# Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics

RODRIGO A. BOTAFOGO, EHUD RIVLIN, and BEN SHNEIDERMAN
University of Maryland

Hypertext users often suffer from the "lost in hyperspace" problem: disorientation from too many jumps while traversing a complex network. One solution to this problem is improved authoring to create more comprehensible structures. This paper proposes several authoring tools, based on hypertext structure analysis.

In many hypertext systems authors are encouraged to create hierarchical structures, but when writing, the hierarchy is lost because of the inclusion of cross-reference links. The first part of this paper looks at ways of recovering lost hierarchies and finding new ones, offering authors different views of the same hypertext The second part helps authors by identifying properties of the hypertext document Multiple metrics are developed including *compactness* and *stratum*. Compactness indicates the intrinsic connectedness of the hypertext, and stratum reveals to what degree the hypertext is organized so that some nodes must be read before others.

Several existing hypertexts are used to illustrate the benefits of each technique. The collection of techniques provides a multifaceted view of the hypertext, which should allow authors to reduce undesired structural complexity and create documents that readers can traverse more easily.

Categories and Subject Descriptors: H.5 [**Information Systems**]: Information Interfaces and Presentation; H.3 0 [**Information Storage and Retrieval**]: General

General Terms: Design, Documentation, Human Factors, Measurement

Additional Key Words and Phrases: Graph theory, hierarchies, hypertext, metrics, networks, structural analysis

## 1. INTRODUCTION

Hypertext systems are used in many applications because of their flexible structure and the great browsing freedom they give to users. However, this same flexibility and freedom is the cause of a major concern: the "lost in hyperspace" problem. Previous attempts to solve this problem have concentrated in two areas: improving the user interface and textual analysis. The

R A. Botafogo was partially supported by the Brazilian National Research Council (CNPq) under grant 201008/88.2 and the NCR Corporation.

Authors' addresses: R A. Botafogo, NEC Corporation, Kanagawa, Japan; E. Rivlin and B. Shneiderman, Human-Computer Interaction Laboratory, Department of Computer Science, University of Maryland, College Park, MD 20742; email: ben@cs.umd.edu

main techniques used to improve the user interface are *multiple windows* [15], *maps* [1, 3, 7, 18] and *tours* or *path mechanisms* [30]. Unfortunately, these techniques have not scaled up effectively. Multiple windows help only in a localized way, maps lose their appeal when hypertexts have more than a few hundred nodes, and path mechanisms are hard to author and maintain.

Textual analysis, on the other hand, scales up much more nicely. By statistically analyzing word frequencies in each article, it is possible to index articles by significant terms. Readers can use those terms to facilitate retrieval and navigation among relevant documents [22]. Although it is possible to index automatically even very large document collections using textual analysis, this can only provide a partial solution for the "lost in hyperspace" problem. As discussed later, it is important for readers to grasp the interdocument structure, and textual analysis is of limited benefit in finding such linkages. Readers would benefit from information about relationships among articles (definition, example, details, map), clues about appropriate reading order, and so on.

User interface techniques and textual analysis do not make use of an important part of hypertexts: their structure. A hypertext is usually defined as a set of nodes that contain information, and as a set of links that connect interrelated nodes. To be understandable, a hypertext must be carefully crafted; each superfluous or missing link contributes to user disorientation. A link between two nodes is not only glue that keeps the hypertext together, but each link represents a semantic relationship between the nodes.

Authoring hypertexts is not an easy task. Although authors are improving their skills, as hypertexts grow, management becomes extremely difficult even for the most qualified professionals. So software tools that would help hypertext authors see global and local properties would be beneficial. Similarly, metrics that would indicate the hypertext complexity or special properties of nodes would be welcome.

In this paper, hypertext structure is analyzed in two ways: first, to provide authors with different views of the hypertext; and second, useful metrics are devised to reflect properties of nodes and of the whole structure. Improvements in the user interface and textual and structural analyses, each provide solutions to different problems. Integrating all three techniques will help to overcome the "lost in hyperspace" problem. One caution is that while metrics can be useful in making precise and objective comparisons, they are only approximations to interesting concepts. Good judgment by authors is more important than staying within strict numerical limits.

Section 2 introduced preliminary concepts that are important for the development of the algorithms and metrics in Sections 3 and 4. Section 3 deals with hierarchical structures, showing that it is possible to automatically differentiate organizational and referential links.[1] The former are used to create a hierarchy while the latter are used to cross-reference information.

[1] In this paper organizational and referential links are called, respectively, hierarchical and cross-reference links.

By changing, in a controlled way, organizational to referential links, and vice-versa, authors get different views of the same hypertext. Metrics are introduced that identify nodes that can be used to generate specialized maps. Section 4 develops metrics that indicate global and local properties of the hypertext. These metrics do not indicate if the hypertext is "good" or "bad," but only provide feedback to authors. With this information authors might decide to add or remove links, add or remove nodes, or change node positions in the hypertext such that the hypertext attains certain desired properties. Finally, Section 5 presents the conclusion and directions for future work.

We concentrate on hypertext systems where nodes and links are untyped and can be represented by a directed graph. The hypertexts used for testing were created in Hyperties.[2] However, we believe that the ideas extrapolate to hypertext systems with more complex structures. Our software tools were developed for research purposes, with the hope of inspiring commercial implementations.

## 2. PRELIMINARIES

### 2 1 The Converted Distance Matrix (CDM)

Central members of an organization are those that can influence a great number of people in it. Using a directed graph it is possible to represent the influence relations in the organization. For instance, a directed link between two nodes indicates that the first node has a direct influence over the second. If there is a path (of length $> 1$) between two nodes, then we can say that the first node in the path has an indirect influence (except for the second node on which it has a direct influence) over all the other members in the path.

Figure 1a shows a graph of what might be a small social organization. Clearly, person "e" is the most central member in this group, since she/he is the only one that has direct influence over all other members of the organization. Figure 1b shows a similar organization; however, none of its members can influence all the others. How can we define which person is the most central? Clearly, either "e" or "b" are most central since they influence the greatest number of other members. But while node "e" has direct influence over 3 nodes, node "b" has direct influence over 2 nodes and indirect influence over the other 2 nodes.

To formalize the notion of centrality, the sum of distances from a node to all other nodes in the organization is used. A matrix that has as its entries the distances of every node to every other node is called the *distance matrix* of the graph. Unfortunately, when a node does not reach another node in the hypertext, the entry in the distance matrix is infinite. Operating with infinite values is not convenient. Figure 2 shows the distance matrix of the graph in Figure 1b. Because of the many infinite entries in it, it is not easy to distinguish which node is more central.

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

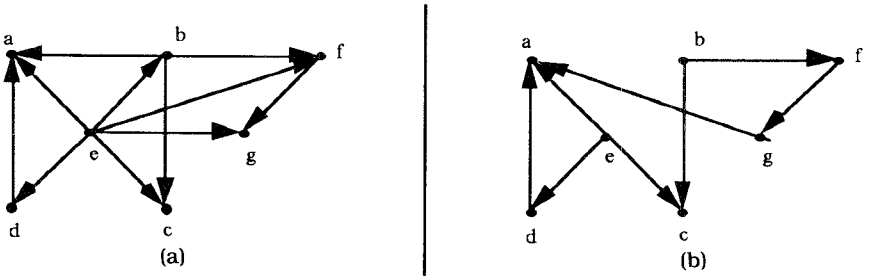



Fig. 1. (a) Person "e" is the most central since he/she has direct influence over all the other members in this organization. (b) Which is more central: node "e" or "b"?

M =

| | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| a | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| b | 3 | 0 | 1 | ∞ | ∞ | 1 | 2 |
| c | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| d | 1 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ |
| e | 1 | ∞ | 1 | 1 | 0 | ∞ | ∞ |
| f | 2 | ∞ | ∞ | ∞ | ∞ | 0 | 1 |
| g | 1 | ∞ | ∞ | ∞ | ∞ | ∞ | 0 |

Fig. 2. Distance matrix for the graph in Figure 1b.

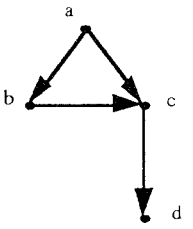
To solve the problem just described, we define the converted distance matrix as follows: Let $C$ be the converted distance matrix and $M$ the distance matrix, then

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

Where the choice of $K$, a finite *conversion constant*, strongly influences the centrality of nodes. Figure 3 shows a graph with the corresponding distance matrix and converted distance matrix. In this example the value of $K$ has been set to 4. Guidelines for choosing $K$ are discussed in a later section.

## 2.2 Index and Reference Nodes

Intuitively, *index* nodes are nodes that can be used as an index or guide to many other nodes. For example, an article that points to all the other articles in a hypertext is certainly an index node. In a hypertext about a computer science department, an article pointing to all the professors that work in human factors is also an index. *Reference* nodes are in a certain way the converse of index nodes. For example, in a hypertext about animals, all birds will link to an article about "feathers," dogs to an article about "sharp teeth," and so forth. It is possible to make reference to the birds by saying that they are animals that have feathers. Since index and reference nodes




Distance Matrix

| | a | b | c | d |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 |
| b | ∞ | 0 | 1 | 2 |
| c | ∞ | ∞ | 0 | 1 |
| d | ∞ | ∞ | ∞ | 0 |

Converted Distance Matrix

| | a | b | c | d |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 |
| b | 4 | 0 | 1 | 2 |
| c | 4 | 4 | 0 | 1 |
| d | 4 | 4 | 4 | 0 |

Fig 3. Graph with its "distance matrix" and "converted distance matrix" ($K = 4$).

usually point to or are pointed to by whole classes of nodes, it is natural to define them as a function of their out- and in-degrees, respectively.

*Definitions.* For a given hypertext:

—Let $\mu$ be the mean of the outdegrees of the nodes in the hypertext and let $\mu'$ be the mean of the indegrees of the nodes in the hypertext. Note that $\mu = \mu'$, since every link that leaves a node has to reach another node. For this reason we use $\mu$ for both means.

—Let $\sigma$ be the standard deviation of the outdegrees of the nodes.

—Let $\sigma'$ be the standard deviation of the indegrees of the nodes.

—Let $\tau$ be a threshold value.

—An index node is a node whose outdegree is greater than $\mu + \tau$.

—A reference node is a node whose indegree is greater than $\mu + \tau$.

In the examples that follow, $\tau$ is given the value of $3 * \sigma$. This value was heuristically chosen to be high, so that only a small number of nodes will be identified as index and reference nodes.

### 2.3 Centrality Metrics

The idea of node centrality is not new [12, 13]; however, because of the problems with infinities, the work cited above could not be used directly with hypertexts. This section extends their results.

*Definitions.* The converted out distance (COD) for a node $i$ is the sum of all entries in row $i$ in the converted distance matrix (C). Formally,

$$\mathrm{COD}_i = \sum_j C_{ij}.$$

Similarly, the converted in distance (CID) for a node $i$ is the sum of all entries in column $i$ in the converted distance matrix. Formally,

$$\mathrm{CID}_i = \sum_j C_{ji}.$$

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

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

It is natural to define a central node as one whose distance to all the other nodes in the hypertext is small. As that distance grows, nodes become less central. Consequently, the smaller the COD the more central the node. For a single hypertext, the COD is a good indication of the node centrality as compared with another node, but this number indicates little when two different hypertexts are compared. For instance, a node with COD 200 in a hypertext with 1000 nodes might be much more central than a node with COD 50 in a hypertext with 100 nodes. To try to compensate for this difference, the *Relative Out Centrality* (ROC) metric for a node $i$ is defined as

$$\mathrm{ROC}_i = \mathrm{CD}/\mathrm{COD}_i.$$

The higher the ROC metric of a node, the more central (the inverse of the COD). Observe also that the ROC is normalized in relation to the size of the hypertext (CD), making it more convenient for comparisons between hypertexts.

The *relative in centrality* (RIC) metric is defined analogously as

$$\mathrm{RIC}_i = \mathrm{CD}/\mathrm{CID}_i.$$

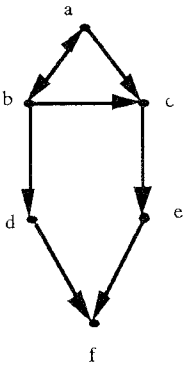
Figure 4 shows a graph with the associated converted distance matrix and its relative out-centrality and in-centrality metric. In this example, the value of $K$ (the conversion constant) is 6, equal to the number of nodes in the hypertext.

