ELECTRONIC PUBLISHING, VOL. 3(4), 207–225 (NOVEMBER 1990)

# Examining usability for a training-oriented hypertext: Can hyper-activity be good?

TRICIA JONES[1]AND BEN SHNEIDERMAN

*Human-Computer Interaction Laboratory*
*Department of Computer Science and*
*Institute for Advanced Computer Studies*
*University of Maryland*
*College Park, MD 20742, USA*

**SUMMARY**

**We describe the design and evaluation of a hypertext-based tutorial for hypertext authors. This 85-article tutorial represents an innovative application of hypertext to procedural learning. The work has been influenced by Carroll's minimalist model, and by the syntactic/semantic semantic model of user behavior. The usability study involved eight subjects who studied the Hyperties Author Tutorial (HAT) for approximately one hour and then performed a set of authoring tasks in an average of 21 minutes. All users successfully completed the tasks. As a result of the study, we provide a characterization of appropriate uses of hypertext for training, and describe the meaning of a hyper-active environment.**

KEY WORDS Hypermedia Hypertext Hyperties Training

## 1 INTRODUCTION

This paper describes the development of a training-oriented hypertext. The goal was to teach individuals how to use the Hyperties Author Tool for creating their own hypertexts. An innovative aspect of this tutorial was a section which emphasized procedural learning. A usability study was conducted for the tutorial. On the basis of this experience, we will discuss the apparent strengths of hypertext environments for procedural learning, and provide some recommendations for its appropriate use in training.

Hypertext is being applied in diverse settings, including writing [1], auditing [2], architectural design [3], and hypertext books about hypertext [4].

Another suggested use is educational settings. Educators have been attracted to the notion of hypertext by its descriptions, which frequently invoke the notion of "connectivity". The descriptions often have their roots in cognitive psychology, and correspond closely to theories of learning, such as David Ausubel's. He states that in meaningful learning, pieces of knowledge are subsumed into a hierarchical structure, they are "anchored to existing ideational structures" [5]. For meaningful learning to take place, material needs to be integrated with prior learning [6]. As Jonassen states:

[1] Current address: Multimedia Cognition and Instruction Lab, School of Education, University of Michigan, 610 E. University, Ann Arbor, MI 48109-1259, USA

0894–3982/90/040207–19$9.50

*Received 7 July 1990*
*Revised 25 October 1990*

> hypertext designs reflect the semantic network models of memory in network principles for displaying content. The theoretical advantage of hypertexts is the freedom they provide the reader to impose their own structure on text, that is, to access text in a sequence that is most easily accommodated by semantic structure. Readers can more easily access information in terms of what they know, which most cognitive theories of learning aver is the way it must be done [7], p.289

A major benefit of hypertext in instructional settings is the ability to tailor the presentation to the needs of an individual user. The student initiative afforded by the flexibility of hypertext differs from other forms of computer-assisted instruction, in that users are more directly responsible for the tailoring as they browse through a textbase and make selections according to their needs and interests.

Indeed, hypertext has found a diverse range of applications in education, including elementary school science [8], high school [9] and college level [10] literature classes, college biology [11], classics [12], psychology [13], [14], geology [15], writing instruction [16], [17], and history [18].

Almost without exception, the examples given above represent *declarative* forms of learning. Conventional wisdom holds that while hypertext is appropriate for education, it may not match the needs of training, where the emphasis is on procedural learning. A procedure is essentially a sequential process; thus, it is usual in procedural instruction for the units to be strictly ordered. In light of this strict sequencing, the flexibility and “non-linearity” of hypertext seem a bit unnecessary. From this viewpoint, it seems somewhat counterintuitive to model a linear process within a non-linear network. On the other hand, it is likely that a hypertext framework *can* add something to procedural instruction.

## 2 COMPUTER-ASSISTED TRAINING AND HYPERTEXT

One of the driving forces in the development of user-centered design for training software has been the work of John Carroll at IBM. Over the course of many years, he has developed and refined a model he calls the minimalist model of instruction. The basic principles are these:

- slash the verbiage,
- focus on real tasks,
- promote error detection and recovery, and
- guide exploration.

