. Since the nodes in the hypernovel can only reach a very limited subset of the hypertext they have low status, and since they can only be reached through the hypernovel introduction they have a high contrastatus. This suggests the use of aggregation instead of hierarchization (hierarchization might result in a very deep tree). 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 browsing through the "Interactive fiction" cluster, the analysis gives a crude notion of their whereabout in hyperspace.[13] We were pleased to find 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 [2]. 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[14]

$$CP = \frac{Max - \Sigma_i \Sigma_j C_{ij}}{Max - Min}$$

Where Max and Min are respectively the maximum and minimum value

[13]For some nodes in the interactive fiction cluster, the hierarchization algorithm gave depth 9. This is an example in which a local metric (depth) suggests using aggregation rather than hierarchization for a crude global view.

[14]This is a metric that reflects the concepts of sparse vs. rich hypertext as it is discussed in [13].

the converted distance can assume. C_{ij} is the converted distance between nodes i and j.[15]

[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.

Stratum is a measure that indicates whether a "natural" order for reading the hypertext exists. 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 defined as the absolute prestige of the hypertext divided by the linear absolute prestige. For example, in Figure 8 we get 14 for 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 $14/30 = 0.47$.

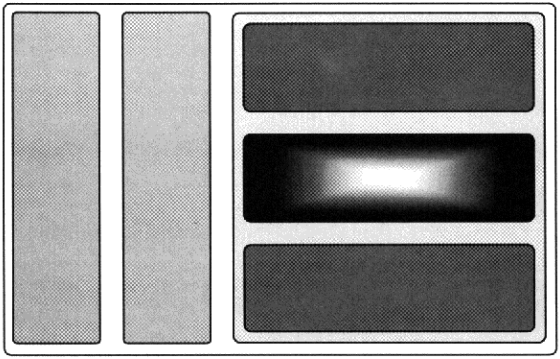
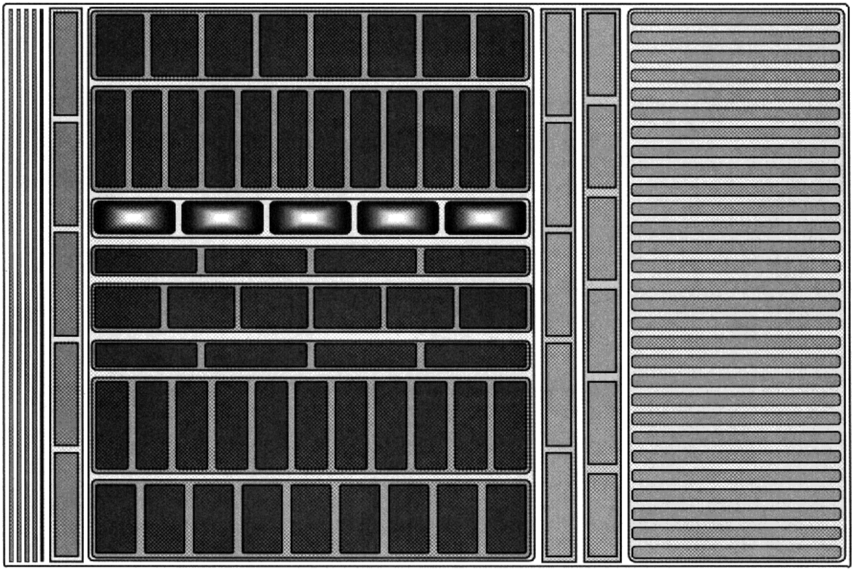


Figure 10. A tree map for a two-level deep tree. The root has three sons, and the last son has three children. The weight assigned to each node, and therefore the area, reflects the total number of its children and their descendants. The weight is used to determine the node bounding box. The user position is indicated by a black rectangle.

Figure 11. A tree map of the CMSC hypertext after the index and reference nodes were removed. The darkened rectangle presents the position of the user ("research in numerical analysis"). This particular node has five children, and it is a child of the node "fields of research." The node on the right side, that has 31 children is "undergraduate courses."



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 metrics to get useful information about the nature of the hyperspace: Is it dense? Is it close to linear (very few branches)? The stratum 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 (0.054 to 0.011). This reflects that HHO is a well-structured hypertext, while GOVA is not. Similar information can be derived from compactness measurements. These global metrics are useful from several points of view. First it is always an advantage to know what to expect. If we know, as readers, that we are going to read a very dense hypertext, we might consider the order of the nodes that are read more carefully. We will put some thought into choosing our path (more than we usually do). In other words we assume more responsibility.