## 3. HIERARCHIES

One common way of organizing information, and even people, is through the use of hierarchies. Many firms and social groups are organized hierarchically. A newspaper is usually separated in sections, for example, "Local," "National," "International." The "Local" section would have an "Arts" subsection, which would be further subdivided in "Theater," "Music," and so on. Textbooks, user manuals, music catalogs are also usually written in a hierarchical way. Of course, hierarchies are not the only way of organizing, but their use is extensive.

Many hypertext systems suggest or even induce their authors to write hierarchically [5, 6, 25, 26]. NoteCards [9] requires users to file every node in a hierarchical card; but Halasz [10] observed that this compels users to decide on a structure too early in the authoring process, and should be relaxed in future versions. Charney cites extensive research that suggests that readers form mental hierarchical representations of texts [4, 29].

In Hyperties, authors are free to create any structure. A new article can be created at any time and linked to any other article in the hypertext. Authors do not need to decide if a link is hierarchical or cross-referential. Hyperties gives authors a high degree of freedom, however, the price is that users are




| | a | b | c | d | e | f | COD | ROC |
|---|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 | 2 | 3 | 9 | 10 2 |
| b | 1 | 0 | 1 | 1 | 2 | 2 | 7 | 13 1 |
| c | 6 | 6 | 0 | 6 | 1 | 2 | 21 | 4.3 |
| d | 6 | 6 | 6 | 0 | 6 | 1 | 25 | 3 7 |
| e | 6 | 6 | 6 | 6 | 0 | 1 | 25 | 3 7 |
| f | 6 | 6 | 6 | 6 | 6 | 0 | 30 | 3 1 |
| CID | 25 | 25 | 20 | 21 | 17 | 9 | 92 | |
| RIC | 3 7 | 3.7 | 4.6 | 4.4 | 5.4 | 10.2 | | |

Fig 4 Graph with its "converted distance matrix" and associated metrics

not guaranteed a clear structure. We seem to be in a vicious circle. Forcing the writer to decide on a structure early in the authoring process is bad, as suggested by Halasz. Waiting until the hypertext is complete or almost complete might be too late, since the author may not be able to see a clear structure.

This section tries to break this vicious circle. By analyzing the hypertext structure—remember that the structure does reflect semantic information—we will find reasonable hierarchies automatically and will separate hierarchical from cross-referential links. By having automatic help in the formation of the structure, authors gain another advantage: Trigg [27] observed that an important function of hypertext systems is to allow authors to create many coexisting organizations of the same hypertext which can evolve in parallel. The techniques presented in this section find several possible hierarchical structures. Each of the possible hierarchies present the hypertext in a different view. Neuwirth [19] observed that the ability to view knowledge from different perspectives is important for creative writing.

### 3.1 Finding Hierarchies

To find a hierarchy in a hypertext, two tasks must be performed: first the root must be identified and then hierarchical and cross-reference links must be distinguished. In hypertexts, some properties are necessary (although not always sufficient) for a node to be a root. The fundamental property of a root is that it has to reach every, or almost every, node in the hypertext. If there is no path from the root to a node, there is no way to navigate through the hypertext to reach this node. A node that cannot be reached is either very special (such as some indices) or there are missing links in the hypertext. Another important property of a root is that the distance from it to any other node should not be too large. If the distance from the root to a node is very large, there will again be a navigation problem. Users will have to go through a long and maybe tedious path before reaching the desired information. Finally, the root should have a "reasonable" number of children. These

requirements are only guidelines, and as with any guideline it is possible to create a hypertext that violates it and is still understandable and well structured. Many books violate grammatical, orthographical, and punctuation rules and become masterpieces of literature.

The hypertext root is highly text and task dependent. Since this paper is not concerned with the integration of textual and structural analysis, we will be satisfied in finding a set of nodes that comply with the requirements above. Those nodes will then be presented to authors who will select the ones that most satisfy their needs. Section 2.3 introduced the notion of the relative out centrality (ROC) of a node. Nodes that have a high ROC satisfy the first two properties of roots; that is, they can reach all (or almost all) the nodes in the hypertext and the distance from them to all the other nodes is not very large. The third property, i.e., that the root should have a reasonable number of children, can easily be met by counting the links of each node. Index nodes (Section 2.2) are by definition nodes that have numerous children and should not be roots. The process of identifying good roots for the hypertext consists of the following:

(a) identifying the nodes that have high relative out centrality metric, and
(b) removing the index nodes.

When authors want the root to be an index, they would interact with the system in order to prevent the removal of these nodes as roots.

For example, the Hyperties CMSC document contains information about professors and courses at the Computer Science Department of the University of Maryland. Figure 5 shows a histogram of the ROC metric of all the nodes in this hypertext. Observe that the "Introduction" and the "Contents" are nodes whose ROC metric is much bigger than the ROC metric of all the other nodes. This indicates that both can be used as roots. The "Introduction" is actually the first article users see when browsing this hypertext. On the other hand, "Contents" is an index node and is removed from the list of possible roots. "Fields of Research" indicated in this picture is among the 10 nodes with highest ROC, and in Section 3.2.1 the hypertext is analyzed with this node as the root.

In this example the value of $K$ was set equal to the number of nodes in the hypertext. This value of $K$ guarantees that $K$ is bigger than any finite entry in the distance matrix, since the maximum distance between two nodes in a hypertext with $n$ nodes is $(n - 1)$. Depending on the author's intention, different values can be assigned to $K$. In Figure 6a we chose the value of $K$ to be equal to $2 * n$ ($n$ is the number of nodes in the hypertext). This strengthens the requirement that, to be central, a node needs to reach all the other nodes. In this figure, node "b" is clearly the most central (ROC = 11.0). Suppose now that this hypertext is still in an authoring phase and that many links are still to be added. Making $K$ have a smaller value than before allows nodes that do not reach all other nodes to be central. In the graph of Figure 6 node "a" reaches all the nodes, but "b," in one step. This suggests that "a" should be central and that the link "ab" might have been forgotten. When $K = n$ (Figure 6b), node "a" is more central than node "b."

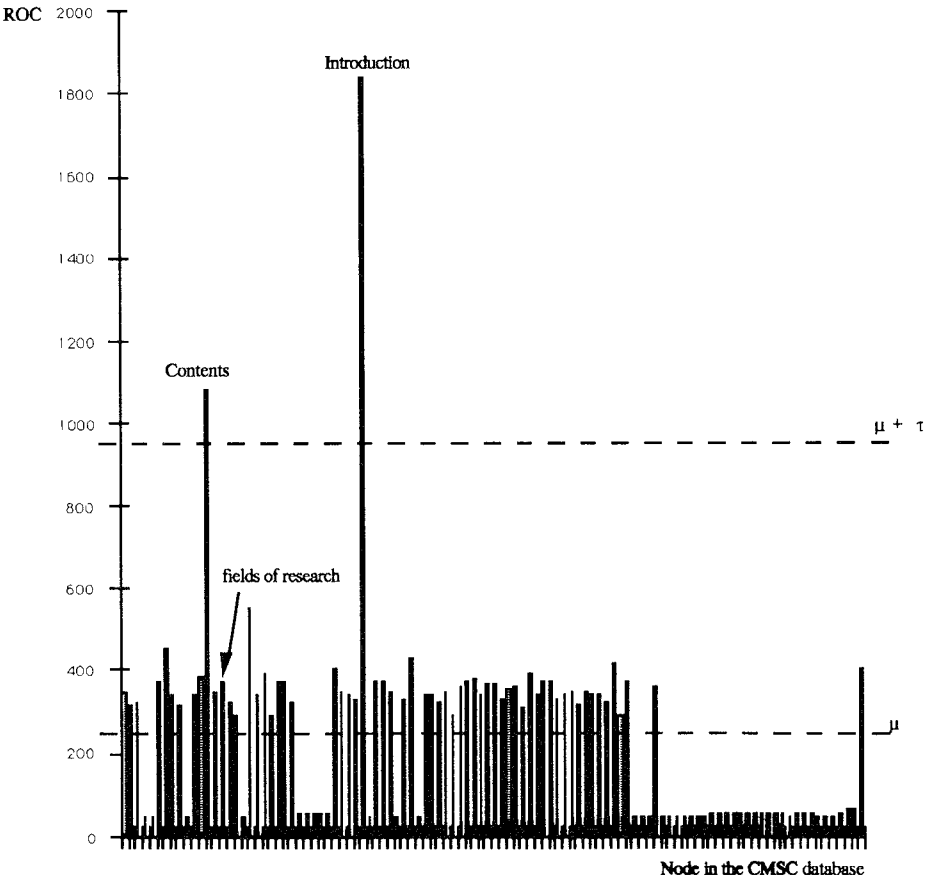
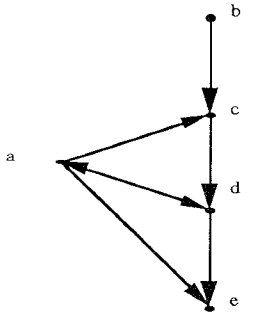



Fig 5 Histogram of the ROC metric for the CMSC hypertext The threshold for the ROC was set analogously to outdegree to the mean plus three times the standard deviation ($\mu + \tau$)




| | a | b | c | d | e | ROC |
|---|---|---|---|---|---|---|
| a | 0 | 10 | 1 | 1 | 1 | 7 6 |
| b | 3 | 0 | 1 | 2 | 3 | 11.0 |
| c | 2 | 10 | 0 | 1 | 2 | 6 6 |
| d | 1 | 10 | 10 | 0 | 1 | 4.5 |
| e | 10 | 10 | 10 | 10 | 0 | 2.5 |

(a)

| | a | b | c | d | e | ROC |
|---|---|---|---|---|---|---|
| a | 0 | 5 | 1 | 1 | 1 | 7 3 |
| b | 3 | 0 | 1 | 2 | 3 | 6 4 |
| c | 2 | 5 | 0 | 1 | 2 | 6 4 |
| d | 1 | 5 | 5 | 0 | 1 | 4 8 |
| e | 5 | 5 | 5 | 5 | 0 | 2 9 |

(b)

Fig. 6. Two converted distance matrices for the same graph (a) $K = 10$ $(2 * n)$. (b) $K = 5$ $(n)$

Once the root has been identified, a very simple physical metaphor can be used in order to separate hierarchical from cross-reference links. Assume that vertices are flexible joints and edges are strings of constant length. Picking up the graph at the root and letting the rest dangle will leave only a tree with the shortest paths taut. Other links are cross references. This strategy is equivalent to finding a spanning tree by a breadth-first search.

Figure 7 shows the "hierarchization" of a graph where plain arrows are hierarchical links and dashed arrows are cross-reference links. Note that links "kd," "ib," and "hm" are cross-reference, all the others are hierarchical. Note also that node "f" appears twice in the hierarchical form of the hypertext: once as a child of node "b" and once as a child of node "c." In both cases there are no structural clues as to which position is better for node "f." This is a case where textual analysis could be used to remove the ambiguity. For example, one could compute the distances between nodes "f" and "b," and "f" and "c" in a vector space (where each dimension of the space is given by the terms that index the articles [22]). The node closer to "f" would be its parent. This algorithm is appealing, but it is not guaranteed to produce the most readable structure. Improvements should be sought and authors must exercise personal judgment.

## 3.2 Showing Hierarchies

Once a hierarchy has been found, it is important to show it to authors in a reasonable way. One alternative is to display the entire generated tree. Unfortunately, this is usually not convenient for a hypertext with more than 100 nodes. This section discusses techniques that can be used to facilitate the visualization of a hypertext.

3.2.1 *Maps and Outlines.* When browsing Nielsen's HyperCard stack on *Hypertext'87 Trip Report* [17], the reader is always presented with two maps: an *overview map* and a *local map*. The overview map, which never changes, is a picture of the hypertext root surrounded by its children. The local map, created for each node as the user reads, shows the current node in the center surrounded by its children. Nielsen's maps were generated manually. However, once a hierarchy has been identified as described in the previous sections, it is not difficult to create maps automatically.

Figure 8 shows a global and local map of the CMSC hypertext with the introduction taken as the root. Figure 8b shows a local map when the node being read is "412: Operating systems." Highlighted nodes in the pictures indicate in the local and overview map respectively the node being read and the higher ancestor of this node that is not the root. By looking at those maps, one can see that "412: Operating systems" is a course in computer systems at a graduate level. It should be clear that those are only two of the many possible maps that can be generated automatically (we show the two maps to follow Nielsen's approach). For instance, a map having "412: Operating systems" in the center, surrounded by all its children, might be better when trying to find what next node to visit.