The emphasis is on getting users to be involved in goal-directed, problem-solving behavior using guided exploration. One particular implementation of the model is the Training Wheels Interface. The basis for these interfaces is a mock-up of the system with limited functionality. There are two purposes for the Training Wheels—to direct users along the proper series of steps, and to help users avoid catastrophic errors. Another feature of the Training Wheels Interface is that advanced topics are ignored at a learning stage where they are not necessary. The focus is to insure that learners will understand the *basic* functionality of the system.

As motivation for the use of hypertext in training, consider this statement:

> On-line training is not just a better medium than self-study manuals for presenting tutorials. It is a different medium entirely . . . The challenge in this area is not to migrate existing approaches to user training into a new medium, but to discover the unique strengths of the new medium and the uniquely appropriate techniques for exploiting them in the design of user training [19], p.11

Their use of the word ‘medium’ implies ‘computer-based’, but their vision should also be applied to the medium of hypertext applications. In other words, the challenge is to discover the unique strengths of hypertext and the uniquely appropriate techniques for exploiting them in the design of procedural learning and other instruction.

Hypertext may possess unique strengths that can be applied to procedural learning. However, it is dangerous to try to apply hypertext to every project, as it may be inappropriate for some. These “Golden Rules” may help designers decide where it is appropriate [20]:

- there is a large body of information organized into numerous fragments,
- the fragments relate to one another, and
- the user needs only a small fragment at any time.

In addition to determining whether or not the material is appropriate, it is also important to determine whether or not it will be appropriate for the task and, if so, under what constraints. Along these lines, Stanton and Stammers [21] offer four reasons for using hypertext in training: because non-linear conditions create learning environments that

- allow for different levels of prior knowledge,
- encourage exploration,
- enable subjects to see a sub-task as part of the whole task, and
- allow subjects to adapt material to their own learning style.

With the exception of the third item, the elements of this list are equally applicable to non-training forms of instruction. It is this third item that is a particularly insightful commentary for procedural, training applications: *the ability to see a sub-task as part of the whole task.*

Training applications which have been proposed are simulations (e.g. with HyperCard, as in [21]), interfaces to simulations, and elaborating text for simulations (e.g. MACH-III, [22]). By contrast, we developed a training-oriented hypertext which does not contain a stand-alone simulation, yet models the procedure involved in adding articles to a Hyperties hypertext.

There were two primary motivations for the development of the Hyperties Author Tutorial (HAT). One was to show potential users of the Hyperties package what could be done with the software, by providing ‘on-line’ help for The Author Tool, written with Hyperties itself. To this end, we were interested in determining common authoring errors, in order to determine the usability of the HAT. At a more abstract level, we were interested in the broader question of types of learning and hypertext. In particular, can Hyperties—and hypermedia in general—be used for procedural learning? This problem must be approached with more formal studies; however, by observing patterns of errors, we hoped to determine what the important variables for such studies might be.

## 3 DEVELOPMENT OF THE HYPERTIES AUTHOR TUTORIAL

Hyperties (Hypertext Interactive Encyclopedia System) is based on a comprehensible book metaphor, where the hypertext consists of articles with titles. An article can be one or several screens long and can combine text, graphics, and video. Hyperties allows users to easily traverse a database of articles by merely pointing at highlighted words or images in context (embedded menus). These links can be reversed; other access methods include an alphabetical list of articles (the index), a table of contents, a path history, and full-text search. The Author Tool assists in managing the index during the development process, by providing synonyms, citation cross-references, and author notes.

