Chapter 13

Using Pathfinder to Evaluate User and System Models*

Wendy A. Kellogg and Timothy J. Breen

Mental Models and User Performance

The notion that a user's mental model of a software system has a critical impact on the user's ability to effectively use systems has gained widespread acceptance in the field of human-computer interaction (e.g., Carroll & Olson, 1988; Hammond, Morton, MacLean, & Barnard, 1983; Kieras & Bovair, 1984; Masson, Hill, Conner, & Guindon, 1988; McDonald & Schvaneveldt, 1988; Waern, 1987). The acceptance of the "mental model hypothesis" is motivated in part by the belief that faulty or incomplete representations (misconceptions) lead to errors, and that the kinds of errors users make can be understood once a model of their knowledge is derived (see, e.g., Masson et al., 1988). However, verifying this claim empirically and applying it in practice to system design or user training has met with mixed success (see Carroll & Olson, 1988, and Rouse & Morris, 1985, for critical reviews). Some studies have found performance benefits for users learning an appropriate model of a system or programming language (e.g., Kieras & Bovair, 1984; Linde, 1986; Mayer, 1987). Others have found little benefit or inconsistent benefits in giving learners device models (Halasz & Moran, 1983; Polson[1]).

In our view, there are two fundamental problems contributing to this state of affairs. The first is that the target body of knowledge represented by a software system has rarely been articulated. Without the definition of a system model, what the user *should* know and therefore what his mental model should contain are unknown. Defining an adequate model of a software system is made difficult by disagreement about what kind of knowledge it should encompass (e.g., "how to do it" vs. conceptual or "how it works" knowledge), and by our currently limited understanding of the relationship of different kinds of knowledge and user performance.

A second fundamental problem with improving the status of the mental model hypothesis is the difficulty of "capturing" the user's mental model, particularly in a way that can be systematically compared with a system model. The variety of techniques employed to date virtually spans the repertoire of psychological methods, and each method yields a different kind of mental model. Clearly, the way a researcher derives a mental model is critical to any assessment of whether the user's model is an important determinant of

*This work was carried out while the second author held a Predoctoral Internship 1985-1986 at the User Interface Institute, IBM Thomas J. Watson Research Center, Yorktown Heights, NY. We thank James McDonald and Roger Schvaneveldt for comments on an earlier version of this chapter. Portions of this research were presented at CHI+GI'87 and reported in Kellogg and Breen, 1987.

[1]Personal communication to W.A. Kellogg, 1987.

performance. In addition, for the purpose of applying assessment of user and system models to system design, pragmatic "cost-benefit" characteristics must be taken into account (e.g., whether the information gained is worth the effort to acquire it). The work reported here assumes the importance of the user's conceptual knowledge of a system. Our focus is evaluating the utility of *scaling techniques* as definitions of user and system conceptual models that can inform system design. Our interest in users' conceptual models of computing systems reflects an instantiation of the mental model hypothesis which asserts that the more congruent the user's structure of knowledge with that represented by the system, the more easily learned and usable the system should be. Our interest in scaling methods reflects both theoretical and pragmatic concerns, as elaborated below.

Defining Mental Models

Various methods have been employed in the study of mental models, including protocol analysis, production system modeling, and scaling techniques. The kind of model a researcher chooses to derive depends both on the kind of knowledge being represented (e.g., declarative or procedural) and the use to which the model will be put, since different methods have different strengths and weaknesses. Protocol techniques (real-time "think-aloud," post-task video confrontation or interviews, or inferring knowledge fragments or misconceptions from a protocol of user keystrokes) are particularly appropriate for relating user errors to misconceptions. However, they are hard to summarize and compare systematically. Analytic approaches, such as GOMS (Card, Moran, & Newell, 1983) or production system modeling (Kieras & Polson, 1985), may be useful for evaluating the efficiency and consistency of designed methods, but may be less useful in understanding the genesis of user errors, or in evaluating the difficulty for learners of the conceptual knowledge represented by a system. Scaling techniques have some of the advantages of both protocol and analytic methods. Like protocol methods, they explicitly represent conceptual knowledge and are empirically derived. Like analytic methods, once derived, models defined by scaling techniques can be systematically and quantitatively compared among users and between users and the system. But scaling techniques have their vulnerabilities as well: How do context-free judgments of the relatedness of system concepts bear on user performance, or on predictions of learnability and usability? The primary weakness of the scaling methodology is less a matter of the method itself than of its application—understanding what can and should be made of its results.

Pragmatic concerns may also influence the choice among methods, particularly if the focus is upon utilizing the information gained for system design purposes. Protocol techniques require a running system and are time-consuming to collect and analyze. Analytic modeling can be performed before a system is implemented, but requires considerable time and skill on the part of the analyst. Scaling techniques, in contrast, are startlingly simple to employ and can also be used before a prototype or running system is available. As such, they are an attractive candidate for evaluating the compatibility of users' knowledge of a task domain with a system or proposed system; the crux of the issue is whether they can yield valuable information to the design process.

Scaling analyses have been used in system design, particularly for organizing the presentation of information in the interface (McDonald & Schvaneveldt, 1988). However, the extent to which the use of scaling techniques can be extended to other interface issues is unknown. One goal of the current work was to extend the use of these techniques to the structure of a system's *functionality* per se. This was possible because of the nature of the system we studied, where the declarative structure of the commands as defined by the system model often determined when and where commands could be successfully used.