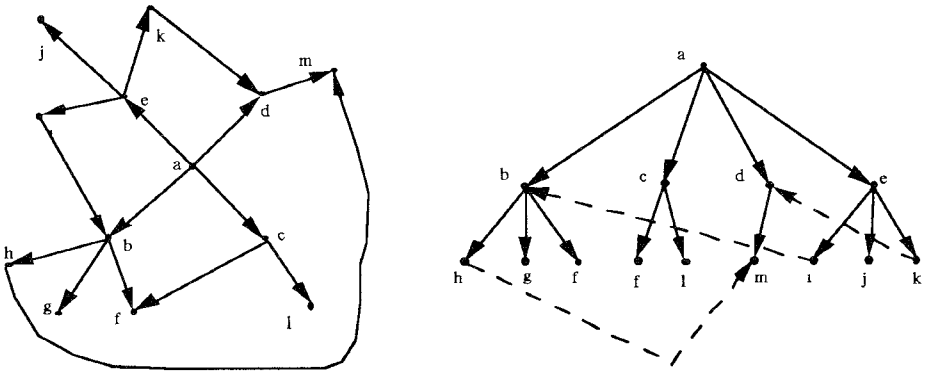



Fig 7 "Hierarchization" of the graph. Dashed lines indicate cross-reference links, while plain lines are hierarchical links

Another way of showing the hierarchy consists of breaking the hypertext into chapters, as in a book. Knowing what chapter a node belongs to and what the other members of this chapter are helps by inserting the node into a context. To break the hypertext into chapters, we assume that the introductory article is the table of contents for the hypertext. We then assume that each link in the introduction is a pointer to one of the chapters.[3] After the "hierarchization" algorithm has been applied, each chapter will be the root of a subtree of the hypertext. We say that a node *belongs to* (*is in*) a chapter if it is in the subtree rooted in that chapter.

Figure 9 shows the chapters in the CMSC hypertext when "Introduction" was taken as root. Numbers before each node indicate their level in the hierarchized hypertext tree. A node at level 4, for example, will be at distance 3 from the root, since the root is at level 1. The node "Introduction" does not appear in this figure because none of the chapters point back to it. In this figure most of the professors appear in two chapters: "building directory" and "faculty member index," and their distance from the root is only 3. Also, all the courses are directly pointed to by chapter "undergraduate courses (upper level)." This suggests that the hypertext author wants readers to be able to access any professor or course easily, independent of the research areas.

Another interesting observation is that while "undergraduate courses (upper level)" (UL) is a chapter, the article "undergraduate courses (lower level)" (LL) is not (it belongs to UL). For this reason all the courses (upper and lower level) belong to UL. This is certainly not desirable and suggests that a link is missing. When a link between the introduction and LL is added to the hypertext, Figure 10 is obtained. Now both UL and LL are chapters, and the lower level courses appear in both those chapters. Although this result is better, it is still not right. By browsing the hypertext, one can see

[3] Henceforth, the word "chapter" indicates a node pointed by the introduction.

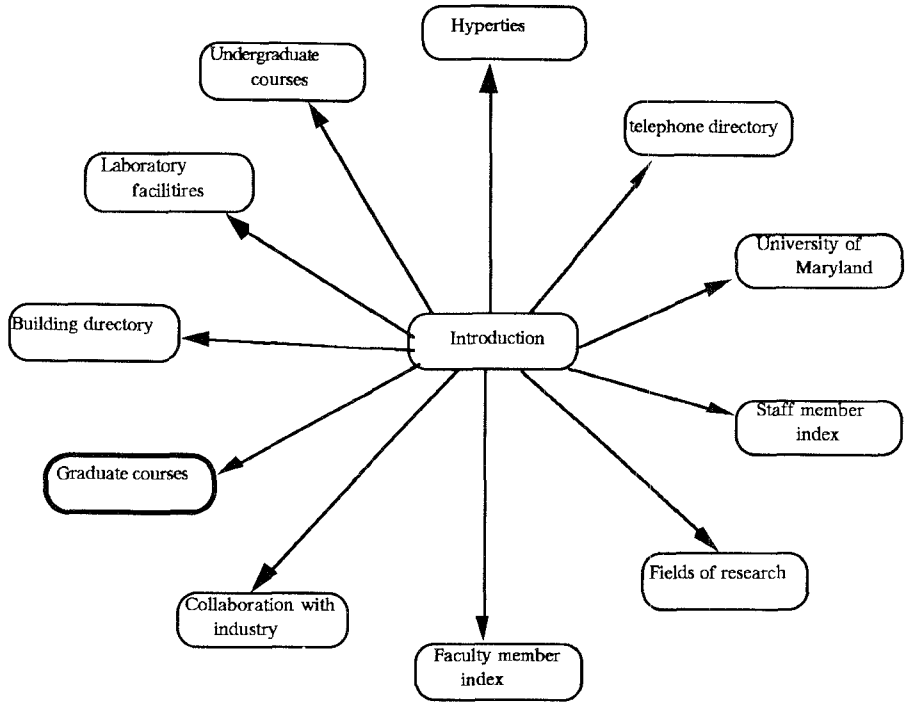



Fig. 8. (a) Overview map of the CMSC hypertext Highlighted node shows path followed by user. (b) Local map of the CMSC hypertext. User is reading node "412: Operating systems "


Fig. 9. CMSC hypertext with "introduction" as the root.

that UL is actually a list of all the courses in the department, and consequently that the name UL was not chosen properly. What action should be taken now depends on the author. In a new version of the CMSC hypertext, changes were made in UL so that it links only to upper level classes, fixing the problem described above.

Figure 11 is a view of the same hypertext, but this time "Fields of research" was taken as the root, since it is one of the 10 articles that have a high ROC metric. Furthermore, when index and reference nodes are removed from the hypertext, it becomes the one with the highest ROC metric (the


Fig. 10. Lower level courses in the CMSC hypertext appear under the proper chapter "undergraduate courses (lower level)."

"Introduction" is an index node). The view from the hypertext now is quite different from the previous one. Each professor is now under the chapter of his or her field of interest. Which of the two organizations to present should be left to the author or the reader. For instance, a reader interested in the structure of courses should probably see the hypertext as in Figure 9, while a reader interested in the fields of research at the University of Maryland should see the hypertext as in Figure 11.

Looking at the hypertext from this new view reveals other interesting properties. For example, in the original view (the one having as root the "Introduction" article) the node "Hyperties" is directly connected to the root, while in the new view it is at level four. This change can be explained by looking at the author's objective when creating this hypertext.

It is important to give users information about the system they are using (the CMSC hypertext was written with Hyperties) as soon as they start reading, which explains why in the original view (the one actually shown to readers) this node is at level 2 (Figure 12). On the new view, however, the main focus is the research fields in the department, and there is no reason for Hyperties to have a prominent place, since it is just another research topic. For the same reason, the view of node "Shneiderman, Ben" is quite different. In the original view he is alone in the chapter "Hyperties," because he heads the laboratory that developed Hyperties. In the new view, he is just another professor in the department, and is put in a chapter together with other


Fig. 11. CMSC hypertext with "fields of research" as the root

professors in the human factors field. Observe also in Figure 12 that in the original view "Shneiderman" is under the node "Hyperties," while in the new view "Hyperties" is the child—indicating that their relative importance is dependent on the objective of the hypertext.

3.2.2 *The Fisheye Lens Model.* Furnas [8] devised the fisheye lens model as another way of showing maps. This model is based on the observation that humans usually show their neighborhood with great care and detail, while more distant areas are fuzzier, as determined by the degree of interest.

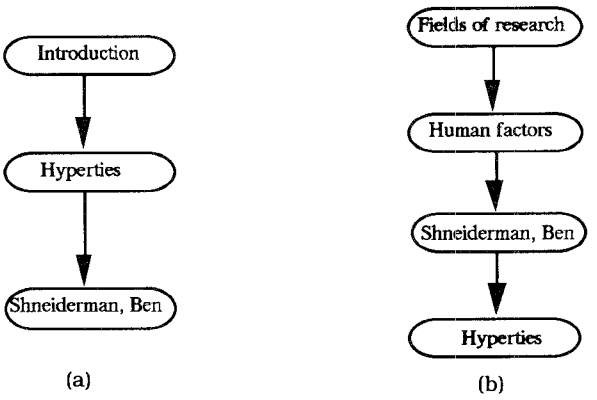



Fig. 12. Two different views of nodes "Hyperties" and "Shneiderman, Ben."

However, even distant sites may stand out if they are marked with some kind of distinguishing *landmark*. Unfortunately, finding nodes that are good landmarks is not a trivial task.

Valdez and Chignell [28] "anticipated that landmarks would tend to be connected to more objects than nonlandmarks, in the same way that major hubs serve as landmarks in airline systems." While running some experiments they observed a high correlation between the recall of words in a hypertext (how easily those words were remembered by subjects) and their *second-order connectedness*. The second-order connectedness is defined as the number of nodes that can be reached by a node when following at most two links. Nodes that have a high second-order connectedness will probably also have a high ROC metric, although the converse is not necessarily so. This suggests that nodes that have high ROC will be good landmarks. For example, in the CMSC hypertext, nodes such as "Introduction," "Contents," and "Fields of research" have very high ROC metric and are very important nodes, which indicates that they are good landmarks.

Following their analogy about landmarks, Valdez and Chignell search for major hubs in the hypertext; but they only look at nodes that can reach other nodes easily (high second-order connectedness). Since hypertexts are directed graphs, it is possible to extend their idea and postulate that nodes that have high *back-second-order connectedness* are also good landmarks. The back-second-order connectedness of a node can be defined similarly to the second-order connectedness of a node, as being the number of nodes that can reach the specified node in at most two steps. Nodes that have high back-second-order connectedness will also have a high *relative in-centrality* (RIC) metric (defined in Section 2.3).

Figure 13 shows the RIC metric for the CMSC hypertext. Two nodes "Cmsc 112" and "Cmsc 113" have a very high RIC metric as compared with the rest of the hypertext. "Cmsc 112" and "Cmsc 113" are articles about introductory courses in a computer science department. Since those two courses are fundamental in the curriculum, they are easily accessible by almost all other

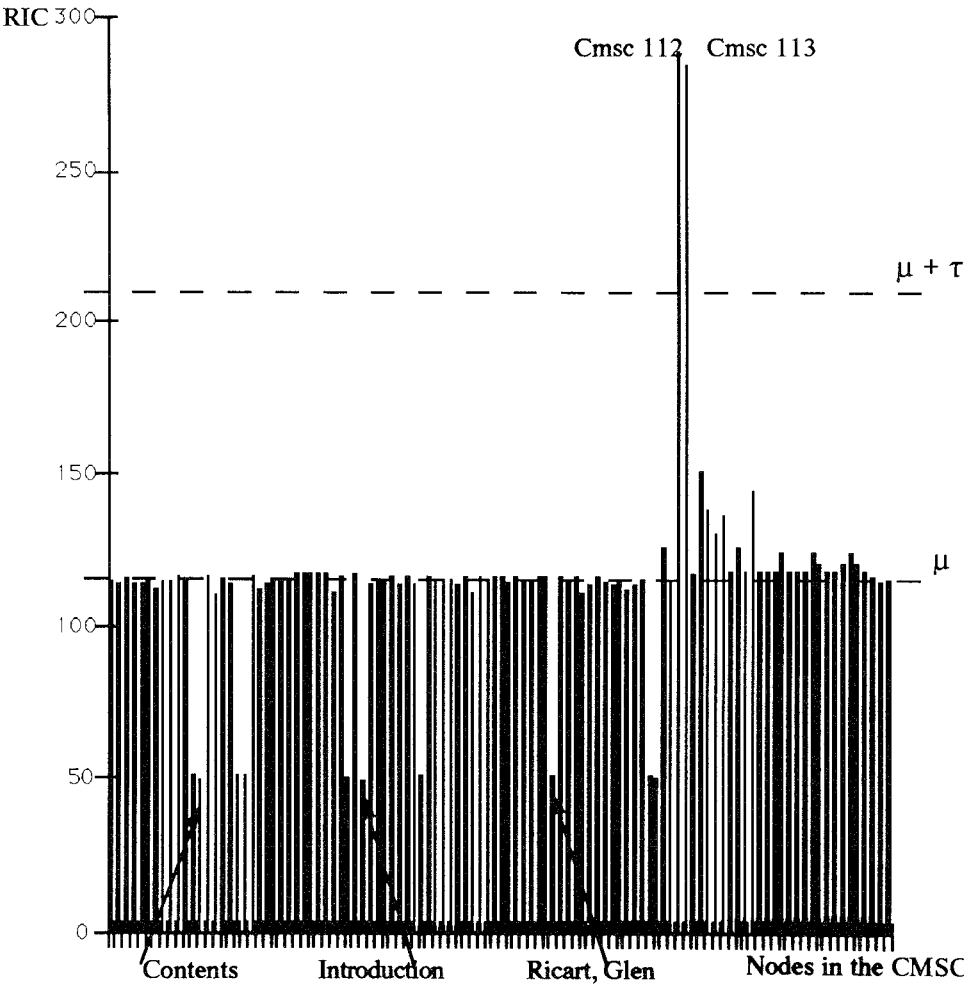



Fig. 13. Histogram of the RIC metric for the CMSC hypertext.

courses in the hypertext, and thus are very in-central. These two nodes are clearly important landmarks in the hypertext. On the other hand, the "introduction" and the "content" have a very low RIC metric, since they are not easily reachable by any of the other nodes in the hypertext. It is possible to expand on Valdez and Chignell's idea by identifying two types of landmarks: *out-landmarks* and *in-landmarks*.