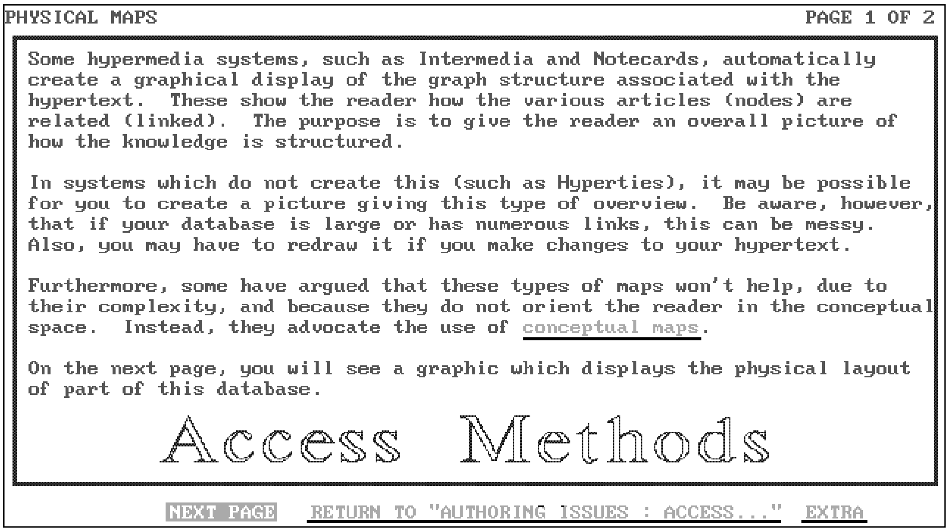
Ideas from Carroll, particularly the notion of a Training Wheels Interface, were used informally in the development of the Hyperties Author Tutorial (HAT). Another important influence is the syntactic/semantic model of user behavior [23]. Within this model, users' knowledge about a system can be divided into two types: *syntactic*, which refers to precise and somewhat arbitrary bits of information about device-dependent details, and *semantic*, which has to do with general concepts independent of any particular system. The semantic knowledge is separated into task concepts and computer concepts; both sets of concepts consist of objects and actions. In order to effectively complete a task, there needs to be a mapping between semantic task and computer concepts. Syntactic knowledge is obtained through rote memorization, while semantic is acquired by meaningful learning.

Based on this, the tutorial was developed with two main sections: the first deals with system-independent issues of authoring hypertexts, while the second deals with Hyperties-specific details; these can be referred to as the *authoring section* and the *training section*, respectively. The authoring section presents the semantic information about authoring hypertexts in general. By contrast, the training section focuses on mapping from these concepts to the semantic computer concepts of the Hyperties Author, as well as presenting some of the syntactic details of the system. Because of its emphasis on procedural learning, the training section represents the more innovative application of hypertext; therefore, it is the focus of this paper.

### 3.1 Description

The target audience for the Hyperties Author Tutorial (HAT) is individuals with a desire to learn how to author with Hyperties, i.e. a novice population. They are assumed to have background knowledge in DOS fundamentals and word processing, know what hypertext is, and know how to use the Hyperties Browser (for each of these topics, however, there was a brief “refresher course”). Basic, system-independent hypertext issues are presented along with Hyperties-specific instruction, with the intent that authors should apply the principles when using Hyperties.

The authoring section and the training section are essentially separate, with very few cross-links between the two. In addition, the general article formats and appearances for each are quite different. The authoring section explains basic concepts, and consists chiefly of text. The training section provides the “how-to” information; therefore, it is more procedurally oriented and contains more graphics—each article contains a mock-up of the Author screen, along with text describing the expected actions of an author at that stage. Highlighted commands are always located at the bottom of the screen mock-up; in some