Another scenario for which this kind of information is useful stems from a hypertext that is broken into clusters. In this case the metrics give information on the different clusters. Using the information we might consider another break of the hypertext (if it is very compact), or not (if we have high stratum). When we plan a path through different landmarks, this information is an element we might consider.

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 [9, 17]. This is a technique for showing large hierarchies (up to 5,000 nodes) compactly on the screen.

The main idea consists of dividing the screen first in vertical sections. Each section represents a file or directory in the top level. Then each of the vertical sections that is also a directory is divided in horizontal sections, which are further divided in vertical sections, and so forth. This process continues until all the files occupy some section of the screen. Color and size of each file depend on their attributes such as date of creation, type, and number of bytes. 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 Figure 10 a tree map for a simple tree (2 levels deep) is presented.

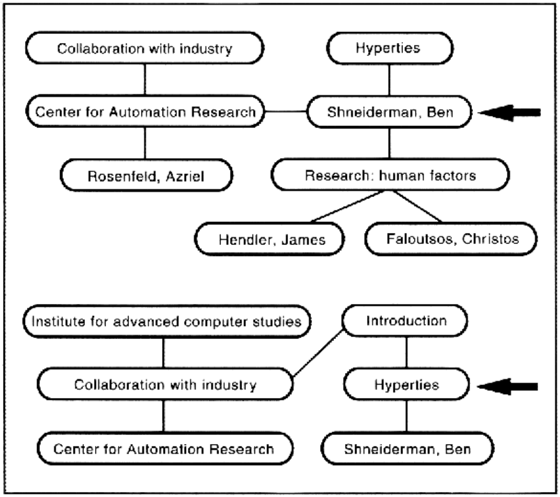
When showing a hypertext, some properties such as ROC, prestige, or the number of siblings can be used to determine the size of each displayed node. In Figure 11 a tree map of the CMSC database is presented. As a weight measure, the size of the subtree rooted at the node was picked. The color could be used as another dimension to indicate any other important property of the node. Properties like the length in words, the frequency of visiting by previous users, all the nodes containing the key word "Austria," the in or out degrees, the prestige, are possible candidates.

In our example the area of each node depends on the total number of children and their descendants. The tree map reveals that a significant part of the database is dedicated to "undergraduate courses." It should also be clear that the database is very shallow (only 3 levels deep). This surprising simplicity cannot be seen by showing a global map of the whole hypertext network, or by showing local maps, as can be seen from Figure 5.

The blackened node indicates the node in which the user is currently located. Most of the database (if we consider the text to be equally distributed among the nodes) is accessible from the nodes "undergraduate courses" and "fields of research." Using tree maps require some practice to gain familiarity, but users can get a clear picture of the database structure, and information about the more significant nodes.

Fisheye views. As we stated earlier, as the hypertext becomes larger, it is

Figure 12. A fisheye view of the CMSC hypertext in nodes "Shneiderman, Ben" and "Hyperties."



impractical to display the entire tree. A possible solution is to use fisheye views [5, 16]. Furnas [5] based his model on the observation that humans usually represent their own neighborhood with great detail, yet their representations further away are limited to major landmarks. Furnas suggests a "degree of interest" (DOI) function which assigns a value to each node in accordance with the degree to 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 is to take the ROC metric and use it as *API*. As mentioned earlier, the analysis was limited to structural features only. Better results can be achieved by combining the structural analysis with a stylistic and content analysis. This is another natural place to use semantic information in determining the *API*. Another option is to use the user behavior, such as the number of times a node was visited, or past uses of the hypertext by others.

In Figure 12 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. Users 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.

How do I Get to Any Destination?

Knowing where we are is a first step toward a natural goal of getting to a desired destination. Getting there at once (jumping via an index to the target node) is not difficult to achieve. Most hypertext systems give the option to move directly to a certain node (via search, or an index). Assuming 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. A structural analysis of the hypertext can suggest some alternatives in planning this path.[16]

Our toolbox gives users the option of imposing 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). 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 dense clusters). Figure 9 shows part of the reduced graph of the HHO hypertext, when users are somewhere in the "interactive fiction" cluster. For users who evaluate whether to go directly and browse through the "Travel guide" section, the reduced graph can help in making that decision. The separation might suggest 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. Alternatively, users can pick the landmarks by checking structural metrics like prestige, and plan the path through these nodes. To carry out this alternative 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 (e.g., content, stylistic) 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 hypertext the system can offer the novices a meaningful (from structural point of view) path between two nodes.

Conclusions

Based on experiences with building and reading dozens of hypertexts, plus our detailed structural analysis of three modest-sized hypertexts, we conjecture that 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 such questions as: Where am I? How do I get to any destination? What can be seen from here looking around? The crude positioning in the hyperspace was based on two processes: hierarchization and cluster identification. Several metrics were presented and used in the preceding processes as well as for locating landmarks and getting global information on the hypertext structure (e.g., compactness, stratum).