Another interesting property seen in Figure 13 can be exemplified by the node "Professor Glen Ricart." Being the director of the computer center means that he does not participate in projects and has no office in the computer science department. His low participation in the department is reflected by his infrequent appearance in the hypertext and his low RIC metric.

## 4. METRICS

Comparing and imitating the work of others is one way to learn and improve. Metrics are a way of giving clear and precise values to objects being studied so that subjectivity is not involved. For example, in the study of texts, metrics such as length of words, length of sentences, and the use of passive voice are used. These metrics can be further combined to form a *readability* metric—a value that indicates how easily one can read the text. With precise values defined, it becomes possible to compare texts, certify that the texts have certain desired properties, and supervise the evolution of a given text. For example, software packages can help writers by identifying possible problems. Too much use of the passive voice usually makes reading difficult; consequently, articles that use the passive voice frequently have a bad readability metric. When authors see a bad readability metric and recognize that the problem comes from excessive use of the passive, they can modify their text and observe how the readability metric improves.

Hypertext authors can profit from readability metrics, but the metrics can only be applied to individual nodes. Making every node readable does not, however, guarantee that the whole hypertext is clear. Users can still get lost and not find the information they seek. New types of metrics that consider the hypertext structure and how the information in one node relates to the information in the others are necessary.

Two types of metrics are developed: *global* and *node*. Global metrics are concerned with the hypertext as a whole, while node metrics focus on the structural properties of individual nodes, i.e., how each node relates to the others. Both metrics should be used during hypertext development. Some node metrics can indicate problems such as missing links that are hard to identify. Global metrics can identify strangely organized or very complex hypertexts.

Characterizing structures by metrics is difficult. Structural metrics for readability can be satisfied even by hard-to-read hypertexts, just as readability metrics can be met by poorly written texts. The metrics in this paper are a first attempt to provide guidance for authors, and more experience is necessary for testing and refinement. The metrics presented here do not make use of some known mathematical quantities such as cutwidth and $k$-connectivity. Interested readers are directed to Botafogo [1] and Botafogo and Shneiderman [2] where those quantities are used to obtain abstraction in hypertexts.

### 4.1 Global Metrics

One of the most obvious global metrics of interest is the size of the hypertext. Since hypertexts are composed of two main components, nodes and links, it is helpful for the author to know the number of nodes and links in the hypertext. Many systems do provide the first piece of information, but few consider the number of links. With that information, the author can start to have an idea of the complexity and connectedness of the hypertext. However, the number of nodes and links and possibly their ratio gives only a rough

idea of the complexity of reading the hypertext In this section we develop two metrics, *compactness* and *stratum*, and try to capture some notions of complexity and connectedness in hypertexts.

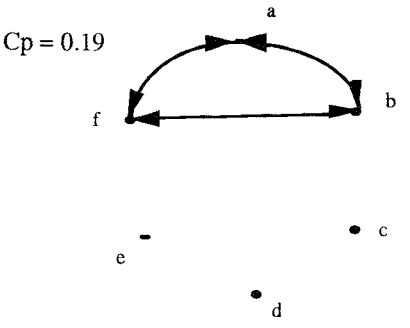
From a reader's point of view, a high compactness indicates that each node can easily reach any other node in the hypertext, suggesting a large amount of cross-referencing. This might be intended, but it also might indicate a poorly structured hypertext that can burden readers and lead to disorientation. For example, a hypertext that is fully connected is equivalent to a hypertext that is fully disconnected but where the user can access any node by using an index. In a fully connected hypertext readers have no clue as to which article should be read next, which is equivalent to choosing from a general index. On the other hand, a low compactness may indicate insufficient links and that parts of the hypertext are possibly disconnected. Since the metric does not take into account other hypertext facilities such as back-up across a link or string search, the hypertext might be easily browsable; however, the metric will help identify the structural aspects. A hypertext that requires use of back-up or string search to reach nodes is certainly different from one that does not. Metrics are an aid to authors who must make the final decision.

Stratum is a metric that suggests whether there is an order for reading the hypertext. For instance, a linear hypertext can only be read one way. On the other hand, if the hypertext is a cycle then structurally it does not make any difference from what node reading starts. If the hypertext is a binary tree, the reading must start from the root. From there either the root of the right or the left subtree must be read, and so on. Maximum stratum is achieved in a linear hypertext. For the reader, a linear hypertext (stratum = 1) is not interesting, since it does not profit from hypertext advantages. If the stratum is equal to zero, this indicates that structurally it makes no difference from what node one starts reading. Of course, no structural difference does not imply no logical or semantic difference. It might still be better to start reading from a given node. However, a person unfamiliar with the hypertext has no indication of where to start. A map of the hypertext does not provide this information. Furthermore, personal experience suggests that the use of structural clues helps in understanding and remembering information.

4.1.1 *Compactness* (*Cp*). Since hypertexts vary widely in the number of nodes and links, it is hard for authors to grasp the significance of those numbers. Furthermore, as Figure 14 shows, two hypertexts might have a completely different structure while having the same number of nodes and links. For this reason, a compactness metric is developed that varies between 0 and 1, independent of the hypertext size, and that reflects differences between hypertexts even when they have the same number of nodes and links. The metric is 0 when the hypertext is completely disconnected and 1 when completely connected. The compactness metric is formally defined as

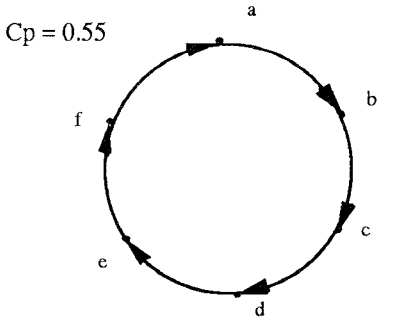
$$Cp = (\text{Max} - \sum_i \sum_j C_{ij})/(\text{Max} - \text{Min}).$$

Where Max and Min are, respectively, the maximum and minimum value the




(a)

| | a | b | c | d | e | f | COD |
|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 6 | 6 | 6 | 20 |
| b | 1 | 0 | 1 | 6 | 6 | 6 | 20 |
| c | 1 | 1 | 0 | 6 | 6 | 6 | 20 |
| d | 6 | 6 | 6 | 0 | 6 | 6 | 30 |
| e | 6 | 6 | 6 | 6 | 0 | 6 | 30 |
| f | 6 | 6 | 6 | 6 | 6 | 0 | 30 |
| CID | 20 | 20 | 20 | 30 | 30 | 30 | 150 |




(b)

| | a | b | c | d | e | f | COD |
|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 2 | 3 | 4 | 5 | 15 |
| b | 5 | 0 | 1 | 2 | 3 | 4 | 15 |
| c | 4 | 5 | 0 | 1 | 2 | 3 | 15 |
| d | 3 | 4 | 5 | 0 | 1 | 2 | 15 |
| e | 2 | 3 | 4 | 5 | 0 | 1 | 15 |
| f | 1 | 2 | 3 | 4 | 5 | 0 | 15 |
| CID | 15 | 15 | 15 | 15 | 15 | 15 | 90 |

Fig. 14. (a) Max = 180; Min = 30. Hypertext with 6 nodes and 6 links ($K = 6$). Links with arrows at both ends are actually 2 links: one link in each direction. (b) Max = 180; Min = 30 Observe that although the two hypertexts above have the same number of nodes and links (6 nodes and 6 links), hypertext 14(b) is much more compact ($Cp = 0.56$) than hypertext 14(a) ($Cp = 0.19$).

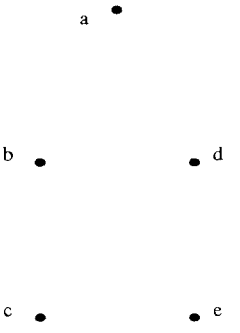
converted distance can assume. $C_{ij}$, is the converted distance between nodes $i$ and $j$ (see Section 1.1). Appendix B gives an alternate definition for $Cp$ and its value for certain familiar classes of graphs.

The maximum and minimum values of the converted distance are given by

$$\text{Max} = (n^2 - n)C; \quad \text{Min} = (n^2 - n),$$

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. This is so since there are $(n^2 - n)$ elements in the converted distance matrix that are different from zero (the diagonal is always zero). Usually, $C = K$, where $K$ is the conversion constant (see Section 1.1). For the minimum, it suffices to observe that the minimum distance between two nodes is 1. Observe that when the hypertext is completely disconnected (Figure 15a) its converted distance is maximal and $Cp = 0$. On the other hand, when the hypertext is fully connected (Figure 15b) the converted distance is minimal and $Cp = 1$. In Figures 15a, b, and c, the conversion constant $K$ is set to 5.

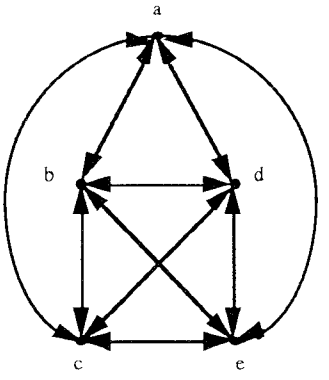
Cp = 0




| | a | b | c | d | e | COD |
|---|---|---|---|---|---|---|
| a | 0 | 5 | 5 | 5 | 5 | 20 |
| b | 5 | 0 | 5 | 5 | 5 | 20 |
| c | 5 | 5 | 0 | 5 | 5 | 20 |
| d | 5 | 5 | 5 | 5 | 5 | 20 |
| e | 5 | 5 | 5 | 5 | 0 | 20 |
| CID | 20 | 20 | 20 | 20 | 20 | 100 |

(a)

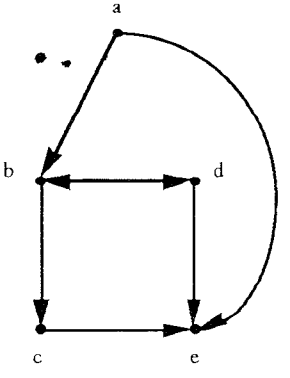
Cp = 1




| | a | b | c | d | e | COD |
|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 1 | 1 | 4 |
| b | 1 | 0 | 1 | 1 | 1 | 4 |
| c | 1 | 1 | 0 | 1 | 1 | 4 |
| d | 1 | 1 | 1 | 0 | 1 | 4 |
| e | 1 | 1 | 1 | 1 | 0 | 4 |
| CID | 4 | 4 | 4 | 4 | 4 | 20 |

(b)

Cp = 0.5




| | a | b | c | d | e | COD |
|---|---|---|---|---|---|---|
| a | 0 | 1 | 2 | 2 | 1 | 6 |
| b | 5 | 0 | 1 | 1 | 2 | 9 |
| c | 5 | 5 | 0 | 5 | 1 | 16 |
| d | 5 | 1 | 2 | 0 | 1 | 9 |
| e | 5 | 5 | 5 | 5 | 0 | 20 |
| CID | 20 | 12 | 10 | 13 | 5 | 60 |

(c)

Fig 15 (a) Max = 100; Min = 20; CD = 100. Completely disconnected hypertext has compactness metric equal to 0. $K$, the conversion constant, is set to 5 in all the examples of Figure 15 (b) Max = 100; Min = 20; CD = 20 Completely connected hypertext has compactness metric equal to 1 (c) Max = 100; Min = 20; CD = 60 As links are removed from a completely connected hypertext the compactness metric also drops It varies between 0 and 1

Coming back to Figure 14, we can now see that although the number of nodes and links does not indicate any difference between the two hypertexts; hypertext 14b is much more compact than hypertext 14a.