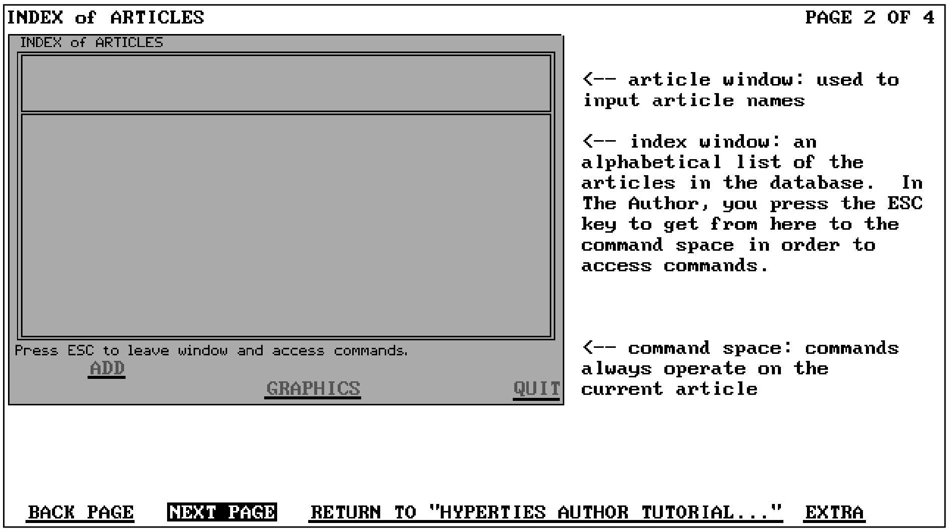
Some hypermedia systems, such as Intermedia and Notecards, automatically create a graphical display of the graph structure associated with the hypertext. These show the reader how the various articles (nodes) are related (linked). The purpose is to give the reader an overall picture of how the knowledge is structured.

In systems which do not create this (such as Hyperties), it may be possible for you to create a picture giving this type of overview. Be aware, however, that if your database is large or has numerous links, this can be messy. Also, you may have to redraw it if you make changes to your hypertext.

Furthermore, some have argued that these types of maps won't help, due to their complexity, and because they do not orient the reader in the conceptual space. Instead, they advocate the use of conceptual maps.

On the next page, you will see a graphic which displays the physical layout of part of this database.

*Figure 1. Screen print from the authoring section:Access Methods: Physical Maps*



Press ESC to leave window and access commands.

<-- article window: used to input article names

<-- index window: an alphabetical list of the articles in the database. In The Author, you press the ESC key to get from here to the command space in order to access commands.

<-- command space: commands always operate on the current article

*Figure 2. Screen print from the training section:Index Screen, page 2*

articles, there are also highlighted terms in the explanatory text. Figure 1 shows a screen print from the authoring section, Figure 2 is a screen print from the training section.