Using Scaling Techniques in System Design

Mental models defined by scaling techniques have the potential to be used to assess a system's usability and learnability in a way that depends on both the user and the system. In order to employ them in this way, it must be shown that the relationship between a user's model and the system model changes with experience in the predicted fashion. In particular, experience in using a system should be correlated with increases in the amount of overlap between the user and system models. This presupposes a methodology for extracting and expressing a model from users and the system in comparable forms. The use of mental models for assessing usability and learnability must also be empirically verified. The closeness of a user's model to the system model should predict performance on the system. Salient discrepancies between expert users' models and the system model should indicate modifications to the system that would improve its usability. Similarly, the distance between novices' models and the system model should correlate with the difficulty of learning a system. To verify these assumptions, a way of measuring the degree of agreement between user and system models is necessary. The work reported here attempts to verify the expected relationship between experience and the amount of overlap between user and system models defined by network scaling. This involved specifying a system model, deriving user models from groups of users with different amounts of experience with the system, and developing methods for assessing the degree of agreement between users' models and the system model. User networks were based on data representing subjective judgments about the structure of a system. A system model based on the system documentation was derived using the same scaling technique. To our knowledge, the work reported here is unique in specifying a system model from the documentation for direct comparison with empirically derived user models, and in its attempt to use the outcome of the comparison to suggest usability improvements.

Method

The Formatting System. The domain we studied was a command-driven text formatting language in which users format documents by labeling their components. The system is designed to take advantage of users' knowledge of typical document structure. Traditional formatting systems require specification of desired format from the user in terms of low-level components, such as spacing, line breaks, justification, and control characters. In contrast, the system under study defines more abstract components which entail a set of low-level formatting effects. These components (called *tags*) can be interpreted appropriately by different output devices. The user's task in formatting, then, is to label parts of the document with appropriate tags.

The structural diagram of the system from the documentation is shown in Figure 1. It divides the system into eight major categories: General Document, Headings, Basic Text, Displays, Lists, Index, Footnotes, and Process-Specific Controls. Within the categories appear the set of 51 tags which users might apply to parts of the document. All of the categories, except Basic Text and Process-Specific Controls, represent structural elements of typical documents.

The declarative structure of the tags is important to their appropriate use. In general, the system is hierarchically structured, and this structure has implications for where and when a tag can be used. In the General Document category, for example, Title Page elements (with the exception of Address and Address Line) can *only* be used within the scope of the Title Page tag. The system documentation explicitly lists Address and Address Line in both the Title Page group and the Basic Text group, since these tags can be used in the Body of the document as well as in the Title Page.

The implications of the system's view of document structure for obtaining desired formatting effects made the use of scaling techniques for evaluating learnability and usability as outlined above attractive. In particular, we hoped the document element networks derived with Pathfinder would show us several things: (1) the evolution of the user's view of document structure with experience using the system; (2) *what* restructuring of knowledge needed to occur for new users (an assessment of learnability); and (3) whether any disagreements with the system's view of document structure remained for experienced users (an assessment of usability). In addition, because of the central role of document structure in this system, and because nonusers should already have a good understanding of document structure in general, the use of scaling techniques represented a potentially strong test of the sensitivity of the method.

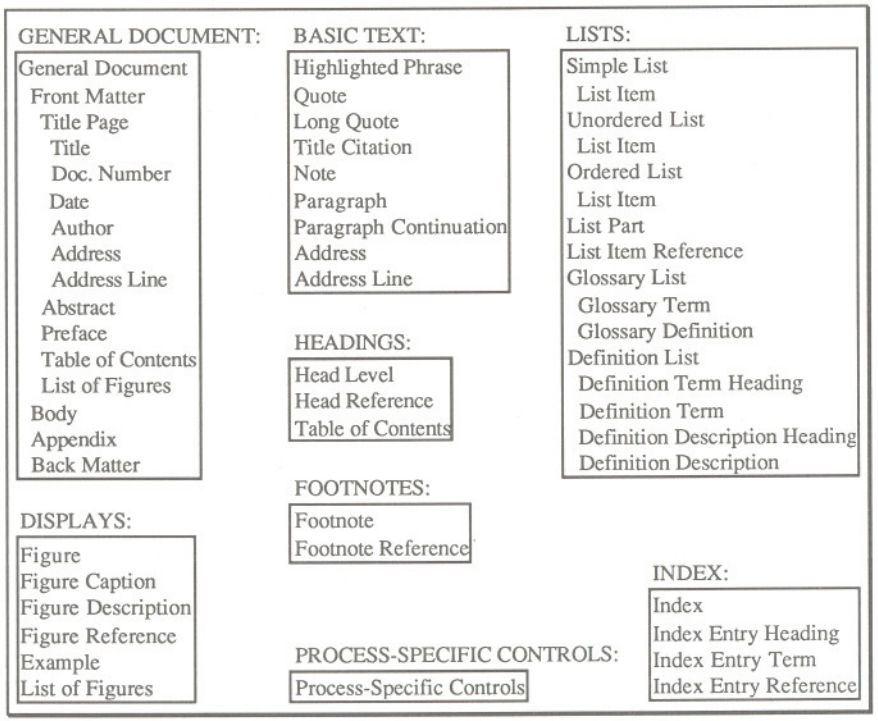


Figure 1. Structural diagram of system from documentation.

Participants. Fifteen experienced users and 15 nonusers volunteered to participate in the study. The experienced users had used the system in their work for a minimum of 7 months, with a mean of 3 years. These participants used the formatting system routinely in producing work documents ranging from research papers, to books, memos, and overhead projector foils. Most had some familiarity with formatting systems other than the one under study.