Three hypertexts, CMSC, GOVA, and HHO were studied in order to verify whether compactness is a good indicator of the complexity of the hypertext. CMSC, introduced in the previous section, is a hypertext that describes the Computer Science Department at the University of Maryland. It contains a description of the facilities in the department, the research areas of the faculty, details of undergraduate and graduate programs, as well as a description of the courses. Since it is a small hypertext with 106 nodes and 402 links, and the Computer Science Department at the University of Maryland has a relatively clear structure, the hypertext is well planned and easy to read.

GOVA, the Guide of Opportunity in Volunteer Archaeology, is a museum application written by historians for the Smithsonian Institution over an extended period of time [21]. This hypertext contains information about places around the world where readers can become volunteer archaeologists. Using a touchscreen, readers select (in maps and regular text) regions of the world or even particular sites. Once the place is selected, the user is presented with a text explaining the cost, best time to go, and so on. To support readers in their choice of the best place to volunteer, the hypertext contains extensive information, such as the period of history when the site was built, geographic information, and other data linked to form a vast network. GOVA is larger than CMSC, containing 222 nodes and 1609 links. It was built by authors with different goals than those of CMSC. GOVA is structured like an encyclopedia with independent articles that reference each other. It was important for the museum to enable patrons to get to the information rapidly and exit; but those who wanted to browse casually could follow the numerous links. The differences in the structure between CMSC and GOVA are reflected both in the compactness and stratum metrics.

*Hypertext Hands-On!* (HHO) [23] is the first commercially published electronic book. Is has 243 nodes and 803 links. This book is carefully crafted, with a traditional hierarchical table of contents. It covers the basic concepts of hypertext, typical hypertext applications, and currently available authoring systems. It also describes design and implementation issues such as user interface, performance, and networks.

Figure 16 summarizes the facts on the three hypertexts. Note that the compactness of CMSC and HHO are much lower than that of GOVA.

4.1.2 *Stratum* (*St*). The *stratum* metric was designed to capture the linear ordering of the hypertext. To better explain what this means, we resort to an analogy from which the concept was drawn. Many companies have a very rigid organization: the president only talks to the vice-presidents, who only talk to the directors, and those only to their immediate assistants, and so forth. For an employee at the bottom of the hierarchy it is very hard, if not impossible, to reach the president without going through all the intermediaries. Other companies are much more flexible, and the interaction

| | Number of Nodes | Number of Links | Cp |
|---|---|---|---|
| CMSC | 106 | 402 | 0.53 |
| HHO | 243 | 1104 | 0.55 |
| GOVA | 222 | 1609 | 0.78 |

Fig. 16. Compactness of three test hypertexts.

between bosses and employees is accomplished much more easily. Which of these approaches is best depends on the company and on the style of the employees. Companies of the first type are said to be stratified while the others are not stratified.

In the same way, authors of hypertexts can either write in a very stratified (hierarchical) way, with few cross-reference links between the nodes, or can be more liberal in the use of cross-reference links. As in our analogy, the best way to organize depends on the application, its authors, and readers. We can, however, postulate that readers will suffer less from the "lost in hyperspace" problem when the hypertext is written in a more hierarchical way.

The stratum metric is based on two concepts defined by Harary [12]: the *status* and *contrastatus* of members of an organization:

> "Consider the digraph of delegated authority in an organization. Call $v$ an immediate subordinate of $u$ if line $uv$ is in this digraph. And in general, call $v$ a subordinate of $u$ if $v$ is reachable from $u$. Call the subordinate vector of $u$ the sequence $(e_1, e_2, \ldots, e_{p-1})$ where $e_n$ is the number of points at distance $n$ from $u$. In analyzing status phenomena in organizations, Harary [Har59] proposed the following as reasonable requirements for the status of a person. For any point $u$ in this digraph, a status measure $s(u)$ is wanted such that:
>
> 1) $s(u)$ is an integer,
> 2) $s(u)$ is 0 if and only if $u$ has no subordinates,
> 3) if the subordinate vector of $v$ is obtained from that of $u$ by adding one subordinate (at any distance from $v$), the status of $v$ is greater than that of $u$, and
> 4) if the subordinate vector of $v$ is obtained from that of $u$ by increasing the distance of any subordinate, the status of $v$ is again greater than that of $u$.
>
> In view of these considerations, the status of person $v$ in an organization may be defined as the number of his immediate subordinates, plus twice the number of their immediate subordinates (who are not immediate subordinates of $v$), plus three times the number of their immediate subordinates (not already included), and so on. It can be shown that this particular definition gives the smallest possible measure satisfying these four requirements and, in this sense, is a natural one."

The contrastatus (as Harary puts it: "the amount of status weighting down on an individual from his superordinates") is analogous to the status; however, instead of looking for a subordinate vector, we look at the superordinate vector. The status and contrastatus can be found easily. The reader is cautioned that to find these values the distance matrix and not the converted distance matrix is used. Also observe that status and contrastatus are complementary definitions for the ROC and RIC metrics defined in Section 1.

*Definitions*

—Let $D$ be a directed graph (digraph).

—Let $d(u, v)$ be the distance between nodes $u$ and $v$ in $D$.

—The distance sum from $v_i$ in $D$ (represented by $a_i$) is the sum of the finite distances $d(v_i, u)$ for all $u$ in $D$. Thus, $a_i$ is the sum of the finite entries in the $i$th row of the distance matrix DM($D$).

—The distance sum to $v_j$ in $D$ (represented by $b_j$) is the sum of the finite distances $d(u, v_j)$ for all $u$ in $D$. Thus, $b_j$ is the sum of the finite entries in the $j$th column of the distance matrix DM($D$).

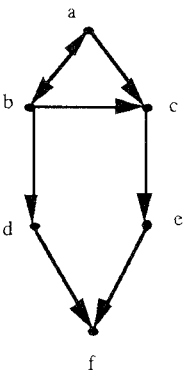
—The total distance ($\sum_i \sum_j d_{ij}$, $d_{ij} \neq \infty$) within a digraph $D$ is the sum of all the finite distances $d(v_i, v_j)$ in $D$. Thus, ($\sum_i \sum_j d_{ij}$, $d_{ij} \neq \infty$) is the sum of all the finite entries in DM($D$).

With these definitions we can see that the status of node $v_i$ is given by $a_i$ and the contrastatus of node $v_j$ is given by $b_j$. We can now define another property of each node in the hypertext, namely its *prestige*. Harary suggests that the net status of an individual, given by $a_i - b_i$, is a better indication of the prestige of an individual in a company. Figure 17 shows a hypertext with the status, contrastatus, and prestige of each node.

The total prestige (the sum of the prestige of each individual) of any organization is always 0, since the total status of the members of the organization and the total contrastatus are identical. We can, however, define the absolute prestige (or the absolute stratum) of the organization as being the sum of the absolute values of the prestige of each individual. Figure 18 illustrates why the absolute prestige is an indication of the absolute stratum of the organization. Observe that the linear hypertext (Figure 18a) has a much higher absolute prestige than the single cycle hypertext. In the first case it is very clear which one is the "president" of the organization and what are the positions of the other members. In the cyclic hypertext (Figure 18b) all nodes have the same influence over the other nodes, and consequently the absolute prestige of the hypertext is 0.

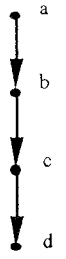
*Definitions*

—The *linear absolute prestige* (LAP) of a hypertext with $n$ nodes is identical to the absolute prestige of a linear hypertext with $n$ nodes. For example, the LAP of hypertext in Figure 18a is 16, since this is the absolute prestige of a hypertext with 4 nodes.




| | a | b | c | d | e | f | Stat | Prest |
|---|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 | 2 | 3 | 9 | 8 |
| b | 1 | 0 | 1 | 1 | 2 | 2 | 7 | 6 |
| c | ∞ | ∞ | 0 | ∞ | 1 | 2 | 3 | 1 |
| d | ∞ | ∞ | ∞ | 0 | ∞ | 1 | 1 | -2 |
| e | ∞ | ∞ | ∞ | ∞ | 0 | 1 | 1 | -4 |
| f | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0 | -9 |
| CStat | 1 | 1 | 2 | 3 | 5 | 9 | 21 | |

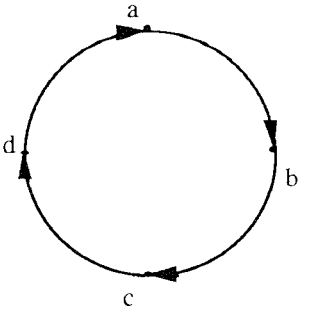
Fig 17. Hypertext with the "status," "contrastatus," and "prestige " Note that nodes that reach all the other nodes in the hypertext have status identical to their COD metric Similarly nodes that are reached by all other nodes in the hypertext have identical contrastatus and CID metric




Distance Matrix (DM)

| | a | b | c | d | Stat | Prest |
|---|---|---|---|---|---|---|
| a | 0 | 1 | 2 | 3 | 6 | 6 |
| b | ∞ | 0 | 1 | 2 | 3 | 2 |
| c | ∞ | ∞ | 0 | 1 | 1 | -2 |
| d | ∞ | ∞ | ∞ | 0 | 0 | -6 |
| CStat. | 0 | 1 | 3 | 6 | 24 | 16 |

(a)




Distance Matrix (DM)

| | a | b | c | d | Stat. | Prest. |
|---|---|---|---|---|---|---|
| a | 0 | 1 | 2 | 3 | 6 | 0 |
| b | 3 | 0 | 1 | 2 | 6 | 0 |
| c | 2 | 3 | 0 | 1 | 6 | 0 |
| d | 1 | 2 | 3 | 0 | 6 | 0 |
| CStat. | 6 | 6 | 6 | 6 | 24 | 0 |

(b)

Fig 18 (a) Absolute prestige of the hypertext is 16 This hypertext is also used as a basis for finding the linear absolute prestige of all hypertexts that have 4 nodes (b) Absolute prestige of the hypertext is 0. No structural clue to which node should be read first is present

—To define the *stratum* of the hypertext, we normalize the absolute prestige of the hypertext by dividing it by its LAP. Formally, the stratum ($St$) is defined as

$$St = \textit{absolute prestige}/\text{LAP}.$$

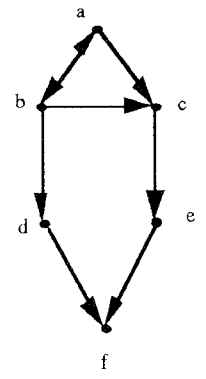
The linear absolute prestige (LAP) is given by the following formula (see the appendix)

$$\text{LAP} = \begin{cases} \dfrac{n^3}{4}, & \text{if } n \text{ is even.} \\ \dfrac{n^3 - n}{4}, & \text{if } n \text{ is odd.} \end{cases}$$

Figure 19 shows two almost identical hypertexts with the same number of nodes, where one differs from the other only by an extra link. Observe, however, that the presence of this link completely changes the stratum of the hypertext. Hypertext 19b has a stratum metric much smaller than hypertext 19a. Although in this simple example one hypertext might be as easy to read as the other, hypertext 19b has no structural clue as to which node should be read first, as in hypertext 19a, since in 19b one could start reading the hypertext by any node and be led to the others. In the case of hypertext 19a, on the other hand, the reader should start reading from either node "a" or "b" only.

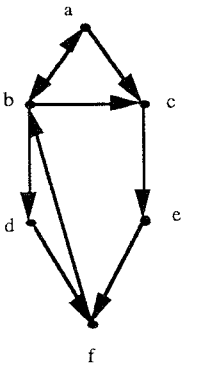
In the definition of stratum, the absolute prestige of the hypertext is divided by its linear absolute prestige. This was done in order to normalize the stratum value so that it would be insensitive to the hypertext size in nodes. However, normalization is not an easy issue. First, from a user interface point of view, it is not clear that the size of the hypertext should be removed as a concern. As the hypertext size grows, the concern with nonlinearity increases because more structuring is needed to prevent disorientation. Normalization against other properties such as number of links, length of paths, or other properties should be considered. Second, normalizing against size is intended to permit some comparison between hypertexts of different sizes—but they are inherently different in size and normalization may obscure the difference. This parallels text readability metrics, which might indicate similar grade levels for two documents that are orders of magnitude different in length. Clearly, the longer document will take longer to read and be more difficult to read. We realize that stratum is not a perfect metric, some problems are evident and only a few hypertexts have been checked, but we are encouraged by these first steps.