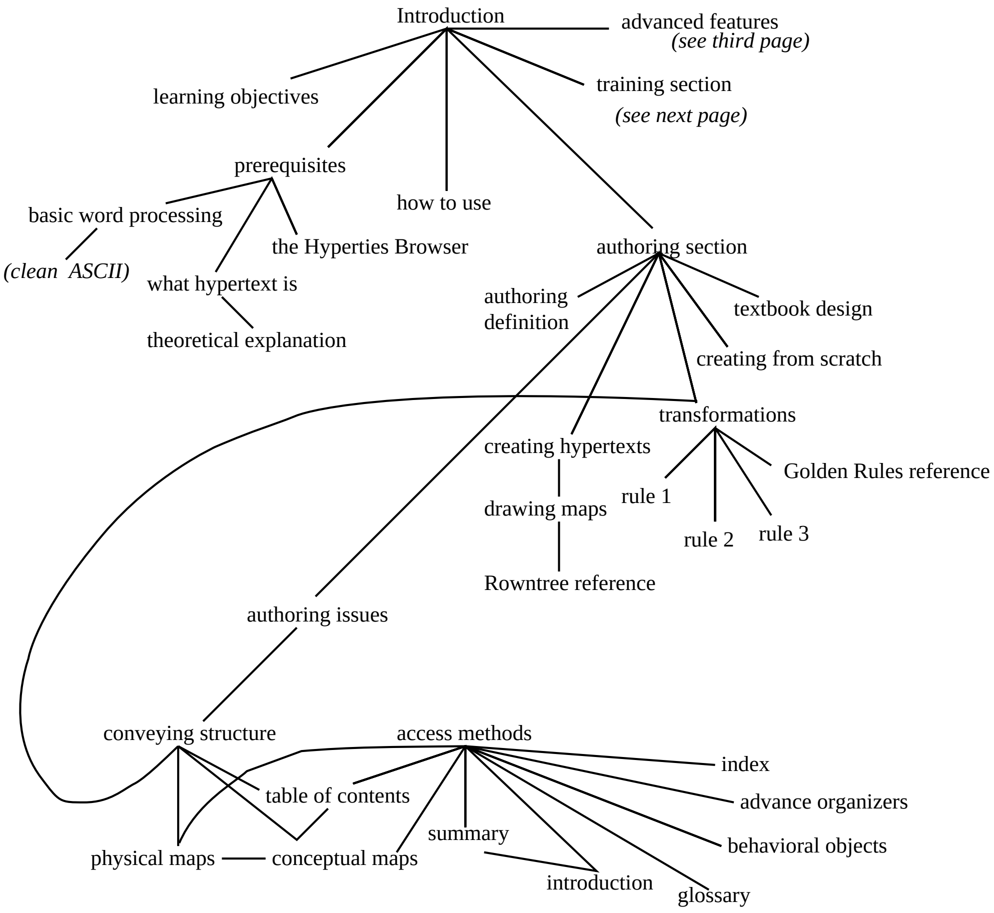
The database consists of 85 articles; 27 articles in the authoring section, 44 in the training section (one overlaps both section), and 15 in the "meta-section" (consisting of the introduction, table of contents, and various other topics such as the "refresher courses" not directly relevant to the other two sections). Within each section, screen design is consistent. 60% (16 of 27) of the articles in the authoring section are only one screen long, the rest are two screens. There are a total of 30 screens of text, and four brief descriptions, in this section. The number of highlighted terms (links) ranges from 0 to 9; the largest number of links is in the **Access Methods** article. In the training portion, 50% (22 of 44) of the articles are only one screen long, with the longest being four screens. The number of highlighted terms ranges from 0 to 16; the largest number of highlighted terms is in the **Advanced Features** article. In the training section, brief descriptions are omitted for the articles contained in the "simulation". This is to reinforce the feeling that what appears on the screen corresponds to what actually occurs in the Author. There is a consistent placement of the screen mock-ups and commands, and the explanatory text is always found in columns 50 through 80. There are 53 screens of text in this section, along with seven brief descriptions. Figure 3 shows the network structure of the HAT.

## 3.2 Design considerations

The best presentation for the training material was not easy to determine. The initial idea was to present the screens, full-size, as they appear in the tool; text explaining each screen would appear within the windows. This idea was discarded, as we realized that it would be very difficult to distinguish between the "meta-information" and what would appear on an Author screen during actual use. We briefly considered creating a simulation of the entire system; this idea was rejected since one of the goals was to use Hyperties for procedural instruction. Finally, the idea of shrinking the screens of The Author Tool into mock-ups was suggested. This way, meta-information could appear on the other side of the screen, and still be distinguishable from the Author snapshots. Furthermore, these screen mock-ups could help the reader coordinate the instructions with the actual environment in which they are used [19].

Following recommendations from Carroll's Training Wheels model, only a limited number of commands are available for selection at each stage. One purpose of this is to limit complexity for the user. Furthermore, in the initial presentation only the basic commands are covered, and are highlighted only where their selection would make sense. Commands are not highlighted until the final page of an article; the purpose of this is to insure that users will read the entire explanation before making selections. Advanced topics are found further down the tree structure of the hypertext. To promote error detection and recovery, certain potential error conditions are explained.

Graphics are used for various purposes. In the authoring section, graphic backgrounds tie together topics across various articles (such as "Access Methods" in Figure 1). In addition to the screen mock-ups, the training section contains graphics which show the results of various operations. This use can be referred to as a "sequence snapshot" as in Figure 4; seven appear within the HAT. Examples include pictures displaying the process of importing an article with the GET command. Graphics were also used to accompany the explanations of the proper structure of a file to be imported, and to describe the formatting commands.