The nonuser participants had no experience with the system under study, and 4 out of 15 had no experience with formatting systems at all. One had not had any experience with computer systems. All except two worked in industrial research or academic settings.

Materials. The materials for the experiment consisted of a set of 51 index cards with one formatting command written on each card. The verbal labels for the commands were used, rather than the actual form of the command that would be used to markup a document.

Procedure. Participants read instructions describing the task. They were told that the set of cards represented parts of a formatting system, and that each card would do something to affect how a document would look when it was printed out (for experienced users, the system was explicitly identified). Participants were asked to look through the set of cards while thinking about what each might do, and then to sort the cards into any number of piles such that things that *seemed* related were in the same pile. They were also told they could use as many piles as they wished, and that they could change their arrangement until it seemed best. The experimenter clarified any questions; the participant then sorted the cards until satisfied with the arrangement. The number and content of the groups were recorded. Finally, participants were asked to rate each of the 51 commands for familiarity by checking any commands they had never used or with which they were unfamiliar.

Extraction of the System Model

The system model was based on a structural diagram created by one of the system designers for the system documentation. The diagram partitions the system into major components, each with hierarchical structure, as described above. Deriving the system model from the structural diagram was based not on the premise that the typical user would explicitly attempt to acquire a model of the system from it, but rather on the judgment that in this case it was the clearest and most objective representation of the system upon which to base a model.

Data for the system model were based on distances between pairs of concepts in the command set. Distances were determined from the structural diagram shown in Figure 1 as follows. Concepts within a subgroup were given a distance of 1. Concepts one level apart in the hierarchy were scored as distances of 2. Each additional level was scored by incrementing the distance count by 1. Concepts which were not direct descendants of a higher-level concept were given the distance score between levels plus 1. Thus, for example, the distance between Title and Front Matter was 3, whereas the distance between Title and Body was 4. Unrelated concepts were given an "infinite" distance of 5. Distances ranged from 1 to 5 for the entire set of concepts. The distances then served as input to the scaling analyses.

Results

The data from participants' card sorts were transformed into a measure of distances between concepts, which then served as input for the scaling analyses.

Individual agreement with the system data. Agreement between an individual participant and the system was determined by computing the mean correlation across the set of 51 concepts in the following way. For each participant, a vector of the 1275 possible pairwise combinations of the 51 concepts was constructed. Each value in the vector represented whether the participant had sorted those two concepts in the same pile or in different piles; the whole vector then represented the participant's entire card sort. The value for each possible combination was expressed as a linear transformation of the distance (0 for

cards sorted together, 1 for cards sorted separately; transformed to 1's and 2's, respectively). The system data were represented in a similar fashion, with distances in the vector ranging from 1-5 as described above. The vector for each participant was then correlated with the system vector, resulting in a score for each participant representing the degree of agreement between each participant's card sort and the structure represented by the system diagram. The average correlation with the system for the experienced user group was .48; for the nonuser group, .30. The correlation for 10/15 of the experienced users was greater than the largest nonuser/system correlation. Participant scores from the two experience groups were compared with the Mann-Whitney U Test (Hays, 1973); the difference was significant: $z = 3.21$, $p < .003$, two-tailed.

Intragroup agreement. The vector for each individual was similarly correlated with all the other individuals in their experience group. In this analysis, 14/15 experienced users had higher correlations with other such users than the highest correlation among nonusers (for experienced users, mean = .37; for nonusers, mean = .20). The difference in intragroup correlation was significant by the Mann-Whitney U Test ($z = 4.33$, $p < .00007$, two-tailed). Replicating previous results (Cooke & Schvaneveldt, 1988), experienced users displayed higher agreement among themselves than did nonusers.

Functional grouping. Another finding reported previously (e.g., Kay & Black, 1984, 1985) is the tendency to structure knowledge more in terms of *function* than in terms of *surface features* with experience. Accordingly, we looked for evidence of functional grouping by individuals in each experience group. For four functional groups defined by the system (Basic Text, Front Matter, Title Page, and Displays), we set a criterion ranging from 67%-80% of the commands in the group for determining whether a user had sorted the commands functionally. For each functional group, we then looked at the number of users in each experience group who used that functional grouping, the mean percent of "correct" commands they included, and the mean percent of "extra" commands included in the card-sort group (relative to the size of the system-defined functional group).

The results of the Basic Text group are representative of the general pattern across the four functional groups. In addition, this group represents a particularly strong test of the presence of functional grouping since the commands it contains have few surface features in common. Using a criterion of 5/7 commands present, 9/15 experienced users had Basic Text groups. These groups contained 80% of the Basic Text commands, and 17% extraneous commands. Only 4/15 nonusers met the criterion for a Basic Text group, on the other hand, and while these contained 79% of the Basic Text commands, they also included 136% extraneous commands. The means across the four functional groups were: for experienced users, 10/15 had functional groups containing 85% correct concepts and 21% extraneous concepts; for nonusers, 4/15 had functional groups containing 84% correct concepts and 138% extraneous concepts. Experienced users showed strong evidence of functional grouping, with groups containing most of the correct commands and very few other commands. Nonusers were much less likely to group commands functionally, and when they did have an intact functional group, it was typically in the presence of many other extraneous commands.

In summary, analyses based on *individuals* indicated that experienced users were in closer agreement with the system than were nonusers, and were in closer agreement with each other than were nonusers. In addition, experienced users showed more evidence of sorting system commands on the basis of function than did nonusers. We then looked at the agreement of user groups as a whole with respect to system concepts based on the user and system models defined by Pathfinder.