Stratum and compactness are not independent measures. For example, if the compactness is equal to 1 (hypertext is totally connected) then its stratum is 0. Figure 20, however, shows how the compactness and stratum indicate different properties in the hypertext. Note that in part 20a and 20b the stratum is the same but the compactness of hypertext 20b is higher. The stratum remains constant, since the amount of status and contrastatus




| | a | b | c | d | e | f | Stat | Prest. |
|---|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 | 2 | 3 | 9 | 8 |
| b | 1 | 0 | 1 | 1 | 2 | 2 | 7 | 6 |
| c | ∞ | ∞ | 0 | ∞ | 1 | 2 | 3 | 1 |
| d | ∞ | ∞ | ∞ | 0 | ∞ | 1 | 1 | -2 |
| e | ∞ | ∞ | ∞ | ∞ | 0 | 1 | 1 | -4 |
| f | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0 | -9 |
| CStat | 1 | 1 | 2 | 3 | 5 | 9 | 21 | |

(a)




| | a | b | c | d | e | f | Stat. | Prest. |
|---|---|---|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 2 | 2 | 3 | 9 | -4 |
| b | 1 | 0 | 1 | 1 | 2 | 2 | 7 | -2 |
| c | 4 | 3 | 0 | 4 | 1 | 2 | 14 | 4 |
| d | 3 | 2 | 3 | 0 | 4 | 1 | 13 | 1 |
| e | 3 | 2 | 3 | 3 | 0 | 1 | 12 | 0 |
| f | 2 | 1 | 2 | 2 | 3 | 0 | 10 | 1 |
| CStat. | 13 | 9 | 10 | 12 | 12 | 9 | 65 | |

(b)

Fig. 19. (a) Absolute prestige = 30; lap = $6^3/4 = 54$; stratum = $30/54 = 0.56$. (b) Absolute prestige = 12; LAP = $6^3/4 = 54$, stratum = $12/54 = 0.22$. Two almost identical hypertexts, where, by adding only a link, the stratum varies substantially.

acquired by each of the nodes in the lower level of the hypertext is identical. From Figures 20b-20c the stratum increases due to the fact that, although the status and contrastatus acquired by the nodes in the lower level of the hypertext are identical, nodes "c" and "d" have acquired status but no contrastatus since they can now reach more nodes, but not be reached by them. Figure 20d shows the extreme case where the stratum is 0, but the hypertext has a reasonable compactness metric.

Again, CMSC, HHO, and GOVA are studied and their stratum compared. Their stratum are respectively 0.13, 0.05, and 0.01. From these numbers, the following conjectures can be made:

—Hypertexts, even the simple ones such as CMSC, are far from linear, which shows that authors use cross-reference links liberally.

—Since the LAP grows in the order of $O(n^3)$ it does not make much sense to compare hypertexts that have large differences in the number of nodes. Although HHO is very well structured, its stratum is half that of CMSC. However, the stratum still indicates the complexity of the hypertexts.

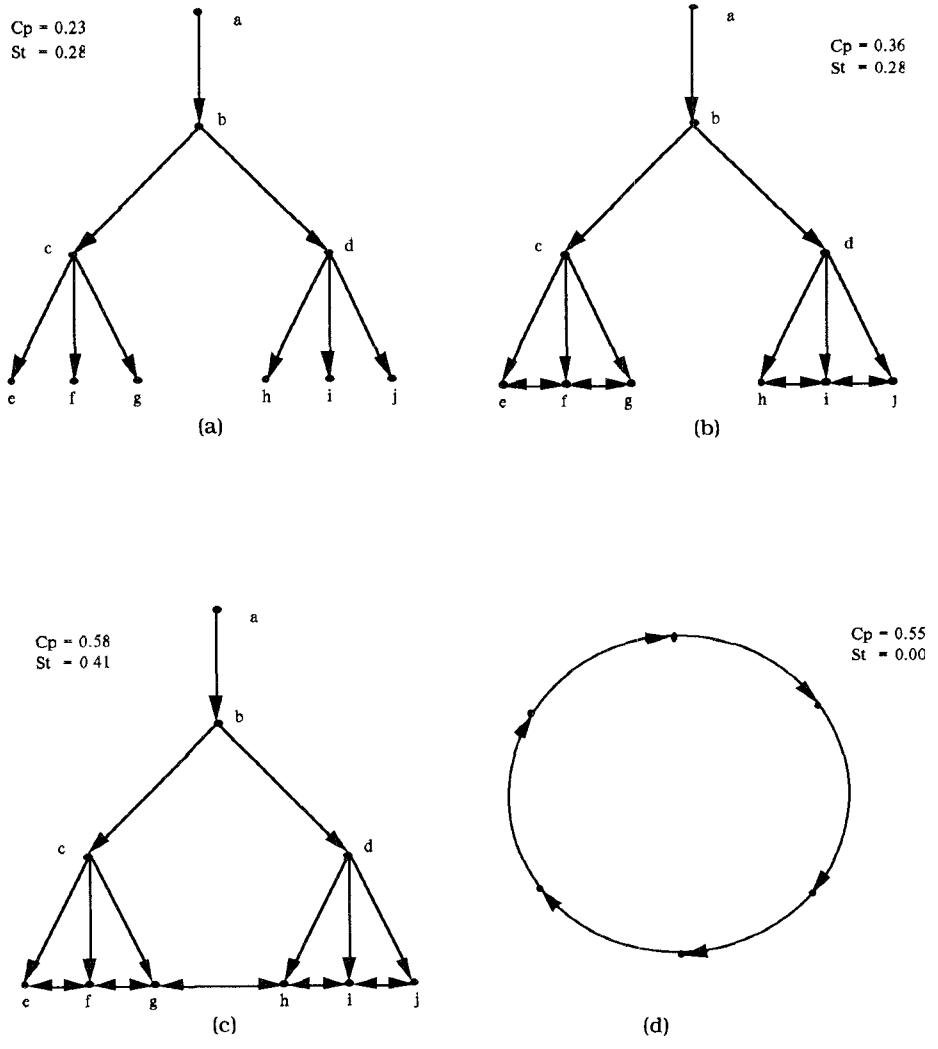



Fig. 20. Although stratum and compactness are not independent measures, they still indicate different properties of the hypertexts.

Whereas HHO has more nodes than GOVA (243 and 222, respectively), its stratum is five times as large. This is appropriate, as the authors of HHO intend to guide readers to introductory articles before guiding them to advanced ones.

—For future work, it would be interesting to analyze other well-structured hypertexts, to propose to authors reasonable ranges for stratum for hypertexts having 100, 200, 300, (etc.) nodes.

Figure 21 shows the effect on the stratum of the removal of the links going to and from reference and index nodes. Contrary to what happens with compactness, the stratum goes up when the links are removed. For CMSC

| | Number of Nodes | Number of Links | St | Number of Links | St |
|---|---|---|---|---|---|
| CMSC | 106 | 402 | 0.13 | 278 | 0.24 |
| HHO | 243 | 1104 | 0.05 | 803 | 0.10 |
| GOVA | 222 | 1609 | 0.02 | 1184 | 0.03 |

Fig. 21 Effect on stratum of the removal of links from index and reference nodes.

and HHO the stratum almost doubles, suggesting that these hypertexts are much more "linear" than it seemed on a first examination. However, GOVA's stratum does not suffer such a dramatic increase. We recommend exploration of alternative normalizations for the stratum that are based on hypertext size.

Nodes that have a very low status and very low contrastatus are nodes that are hard to leave and hard to reach. For example, a disconnected node can only be reached and be left through the index. Knowing about disconnected nodes is important to authors, and all authoring systems should provide this information. However, the case of nodes with low status and low contrastatus is more interesting, since they cannot be spotted so easily. In general, nodes with very low contrastatus are less likely to be browsed, since their distance to other nodes is high. An exception is a node that is only connected to the root. In general, nodes with very low status indicate that they are reachable largely by a back-up facility or the index, and therefore they are almost disconnected. Authors may wish to add links to make these nodes more accessible.

### 4.2 Node Metrics

In this section we define two new metrics for each node in the hypertext: *depth* and *inbalance*. These metrics are designed to further indicate special nodes in the hypertext. For instance, nodes that are too deep in the hypertext are hard to reach. This might indicate that links are missing or that the information contained in this node is very specialized. Imbalanced nodes are those that are at the root of an imbalanced tree. Having an imbalanced tree might indicate to the author that the sparse branches should be developed more thoroughly.

4.2.1 *Depth*. By definition the depth of a node in a hypertext is just its distance from the root. The bigger the distance of a node from the root, the harder it is for the reader to reach this node and consequently the less important this node will be in the hypertext. Nodes that are very deep inside the hypertext are unlikely to be read by the majority of the readers. Such nodes might indicate possible "bugs" in this hypertext. A special case of very hard to reach nodes is when the node cannot be reached by the root. In this

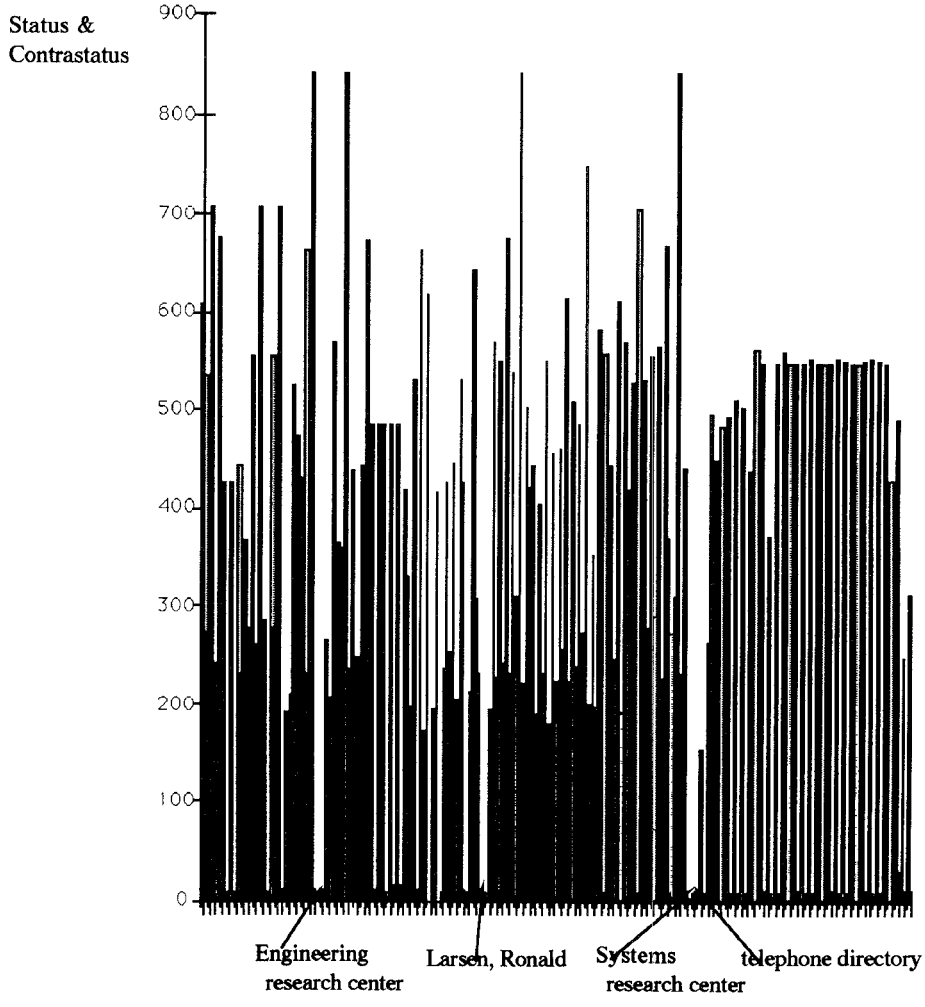



Fig. 22. Superposition of the histograms of the status and contrastatus in CMSC.

case the node is said to be at infinite distance from the root, and it will be impossible for a reader to reach it (actually almost all hypertext systems have some kind of index or string search that allows users to reach any node, thus bypassing the hypertext structure).