We have tested our algorithms on three modest-sized hypertexts, and believe that scaling up to large hypertexts is feasible. Our implementations were research prototypes and much work remains to design an integrated software application. We emphasize that the analysis, and the answers we get, reflect the input we are using—the structure. We cannot get more than what a structural analysis can give. The results could be much stronger if the analysis were extended to include content and the stylistic dimensions as well. This extension, the multiconstraints path planning, and the analysis of user satisfaction are directions for future research.

References

1. Akscyn, R.M., McCracken, D.L. and Yoder, E.A. KMS: A distributed hy-

permedia system for managing knowledge in organization. *Commun. ACM 31*, 7 (1988), 820–835.

2. Botafogo, R.A., Rivlin, E. and Shneiderman, B. Structural analysis of hypertext: Identifying hierarchies and useful metrics. *ACM Trans. Inf. Syst. 10*, 2 (1992), 142–180.
3. Botafogo, R.A. and Shneiderman, B. Identifying aggregates in hypertext structures. In *Proceedings of the Hypertext 91 Conference* (1991), pp. 63–74, Available from ACM, New York.
4. Charney, D. Comprehending nonlinear text: The role of discourse cues and reading strategies. In *Proceedings of the Hypertext '87 Conference* (1987), pp. 109–120. Available from ACM, New York.
5. Furnas, G.W. Generalized fisheye views. In *Proceedings of the CHI '86 Conference* (1986), pp. 16–23. Available from ACM, New York.
6. Halasz, F.G. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM 31*, 7 (1988), 836–852.
7. Hammond, N. and Allinson, L. Travels around a learning support environment: Rambling, orienteering or touring? In *Proceedings of the CHI '88 Conference* (1988), pp. 269–273. Available from ACM, New York.
8. Harary, F. Status and contrastatus. *Sociometry 22* (1959), 23–43.
9. Johnson, B. and Shneiderman, B. Tree maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization 91 Conference* (1991), pp. 284–291.
10. Marchionini, G. and Shneiderman, B. Finding facts and browsing knowledge in hypertext systems. *IEEE Comput. 21*, (1988), 70–80.
11. Marshall, C.C. Guided tours and online presentations: How authors make existing hypertext intelligible for readers. In *Proceedings of the Hypertext '89 Conference* (1989), pp. 15–26. Available from ACM, New York.
12. Nielsen, J. The art of navigating through hypertext. *Commun. ACM 33*, 3 (1990), 296–310.
13. Nielsen, J. *Hypertext and Hypermedia*. San Diego, Calif., Academic Press, 1990.
14. Norman, K.L. Personal communication, (1991).
15. Plaisant, C. Guide to opportunities in volunteer archaeology. Case study on the use of hypertext system in a museum exhibit. In *Hypertext/Hypermedia Handbook*, E. Berk, J. Devlin, Eds. McGraw-Hill, San Diego, Calif., 1991, pp. 498–505.
16. Sarkar, M. and Brown, M.H. Graphical fisheyeviews of graphs. In *Proceedings of the CHI '92 Conference* (1992), pp. 83–91.
17. Shneiderman, B. Tree visualization with Tree maps: A 2d space-filling approach. *ACM Trans. Graph. 11* (1992), 92–99.
18. Shneiderman, B. and Kearsley, G. *Hypertext Hands-On!*. Addison-Wesley, Reading, Mass., 1989.
19. van Dijk, T. and Kintsch, W. *Strategies of Discourse Comprehension*. Academic Press, New York, 1983.

CR Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval; H.5.1 [**Information Systems**]: Multimedia Information Systems

General Terms: Algorithms, design, human factors

Additional Key Words and Phrases: Clustering, hypertext, metrics navigation, structural analysis

About the Authors:

EHUD RIVLIN received his Ph.D. at the University of Maryland. He is an assistant professor at the Technion, Haifa, Israel. His research covers robot vision, image processing, human-computer interaction and hypertext. email: ehud@cvl.umd.edu

RODRIGO BOTAFOGO is currently working at the NEC Corp. in Kanagawa, Japan. His research interests include hypertext, human-computer interaction and information retrieval. email: botafogo@joke.cl.nec.co.jp

BEN SHNEIDERMAN is a professor in the department of Computer Science, head of the Human-Computer Interaction Laboratory, and member of the Institute for Systems Research, all at the University of Maryland at College Park. email: ben@cs.umd.edu

Authors' Present Address: Dept. of Computer Science, University of Maryland, College Park MD 20742