Network Scaling

Comparison of User and System Networks

Pathfinder networks were derived from the distance data for both experience groups and the system with $r = \infty$ and $q = n-1$ (Schvaneveldt, Durso, & Dearholt, 1985). The resulting networks for the system, for experienced users and nonusers, are shown in the Appendix. In the user networks, bold lines are used to denote links *shared* with the system network, and thin lines are used to indicate links not present in the system network. In addition, on the user networks, "lassos" have been drawn around eleven subgroups defined by the system (General Document, Front Matter, Title Page, Basic Text, Displays, Headings, Lists, Definition List, Glossary List, Footnotes, and Index) to the extent that they are present. The system network contained 138 links, the experienced user network contained 82 links, and the nonuser network contained 73 links. Of the experienced users' 82 links, 69 (84%) were shared with the system network, 13 (16%) were not shared. For nonusers, 43 (59%) links were shared with the system network and 30 (41%) were not shared.

We examined the degree of agreement between a user network and the system network by computing for each of the 51 concepts the correlation of user-defined and system-defined links. This resulted in a set of scores for each user group, with each score representing the degree of agreement between a user groups' treatment of a concept (in terms of what it was linked to) and the system's treatment of the concept. Across the 51 concepts, experienced users correlated .640 with the system network, nonusers correlated .421 with the system network. This difference was tested by examining the difference (in direction and magnitude) between scores from each experience group for each concept with the Wilcoxon Test (Hays, 1973) and was significant: $z = 4.39$, $p < .00003$. Similarly, the correlation between experienced users' and nonusers' networks was .425; experienced users were more highly correlated with the system network than with the nonuser network ($z = 4.48$, $p < .00003$).

Our previous analysis of the extent of functional grouping in the raw data indicated that experienced users utilized functional grouping more than nonusers. Another aspect of this question is the role played by surface feature similarity in each experience group's network. We examined this question by constructing a three-way contingency table which classified each link in a user group's network as connecting nodes that were *functionally related* or *not functionally related* (nodes linked in the system network were considered to be functionally related) and which had *surface similarity* or *no surface similarity*. Surface similarity was defined as command names having one or more words in common (e.g., Front Matter and Back Matter). A test for the degree of association between functional relatedness and surface similarity for each experience group (Fienberg, 1977) was significant ($z = 3.03$, $p < .002$). Nonuser network links showed a greater dependency between functional relatedness and surface similarity ($\alpha = 3.99$) than experienced user network links ($\alpha = .291$), and the association was in opposite directions for each group. Nonuser links that were shared with the system model (i.e., were functionally related) tended to be those with surface similarity. Functional relatedness and surface similarity were much less strongly related for experienced users, on the other hand, and to the extent they were related at all, experienced user links tended to match the system network more closely for nodes *without* surface similarity.

The network derived by Pathfinder for the system (see Appendix) is regular and simple to describe. The system defines eight basic categories, three of which (General Document, Lists, and Displays) have hierarchical structure. Each basic category appears in the network as a *cluster* with members of the category fully linked (one category, Process-Specific Controls, does not appear in the network because it contained a solitary command). Certain commands *connect* different clusters (e.g., Title Page connects the Front Matter cluster with the Title Page cluster). Such commands reflect one of two characteristics of the system. The majority of connecting commands reflect hierarchical structure in the system. This can be seen clearly in the General Document cluster (containing Appendix, Back Matter, Body, and Front Matter), connected through the Front Matter tag to the Front Matter cluster (Abstract, Table of Contents, List of Figures, Preface, and Title Page), connected through the Title Page tag to the Title Page cluster (Address, Address Line, Title, Author, Date, and Document Number). In the three other cases where clusters are joined, the connecting commands belong to *two* categories in the system model (e.g., Table of Contents is contained in both the Headings and General Document categories). For two of these cases (Table of Contents and List of Figures), the system's double categorization reflects a functional relationship: for example, the Table of Contents tag *uses* Headings to construct the Table of Contents. The third case, Address and Address Line, categorized in both the Title Page and Basic Text categories, reflects the fact mentioned previously that these tags, unlike the other Title Page tags, can be used within the body of a document as well as in the title page. Thus, overall, the network produced by Pathfinder for the system cleanly reflects its categorical and hierarchical structure, and some of its functional aspects.

The user networks, in contrast, show categories (as evidenced by tightly interconnected clusters) and hierarchical structure (in the pattern of links) much less clearly. However, clustering is much more developed in the network of experienced users than in the nonuser network. Nonusers show rudimentary clustering for Lists and Title Page elements. Within the Lists category, some hierarchical structure can be seen in the cluster of Definition List tags, connected to the Lists cluster through the Definition List tag. Otherwise, the nonuser network tends to be linear (i.e., exhibits point-to-point connections between commands). Experienced users, on the other hand, show well-developed clusters for Title Page, Basic Text, Lists, and Displays, less-developed but discernible clusters for Definition List and General Document elements, and two clusters *not* defined by the system model—a References group and an Index/Glossary group.

The Reference tag group (List Item Reference, Index Entry Reference, Figure Reference, Footnote Reference, and Head Reference) serves as a bridge between the experienced users' Footnotes, Headings, Displays, Lists, and Index/Glossary clusters. The nonuser network also links four of the five Reference tags, connecting Lists, Figure, and Index commands. While a References cluster is *not* present in the system network (the system categorized each Reference tag within its category type), its presence in the user networks is not in *conflict* with the system model in two senses. First, its presence in the user network is similar to the kind of connecting nodes the system network displays for functional relatedness (e.g., the Table of Contents tag described above); from the user's point of view, the Reference tags share a similar function and are, in fact, the only semantic similarity that the "categories" they connect share. Second, the Reference bridge structure has no negative implications for *using* Reference tags in marking up a document; unlike some of the other tags, their use is not constrained by the overall document structure.