Deep nodes are not always a "bug." An author may decide to store a low relevance piece of information deep inside the hypertext. In this way, readers whose interest is not so strong can browse the hypertext without seeing the low relevance information, while more interested readers will be able to achieve a deep understanding of the subject. An example of the appropriate use of deep nodes can be taken from the *Hypertext Hands-On!* book [23] where the average depth of a node is 4. The hyperfiction has a node whose distance from the root is 10. This high depth is not an error, but is intentional

complexity in the story. By having access to a depth metric, authors can locate unreachable or deep nodes and verify that they were intentional. Figure 23 shows the title of some articles in the *Hypertext Hands-On*! book and their depth in the hypertext.

4.2.2 *Imbalance Metrics.* When writing a linear text a rule usually recommended is to develop the ideas with the same care. This implies that the length (in number of words written) of the ideas should be similar and that the number of subideas of each of the main ideas should also be equal. Looking at the outline of a text written in the way just described one observes that each item has an equal number of subitems. In hypertexts, if we assume that each node contains only one idea and that the links emanating from a node are a further development on that idea (unless they are cross-reference links), then we might want the hypertext to be a balanced tree.

Of course, imbalance in a hypertext is not always bad. What is really important is that each topic is fully developed. Thus, in a university department hypertext, there may be many more levels of information on professors and their areas than on locations of copying machines. However, we should expect that each of the major areas will be treated with the same care. Balance is not necessary but too much imbalance might indicate a bias of the author or a poorly designed hypertext. Again it should be remembered that those metrics are to be used for feedback to authors. If they decide that imbalances are desired then they can overlook the information. Figure 24 shows two trees with the same number of nodes, but tree 24a is more balanced than tree 24b.

This section will try to answer two questions: first, is the hypertext well-balanced? And second, if not, then what are the nodes that root imbalanced subtrees? After defining more clearly the notion of balance, imbalance metrics for each node are defined. Those having a very high imbalance metric are the roots of subtrees that may need to be balanced. Two different types of imbalance are studied: depth and child imbalance.

We assume that a link represents a development of an idea, unless it is a cross-reference link. For this reason we use the results found in the previous section that distinguish between cross-reference links and hierarchical links. In what follows we assume that the algorithm developed in the previous section has already been run on the hypertext and that we are dealing with rooted trees instead of general graphs.

*Definitions*

—Let $T$ be a general rooted tree.

—Let $a_1, a_2, \ldots, a_n$ be children of node $a$. We define the *depth vector* of a ($D(a)$) as being:

$$D(a) = \begin{cases} [1 + \text{Max}(D(a_1)), 1 + \text{Max}(D(a_2)), \ldots, 1 + \text{Max}(D(a_n))], \\ [0] \qquad \text{if } a \text{ has no child } (n = 0) \end{cases}$$

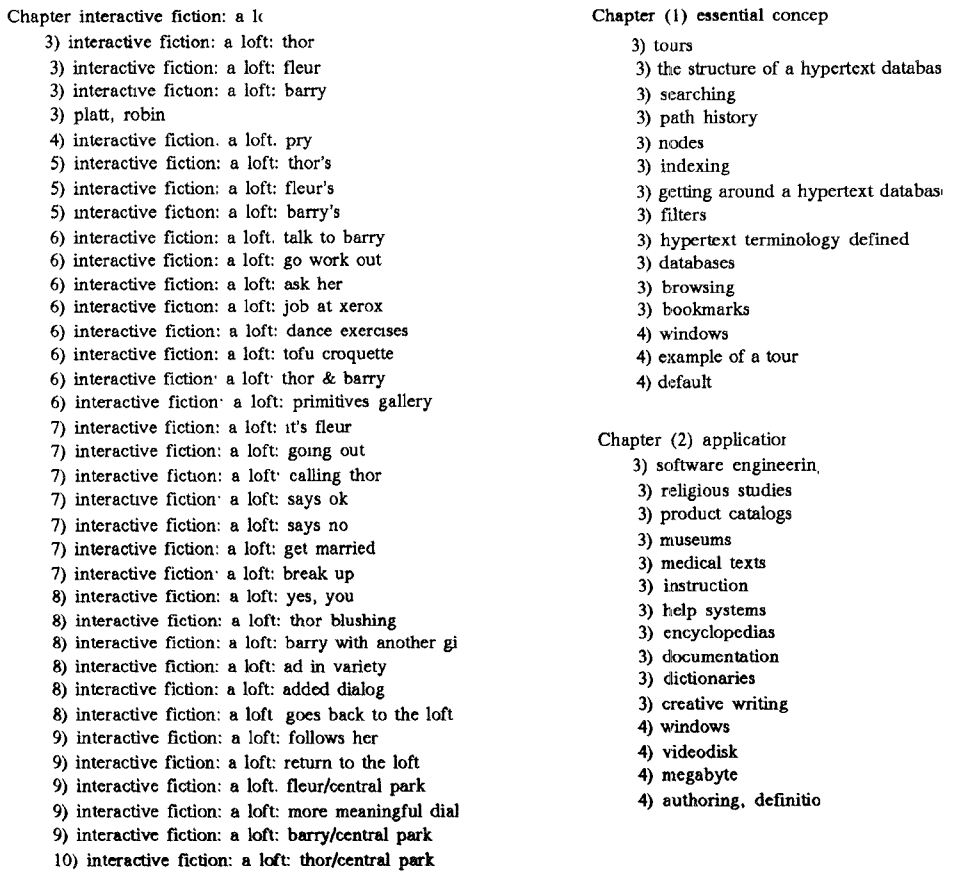



Fig. 23. *Hypertext Hands-On!* hypertext. Number before the article name indicates the depth The maximum depth of the interactive fiction chapter, which is 10, is much larger than the average depth of the rest of the hypertext, which is approximately 4.

where $\text{Max}(D(a_i))$ indicates the value of the biggest element in the vector $D(a_i)$. The depth vector is represented inside square brackets ([ ]). Intuitively, this vector indicates the maximum distance one can go by following each of the children of node $a$.

—The *child vector* of a ($C(a)$) is defined as

$$C(a) = \begin{cases} \{1 + \sum C(a_1), 1 + \sum C(a_2), \ldots, 1 + \sum C(a_n)\}, \\ \{0\} \qquad \text{if } a \text{ has no child } (n = 0) \end{cases}$$

where $\sum C(a_i)$ indicates the sum of all elements in vector $C(a_i)$. The child vector is represented inside braces ({}), and indicates the size (number of elements) of the subtrees rooted at $a_1, a_2, \ldots, a_n$.

—We define the *absolute depth imbalance* for node $a$ to be the standard deviation of the elements in vector $D(a)$.

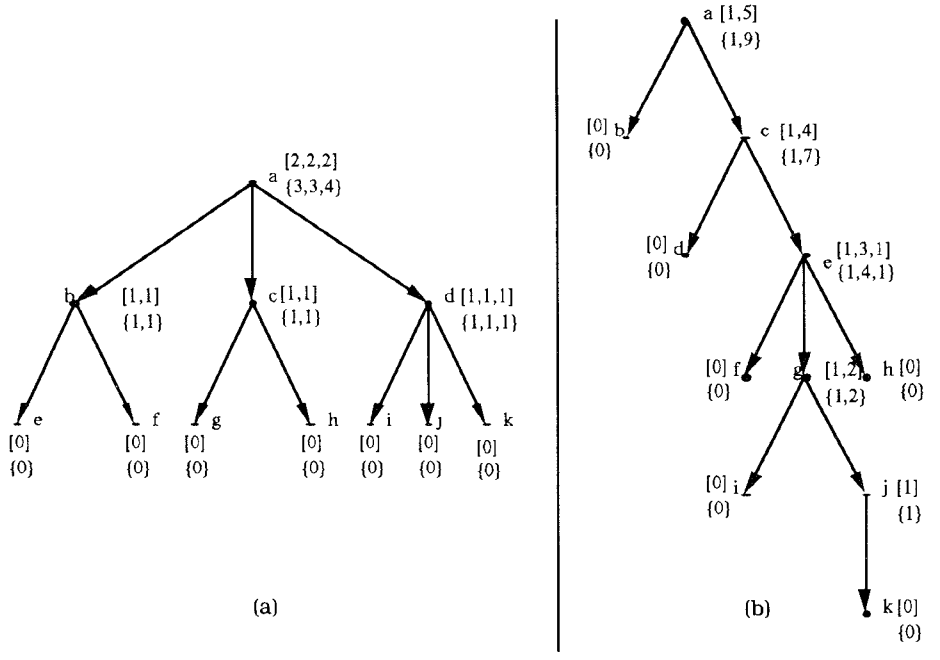



Fig. 24. (a) Balanced hypertext (b) Imbalanced hypertext.

—Similarly, we define the *absolute child imbalance* for node $a$ to be the standard deviation of the elements in vector $C(a)$.

## 5. CONCLUSION

We suggest one way to analyze the structure of a hypertext by identifying hierarchies and metrics. Diverse hierarchies give authors different views of the hypertext, facilitating its structuring and the identification of errors. Metrics are useful tools for comparing different hypertexts and also ensuring that one's hypertext has the expected properties. However, numeric objective approaches should never override the creativity and judgment of authors. Interpretations made from metrics should be carefully verified.

In Section 3, we saw how to identify potential roots, that is, nodes that can easily reach all other nodes in the hypertext. Floyd's algorithm ($O(n^3)$) was used to find the distances between all the nodes [7]. With these distances, it is easy to find the ones with high *relative out-centrality*. Once a root is selected by the author, a simple modification of breadth-first-search ($O(n)$) is used to separate cross-reference from hierarchical links. If authors know which node they want as the root, then finding the root does not need to be performed.

If these algorithms are being used to analyze very large automatically created hypertexts, then algorithm execution time is a serious issue, but this analysis is still much faster than going over the hypertext manually. Faster mean time algorithms [16, 24] or parallel execution should be explored.

Another solution is finding the distance between nodes incrementally as authors add new nodes. This can be implemented in $O(|E| + |V|)$ with a simple modification of breadth-first-search. Since hypertext graphs are usually sparse (on average no more than a dozen links per node), this operation can be performed fairly fast. The compactness and stratum computations depend on the distance between nodes. The imbalance and depth metrics can both be implemented in $O(|E| + |V|)$.

We applied these algorithms in three Hyperties databases: *Hypertext Hands-On!* (HHO), the *Guide in Opportunities in Volunteer Archaeology* (GOVA), and a database about the Computer Science Department of the University of Maryland (CMSC). While HHO and CMSC are hierarchically separated into chapters and sections, GOVA is structured as an encyclopedia, with no hierarchy. This difference is reflected in the compactness and stratum of these hypertexts. Although GOVA does not have a clear global structure, there are many hierarchies. By finding the roots of these hierarchies, authors can have a much clearer notion of this hypertext's contents.

By using structural analysis it was also possible to identify errors in the hypertexts. For instance, in CMSC some nodes were in the wrong place in the intended hierarchy. In analyzing the structure we realized that a link was missing. Adding the link removed the problem. Finding this problem by inspection would have required a thorough study of the whole hypertext, since it was hard to even see that there was a problem without structural analysis.

We believe that the analysis of the hypertext structure will yield many interesting results and that as more research is done new problems and ideas will present themselves. A suggested list for future research follows:

—We have concentrated our analysis on hypertext systems that have a simple underlying structure. Will the ideas that we present work in more complex hypertext systems? For instance, how to deal with typed or conditional links?

—Since the hypertexts we have studied are all authored with Hyperties, one can argue that the system induced authors to produce similarly structured hypertexts. Will the results be as effective in other systems such as HyperCard or Guide?

—The only area where we suggest how to show the results of our algorithm to the end user is in the automatic generation of maps. New techniques need to be created for users (authors and readers) to easily identify relevant nodes in a hypertext.

—Hypertext databases can be generated automatically. It would be interesting to discover if even in those cases hierarchical structures exist. For example, if a hypertext database is automatically generated from the "man" pages in Unix, will the stratum of the hypertext be very low or will it be similar to other examples? Is it possible to classify hypertexts by looking at their metrics?

—The stratum metric is highly dependent on the number of nodes, making it hard to compare two hypertexts. This dependence is caused by the proposed

normalization, but a better normalization method might solve the problem. A tree, which offers very clear structural clues to the reader, can have a low stratum metric because normalization is done against a linear hypertext. Maybe normalization should be done against some type of tree.

—Many metrics were developed. Is it possible to devise other interesting metrics? For instance, would it be possible to devise a "readability" or "comprehensibility" metric? Our experience with the three hypertexts presented here plus other hypertexts that we analyzed suggests that the answer is yes. Compactness and stratum revealed interesting properties of the hypertexts we studied. Also, depth and the other node metrics revealed properties of the hypertexts that we were not aware of. We believe that further study will suggest new metrics that will capture other important properties of hypertexts.

## APPENDIX A: THE LINEAR ABSOLUTE PRESTIGE

LEMMA. *The linear absolute prestige (LAP) of a hypertext with n nodes is given by*

$$LAP = \begin{cases} \dfrac{n^3}{4}, & \text{if } n \text{ is even.} \\ \dfrac{n^3 - n}{4}, & \text{if } n \text{ is odd.} \end{cases}$$

PROOF. Let $n$ be the number of elements in the linear hypertext. Assume that node "1" is the root and that it links to node "2" that links to node "3," and so on. Let $s_i$ be the status of node $i$ and $cs_i$ its contrastatus. The following is easily seen for the contrastatus:

$cs_1 = 0$; since node 1 is not reachable.

$cs_2 = 1$; it is at distance 1 from node "1"

$cs_3 = 3$; it is at distance 1 from node "2" and distance 2 from *node* "1". 2 + 1 = 3.

and in general:

$$cs_n = \sum_{i=1}^{n-1} i \therefore cs_n = \frac{n^2 - n}{2}.$$

Status can be related to the contrastatus in the following manner:

$$s_1 = cs_n;\ s_2 = cs_{n-1};\ \ldots;\ s_i = cs_{n-(i-1)};\ \ldots;\ s_n = cs_1;$$

By definition, LAP is $LAP = |s_1 - cs_1| + |s_2 - cs_2| + \cdots + |s_n - cs_n|$. When $n$ is an even number the above formula can be expanded as follows:

$$LAP = |cs_n - cs_1| + |cs_{n-1} - cs_2| + \cdots + |cs_{n-((n/2)-1)} - cs_{n/2}| + |cs_{n-(n/2)} - cs_{(n/2)+1}| + \cdots + |cs_1 - cs_n|.$$

Observe that because of the absolute value, the formula above is symmetric. Therefore,

$$\text{LAP} = 2\left[(cs_n - cs_1) + (cs_{n-1} - cs_2) + \cdots + (cs_{(n/2)+1} - cs_{n/2})\right] \Rightarrow$$

$$\text{LAP} = 2\sum_{i=1}^{n/2} cs_{n-i+1}cs_i$$

but

$$cs_{n-i+1} - cs_i = \frac{(n-i+1)^2 - (n-i+1)}{2} - \frac{i^2 - i}{2} = \frac{n^2 - 2ni + n}{2}$$

from this we get

$$\text{LAP} = \sum_{i=1}^{n/2} (n^2 - 2ni + n).$$

As a result, when $n$ is even

$$\text{LAP} = \frac{n^3}{4}.$$

When $n$ is odd the expansion done in Equation 2 has to be slightly modified, resulting in a different upper limit for the summation. We have

$$\text{LAP} = |cs_n - cs_1| + |cs_{n-1} - cs_2| + \cdots + |cs_{(n+1)/2} - cs_{(n+1)/2}| + \cdots + |cs_1 - cs_n|$$

$$\text{LAP} = 2\sum_{i=1}^{\frac{n-1}{2}} cs_{n-i+1} - cs_i$$

but it has already been seen that

$$cs_{n-i+1} - cs_i = \frac{n^2 - 2ni + n}{2}$$

as a result,

$$\text{LAP} = \sum_{i=1}^{\frac{n-1}{2}} (n^2 - 2ni + n).$$

Resolving the above summation yields

$$\text{LAP} = \frac{n^3 - n}{4}, \qquad \text{when the number of nodes is odd.} \qquad \square$$

## APPENDIX B: SOME CP RESULTS FOR FAMILIES OF DIGRAPHS

Values that the *Cp* metric yields for familiar classes of graphs can help with the interpretation of the compactness metric. Basically, compactness

indicates the level of connectedness. Compactness can be defined as follows:

$$Cp = 1/\big(n(n-1)\big) * \sum \left[i \neq j\right] s_{ij},$$

where $s_{ij} = (c - c_{ij})/(c-1)$ with $c$ the conversion constant.

*Unidirectional star* (single root, connected by unidirectional out-links to all others; a hypertext structured only by an index):

$$\begin{aligned} Cp &= 1/\big(n(n-1)\big) * \big((n-1) * (c-1)/(c-1) + (n-1) * n * (c-c)/(c-1)\big) \\ &= 1/\big(n(n-1)\big) * (n-1) = 1/n \\ &\Rightarrow 0 \text{ as } n \Rightarrow \text{large}. \end{aligned}$$

*Bidirectional star* (single root, connected by bidirectional links to all others):

$$\begin{aligned} Cp &= 1/\big(n(n-1)\big) * \big(2 * (n-1)(c-1)/(c-1) \\ &\quad + (n-1) * (n-2) * (c-2)/(c-1)\big) \\ &= 2/n + (n-2)(c-2)/\big(n(c-1)\big) \end{aligned}$$

which, if you choose $c = n$, $\Rightarrow 1$ as $n \Rightarrow$ large.,

*Two disjoint complete graphs* with cardinality $n1$ and $n2$, $n1 + n2 = n$:

$$\begin{aligned} Cp &= 1/\big(n(n-1)\big) * \left[n1 * (n1-1)(c-1)/(c-1)\right. \\ &\quad \left. + n2 * (n2-1)(c-1)/(c-1)\right] \\ &= 1 - \big(2 * n1 * n2\big)/\big(n(n-1)\big), \end{aligned}$$

this is minimal at $n1 = n2 = n/2$, where

$$Cp = 1/2 - 1/(2n-2) \Rightarrow 1/2 \text{ (as recommended)}.$$

*Unidirectional cycle*

$$\begin{aligned} Cp &= 1/\big((n-1)n\big)\left[(c-1) + (c-2) + \cdots + \big(c-(n-1)\big)\right] * 1/(c-1) \\ &= 1/(n-1)\left[(n-1) * c - (n**2 - n)/2\right] * 1/(c-1) \\ &= (c - n/2)/(c-1) \end{aligned}$$

which, if you choose $c = n$, $\Rightarrow 1/2$ as $n \Rightarrow$ large.

*Bidirectional cycle*

$$\begin{aligned} Cp &= 1/\big(n(n-1)\big) * 2/(c-1) * \left[(n/2-1) * c - \big((n/2-1)+1\big)/2\right] \\ &= (c - n/2)/(c-1) \end{aligned}$$

which, if you choose $c = n$, $\Rightarrow 3/4$ as $n \Rightarrow$ large.

*Unidirectional linear*

$$Cp = 1/(n(n-1))[((n-1)*n*c)/2*(c-1) - 1/(c-1)*$$
$$1/2*[n*(n-1) + (n*(n-1)*(2n-1) + 3*n*(n+1))/6]$$
$$= c/((c-1)*2) - 1/2*(c-1)[1 + (2n-1)/6 + (3(n+1)/2(n-1))]$$

which, if you choose $c = n$, $\Rightarrow 1/2 - 1/6 = 1/3$ as $n \Rightarrow$ large.

ACKNOWLEDGMENTS

We appreciate the dilligent efforts of Robert B. Allen in expediting the reviewing process and also the numerous thoughtful comments of the referees, especially those of G. Furnas.

REFERENCES

1. Botafogo, R. A. Structural analysis of hypertexts. Unpublished Master's Thesis, Univ. of Maryland, College Park, 1990.
2. Botafogo, R. A , and Shneiderman, B. Identifying aggregates in hypertext structures. In *Proceedings of the Hypertext '91 Conference*. ACM, New York, 1991, pp. 63–74.
3. Brown, P. J. Do we need maps to navigate round hypertext documents? *Electron. Publishing 2*, 2 (1989), 91–100.
4. Charney, D. Comprehending non-linear text: The role of discourse cues and reading strategies. In *Proceedings of the Hypertext '87 Conference* (Charlotte, N.C., Nov. 13–15, 1987). ACM, New York, 1987, pp. 109–120.
5. Conklin, J., and Begeman, M. gIBIS: A tool for exploratory policy discussion, *ACM Trans Office Inf. Syst. 6*, 4 (1988), 303–331.
6. Egan, D. E., Remde, J. R., Gomez, L. M., Landauer, T. K., Eberhardt, J., and Lochblm, C. C Formative design-evaluation of SuperBook, *ACM Trans. Inf. Syst. 7*, 1 (1989), 30–57
7. Floyd, R. W. Algorithm 97: Shortest path *Commun. ACM. 5*, 6 (1962), 345
8 Furnas, G W. Generalized fisheye views. In *Proceedings of the CHI '86 Conference* (Boston, Apr. 13–17, 1986) ACM, New York, 1986, pp. 16–23.
9. Halasz, F. G., Moran, T. P., and Trigg, R. H. NoteCards in a nutshell In *Proceedings of the ACM CHI + GI '87 Conference* (Toronto, Ont , Apr. 5–9, 1987). ACM, New York, 1987, pp 45–52
10 Halasz, F. G. Reflection on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM 31*, 7 (1988), 836–852
11 Harary, F. Status and contrastatus *Sociometry 22* (1959), 23–43.
12. Harary, F., Norman, R. Z., and Cartwright, D. *Structural models. An Introduction to the Theory of Directed Graphs*. Wiley, New York, 1965
13. Kaufmann, A. Graphs, dynamic programming and finite games. In *Mathematics in Science and Engineering 36*. Academic Press, New York 1967.
14. Kintsch, W , and van Dijk, T. Toward a model of text comprehension and production *Psychol. Rev. 85* (1978), 363–394.
15 Marshall, C. C Guided Tours and on-line presentations: How authors make existing hypertext intelligible for readers. In *Proceedings of the Hypertext '89 Conference* (Pittsburgh, Pa., Nov. 5–8, 1989). ACM, New York, 1989, pp. 15–26.
16. Moffat, A. and Takaoka, T An all pairs shortest path algorithm with expected running time $O(n^2 \log n)$. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, 1985, pp. 101–105.
17. Nielsen, J. The art of navigating through hypertext *Commun. ACM 33*, 3 (1990), 296–310.
18. Nielsen, J. *Hypertext & Hypermedia*. Academic Press, New York, 1990.

19. NEUWIRTH, C., KAUFER, D., CHIMERA, R., AND TERILYN, G. The notes program: A hypertext application for writing from source texts In *Proceedings of the Hypertext '87 Conference*, ACM, New York, 1987, pp. 121–135.
20. PAUSCH, R., AND DETMER, J Node popularity as a hypertext browsing aid. *Electron. Publishing· Original Dissemination Des. 3*, 4 (1990), pp. 227–234.
21. PLAISANT, C. Guide to Opportunities in volunteer archaeology—Case study of the use of a hypertext system in a museum exhibit. In *Hypertext/Hypermedia Handbook*, E. Berk and J. Devlin, Eds., McGraw-Hill, New York, 1991, pp. 498–505.
22. SALTON, G *Automatic Text Processing*. Addison-Wesley, Reading, Mass., 1989.
23. SHNEIDERMAN, B , AND KEARSLEY, G. *Hypertext Hands-On!* Addison-Wesley, Reading, Mass , 1989
24 SIRA, P. M. A new algorithm for finding all shortest paths in a graph of positive arcs in average time $O(n^2 \log^2 n)$, *SIAM J. Comput. 2*, 1 (1973), 28–32.
25 STOTTS, P. D., AND FURUTA, R. Petri-net-based hypertext: Document structure with browsing semantics *ACM Trans Inf Syst. 7*, 1 (1989), 3–29
26. STOTTS, P D , AND FURUTA, R Hierarchy, composition, scripting languages, and translators for structure hypertext. In *Proceedings of the European Conference on Hypertext* (Paris, 1990) pp. 180–193.
27. TRIGG, R , AND IRISH, P Hypertext habitats: Experiences of writers in NoteCards In *Proceedings of the Hypertext '87 Conference* (Charlotte, N.C., Nov 13–15, 1987) ACM, New York, 1987, pp. 89–108
28 VALDEZ, F , AND CHIGNELL, M Browsing models for hypermedia databases. In *Proceedings of the Human Factors Society, 32nd Annual Meeting* (Santa Monica, Calif., 1988), Human Factors Society, 1988, pp. 318–322.
29 VAN DIJK, T , AND KINTSCH, W *Strategies of Discourse Comprehension*. Academic Press, New York, 1983.
30 ZELLWEGER, P T. Scripted documents: A hypermedia path mechanism In *Proceedings of Hypertext '89 Conference* (Pittsburgh, Pa., Nov. 5–8, 1989). ACM, New York, 1989, pp. 1–14