The presence of the Index/Glossary cluster in the experienced user network, on the other hand, is a deviation that *does* conflict with the system. The Index and Glossary List elements are connected through the Index command and in turn the Appendix command to

Back Matter. This pattern of network links suggests that experienced users think of Back Matter as a higher-order structure containing the Appendix, Index, and Glossary structures.[2] Examination of the system network, however, reveals a conflict, which in this case has implications for using the Appendix tag correctly. The system treats Appendix as a *major structural element*, along with Body, Front Matter, and Back Matter, under the higher-order structure of General Document. If a user tries to create an Appendix *within* the Back Matter of a document, it will not function correctly; in fact, the Back Matter tag acts as a delimiter for Appendices (i.e., indicating that the end of the Appendices section has been reached). The reason that Appendix was treated as a major document element, we found by contacting one of the system designers, was not one totally unfamiliar to aficionados of usability: The designer told us that Appendix became a major document element because it was an efficient means for implementing the automatic heading function associated with appendices. Interestingly, the pattern of links from Back Matter through Appendix to Index is also present in the nonuser network (although there Appendix is also mysteriously linked to Table of Contents and Preface).

In summary, overall comparison of the networks defined by Pathfinder revealed that the experienced user's view of document structure, as defined by patterns of shared and nonshared links, is closer to the system's view than is the nonuser's view, and is more like the system's view than like the nonuser's view. Examination of user networks, particularly in contrast to the cleanly structured "ideal" system network, reveals both the evolution in network structure with system experience and "failures to evolve," instances where discrepancies with the system structure persist and are indicated by the treatment of particular commands. In the next section, we describe a more systematic approach to the treatment of individual commands by experienced users and nonusers.

Command Definition

Following Cooke and Schvaneveldt (1988), we examined experienced users' and nonusers' understanding of system commands by categorizing each command with respect to the proportion of shared and extraneous links. For each command, the proportion of links shared with the system network links for that command and the proportion of extraneous links (relative to the number of links for that command in each experience group's network) was computed. Commands were then categorized as *well-defined, misdefined, overdefined*, or *underdefined*, based on a grand median split (over both experience groups) for shared and extraneous links. Well-defined concepts were above the median on shared links, below the median on extraneous links. The other categories reflected the other three possibilities: misdefined (low shared, high extraneous), overdefined (high shared, high extraneous), and underdefined (low shared, low extraneous).

In this analysis, experienced users had 49% well-defined concepts, 11.8% misdefined, 19.6% overdefined, and 19.6% underdefined concepts. Nonusers had 21.6% well-defined, 43.1% misdefined, 19.6% overdefined, and 15.7% underdefined concepts.

Consideration of the commands which fell into each of these categories, supported the analysis presented above based on inspection of the networks. First, the large difference between experienced users and nonusers in the proportion of well-defined concepts (49% vs. 21.6%) reflects both the overall amount of clustering in each experience groups' network and its degree of alignment with the system network. This is because the system network is so dominated by clusters: For a concept to be well-defined, it had to be linked to at least half of the other concepts in the system cluster (the grand median for shared links

[2]This view of Back Matter for experienced users was also confirmed by a hierarchical clustering analysis (see Kellogg & Breen, 1987).

was .50) and have *no* other links (the grand median for extraneous concepts was 0). This means that concepts could be well-defined in user networks *only* when they exhibited a fair degree of clustering (interconnectedness), and only with concepts included in the system cluster. In a complementary fashion, misdefined concepts reveal structural deviances in user networks. For example, the 22 commands misdefined by nonusers constituted substantial representation of seven of the eight basic categories defined by the system (the eighth category, Process-Specific Controls, was overdefined). The well-defined commands for nonusers suggest that only the Title Page and Definition List clusters were intact; the misdefined commands reflect the more general lack of correspondence between these participants' and the system's view of document structure. For experienced users, the well-defined concepts reflect intact clusters for five of the eight categories (General Document, Basic Text, Lists, Displays, and Process-Specific Controls) as well as the "subclusters" of Front Matter, Title Page, and Definition List. Two of the other categories (Headings and Footnotes) are congruent with the system definition, but were overdefined because of the links between the Reference tags. The misdefined concepts for experienced users did not reflect the kind of pervasive deviations from the system network shown in the nonuser network, with the exception of the Back Matter problem discussed above: Three of the six misdefined concepts were related to the Appendix and Index tags.

Finally, we analyzed users' familiarity ratings with respect to the definitional status of commands. Experienced users' ratings of commands were congruent with their well-definedness as defined by network links: Overall, .17 of these users rated well-defined concepts as unfamiliar or never used, .31 for misdefined, .30 for overdefined, and .28 for underdefined concepts, respectively. Nonusers showed less discrimination, as well as less of a tendency to rate commands as unfamiliar (none, of course, had ever been used). Overall, .12 of the nonusers rated well-defined concepts as unfamiliar, .10 for misdefined, .12 for overdefined, and .17 for underdefined concepts, respectively.

Discussion

User and System Models Defined by Pathfinder

Our intent in deriving users' mental models and an idealized system model with Pathfinder was to explore the utility of network scaling for evaluating aspects of a system's learnability and usability. The first step in assessing this utility is confirming that users' models grow closer to the system model with increasing experience. The models defined by Pathfinder confirm this assumption and reveal details of the evolution of user knowledge with experience using the system.

The nonuser network reveals a significant amount of disorganization in structure. Of the 11 "lassoed" groups, only four exist completely intact (Definition List, Glossary List, Footnotes, and Index), and *all* of the commands included in these groups have surface similarity as previously defined. Almost intact are Lists, Displays, and Title Page—but here, *discrepancies* related to surface features stand out: Nonusers linked "List of Figures" to the Lists group (it belongs to the Front Matter group), linked "Title Citation" to the Title Page group (part of Basic Text tags), and did *not* link "Example" to the Displays group (all of its other members had the word "figure" in the command name). The remaining four groups are not substantially represented.

What does the nonuser network suggest about the learnability of this system for naive users? First, despite the real-world familiarity of document structures (e.g., most naive users can be expected to be familiar with the structural layout and composition of books),

only one subgroup (title page elements) is well-developed for these users. The higher-level structure of major document elements (Front Matter, Body, and Back Matter) is not well expressed. This suggests that a strong emphasis be placed on the overall hierarchical document structure used by the system in user training and the interface itself.[3]

Second, it is clear from the network that the roles of Front Matter and Back Matter are not understood by the nonusers. This was confirmed by the familiarity ratings: 60% of the nonuser participants rated Front Matter and Back Matter as unfamiliar; in fact, these were the most frequent commands to be labeled unfamiliar. Finally, it is clear that nonusers will benefit from surface similarity in the naming of functionally related commands. This suggests that special attention be given to groups of functionally related commands that cannot be designed with surface features in common. In the system studied here, for example, commands in the Basic Text group did not share any surface features. Our results suggest that a redesign of these commands to reflect their common functionality in such features might enhance the comprehensibility of the system for learners.

Of course, it is an open question whether conceptual knowledge, such as the functional similarity among Basic Text commands, will have any real effect on learners. This question cannot be answered without performance data for learners of this system, which we did not collect. However, the present data *do* show that the comparison of system and nonuser Pathfinder networks can reveal the lack of appreciation of the functional similarity.

An example discussed by Carroll, Mack, and Kellogg (1988) suggests an analogous lack of understanding of functional similarity that did have performance consequences for learners. The example involved the task of creating new folders in the *Lisa*. All new documents were created in this system by "Tearing Off Stationery" from a paper pad icon. This method was applied to the creation of folders as well. However, the folder icon and other paper pad icons, which were functionally similar, did not share salient surface features (e.g., paper pads were all labeled "Paper" and had similar icons, but the folder icon looked different and was simply labeled "Folders"). Carroll and Mazur (1986) observed that learners had difficulty discovering how to create new folders with the system, though they were able to create other types of documents. In fact, subsequent versions of this desktop interface changed the new folder method by adding a special action for folder creation. In this case, of course, there is no Pathfinder data. But the present data strongly suggest that a Pathfinder network for naive *Lisa* users *would* reveal the lack of perceived similarity between folders and other paper pads.

The evolution of user knowledge toward the system model is shown by the network for experienced users. In their network, all 11 subgroups defined by the system are intact (see the "lassoed" groups in the Appendix). The experienced users' network also shows more developed clustering than the nonuser network and clearer hierarchical structure. Experienced users demonstrate an understanding of the major structural elements of the system model (with the exception of Appendix, discussed previously). They *use* this structure, particularly Front Matter and Back Matter, to organize the major document elements that appear from the front to the back of a document. Experienced users rarely marked Front Matter or Back Matter as "unfamiliar" or "never used."

On the other hand, there are also similarities remaining with the nonuser network. For example, the use of Reference tags as a bridge among different clusters can be seen in both user networks. Experienced users link the Basic Text group to the rest of the network through a "Footnote-Note" link which also occurs in the nonuser network. Thus, bridging

[3] Although the system we studied was command-driven, a menu-based interface to the system has been implemented and could emphasize the system view of document structure in a way the current system is unable to do.

links in the experienced users' network often reflects the kind of (a priori) semantic similarity that characterize nonuser networks in general.

The experienced users' view of document structure and system commands does not coincide perfectly with the system model. The two major discrepancies involve the linking of Reference tags and the grouping of Appendix, Index, and Glossary commands under Back Matter. As discussed previously, the linking of Reference tags is a discrepancy without usability implications. The organization of commands under Back Matter, however, does have performance implications: The experienced users' treatment of Back Matter, and particularly the Appendix tag, suggests that they would experience difficulty marking up a document containing an appendix, because they would place the appendix *within* the back matter, rather than *before* it as required by the system. The comparison of the experienced user and system models suggests that system usability could be improved if it was redesigned to allow placement of appendices within the back matter while still providing the automatic heading function. In the current arrangement, the system handles the headings, but at the cost of an unintuitive structuring of document elements.

The present results underscore the importance of defining a system model. Without representing the system model, the difference in experienced users' organization of major document elements would not have been found. The more typical comparison of expert and novice knowledge structures *can* show the evolution of the organization of user knowledge with experience, but for assessing usability in terms of "model congruence" as suggested here, a representation of the system's view of the task domain is essential. The more completely the system model incorporates the functional relations and organization of conceptual knowledge that users *ought* to have of the system, the more informative the comparison with user models will be. Alternative methods of defining a system model (e.g., obtaining judgments from system designers, using system specifications, examining the system directly) are possible and must be evaluated in terms of how well they represent the target knowledge for the system.

The system model we derived from the system documentation for this study, in retrospect, is a fairly good representation of the necessary conceptual knowledge for this system, but it could easily be improved. Our system model was incomplete in representing some of the functional relationships in the system: for example, it did not represent what the Back Matter is *supposed* to contain from the system's point of view (Index and Glossary List). Rather, Index tags were represented as an isolated cluster, and Glossary List tags were represented only within the Lists cluster.

Another aspect of functionality only partially represented in our system model has to do with the system's double categorization of the Table of Contents and List of Figures tags. The system linked Table of Contents to the Headings tags in addition to Front Matter tags because the Head Level tags are used to generate the table of contents. Similarly, List of Figures relies on Figure tags to construct itself. These functions only work properly if the user specifies a "twopass" command option when sending the marked-up document to an output device. The same is true of footnote references: They will only print properly with the "twopass" option. We might have been able to examine our users' understanding of the twopass mode had we included "twopass" as a concept to be sorted and allowed participants to place concepts in multiple piles (or had we insisted that duplicates of some of the concepts be included). Again, in retrospect, we might well have left out some of the more detailed commands (e.g., the subgroup of Definition List tags) which share obvious surface features (and thus are likely to be grouped together by all participants), and which are less interesting in terms of the functional relations embedded in the system. To gain the most information from the comparison of user and system models, and to keep the number

of concepts to be judged within reasonable bounds, the analyst may have to select the most important functional and conceptual relationships to include in the system model and the comparison set.

Using Network Scaling in System Design

The results of the present study suggest that Pathfinder networks have much potential for providing information on the match between users' views and the system's organization of a task domain. In particular, it seems possible to extend the use of network scaling beyond issues of organizing information in the interface to questions about the inherent conceptual structure of the system and its functionality.

The comparison of user and system networks can yield interesting and potentially valuable information about a system's learnability and usability. However, discrepancies revealed by such a comparison must be evaluated in the larger context of the user's task domain if their import for usability in the system's real-usage context is to be correctly anticipated. For example, our results suggest that on the whole the system is congruent with experienced users' models of document structure. How significant the Appendix/Back Matter deviation is for the system's usability will depend on *how often* users will engage in the task of marking up appendices. Nevertheless, the ability of the Pathfinder analysis to reveal the discrepancy, and the relatively low cost of deriving user and system networks, suggests that scaling methods have an appropriate cost/benefit profile for use in system design.

A second issue is *how* the outcome of the Pathfinder analysis and comparison of user and system models can guide design. McDonald and Schvaneveldt (1988) recommend that systems be designed to conform to user models. The comparison of Pathfinder networks for users (or potential users) and a system can suggest specific ways the system might be restructured to be more compatible with users' models. We suggested this in the case of the treatment of appendices.

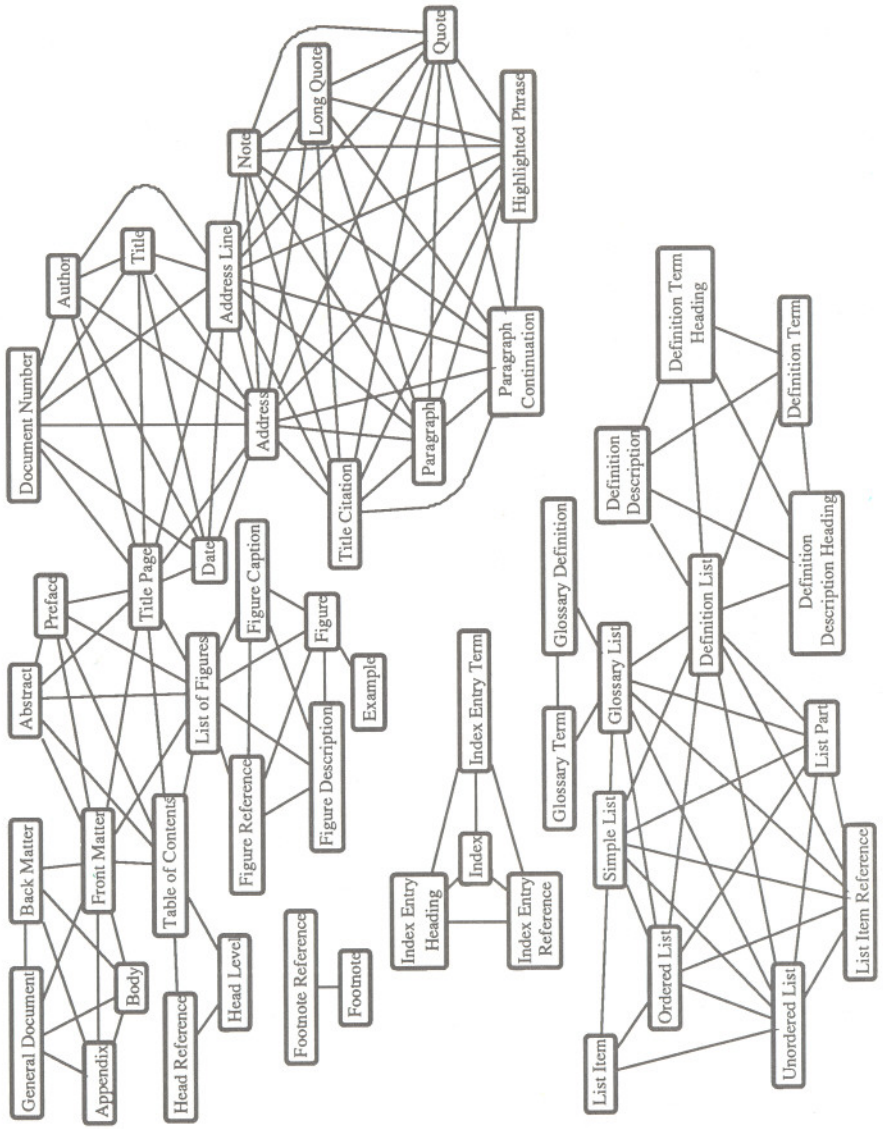
However, another use of Pathfinder analyses is possible. While we agree that advantages arise from structuring the system in congruence with the users' model of the task domain, we do not believe this is strictly necessary for a system to be learnable or usable. Discrepancies revealed by system and user networks can be viewed as imposing a *communicative burden* on the system and the system image: They indicate where designers must take extra measures to convey the system's (deviant) conceptual structure to the user.

By this we do not mean to suggest that discrepancies with the system model are the users' problem, nor to ordain that user models be based on system models. In fact, it seems likely that attempting to migrate the users' model to the system model through interface design, as opposed to redesigning the system to be in congruence with the users' model, is the more difficult alternative. Rather, we mean to emphasize that user/system congruence is the desired goal state, and there is more than one way to support it through system design. McDonald and Schvaneveldt (1988) offer several suggestions about how interface characteristics might be designed to effectively communicate a system's structure. The Pathfinder analysis can indicate where special attention should be given when using such techniques.

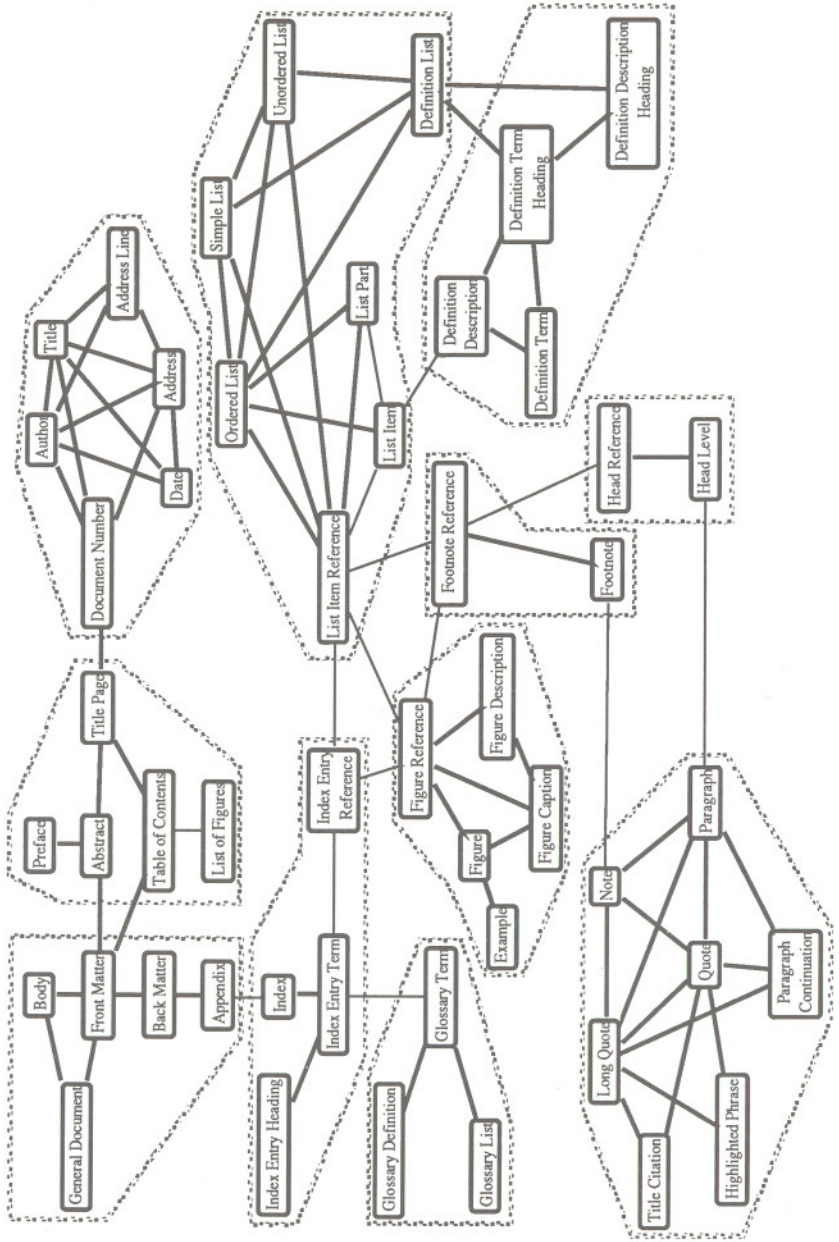
Pathfinder networks can provide a useful summary and representation of the conceptual structure of a system, from both the system's and the users' points of view. The more completely and accurately the analyst models the functional and conceptual structure of the system in the system model, the more the comparison with user models can reveal about

the system's learnability and usability. Pathfinder networks do show the evolution of structure in user knowledge with system experience, and the treatment of individual commands and patterns of links are interpretable. While further work will be needed to use network scaling models as part of an empirical test of the mental model hypothesis (i.e., relating user/system discrepancies to performance outcomes), the results reported here validate the ability of network scaling analyses to reveal potential usability and learnability problems, and lay the foundation for empirical validation of the relationship between a user's declarative knowledge of an interactive system and performance.

Appendix 1 - System Network



Appendix 2 - Experienced User Network



Appendix 3 - Nonuser Network