(term in italics indicates article also appears within another section)

*Figure 3(a). Physical map of HAT database (part 1 of 3)*

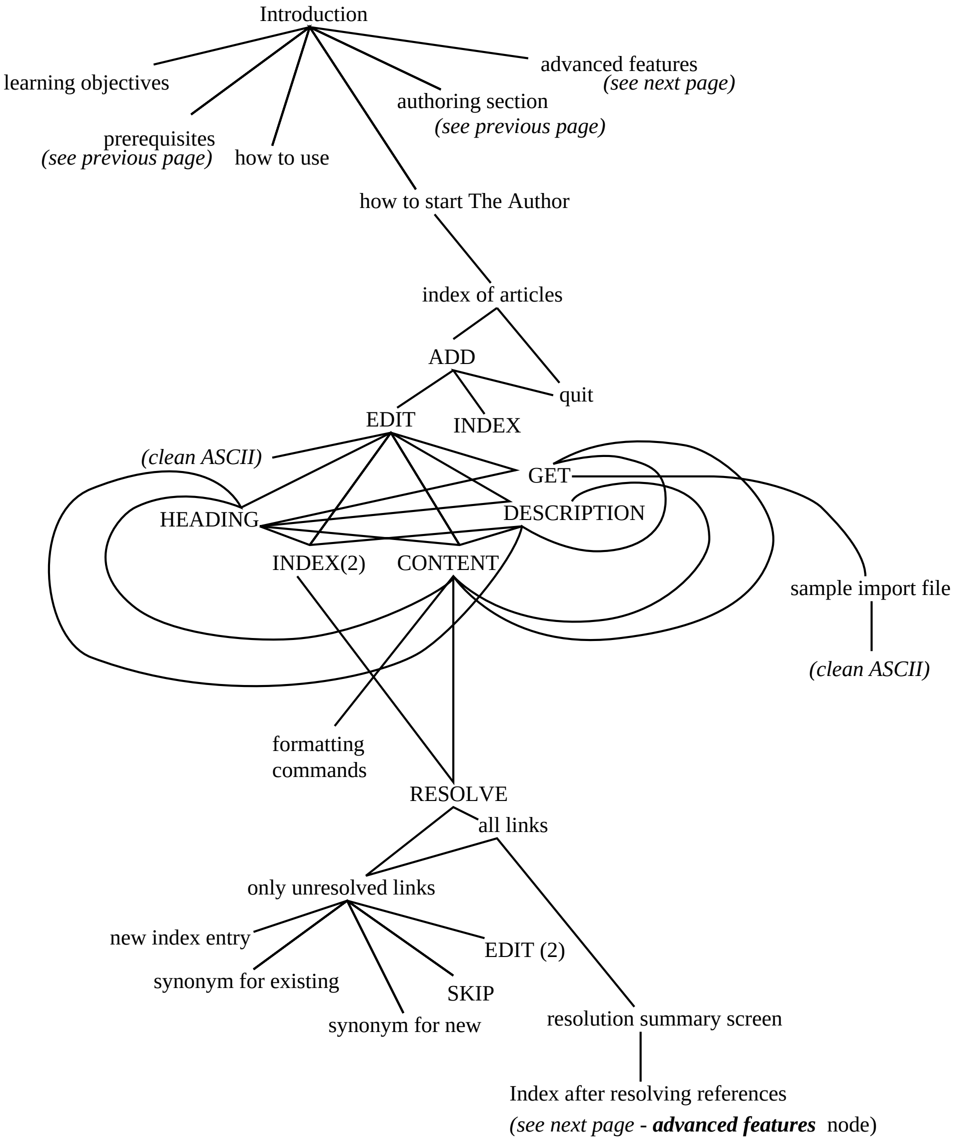


*Figure 3(b). Physical map of HAT database (part 2 of 3)*

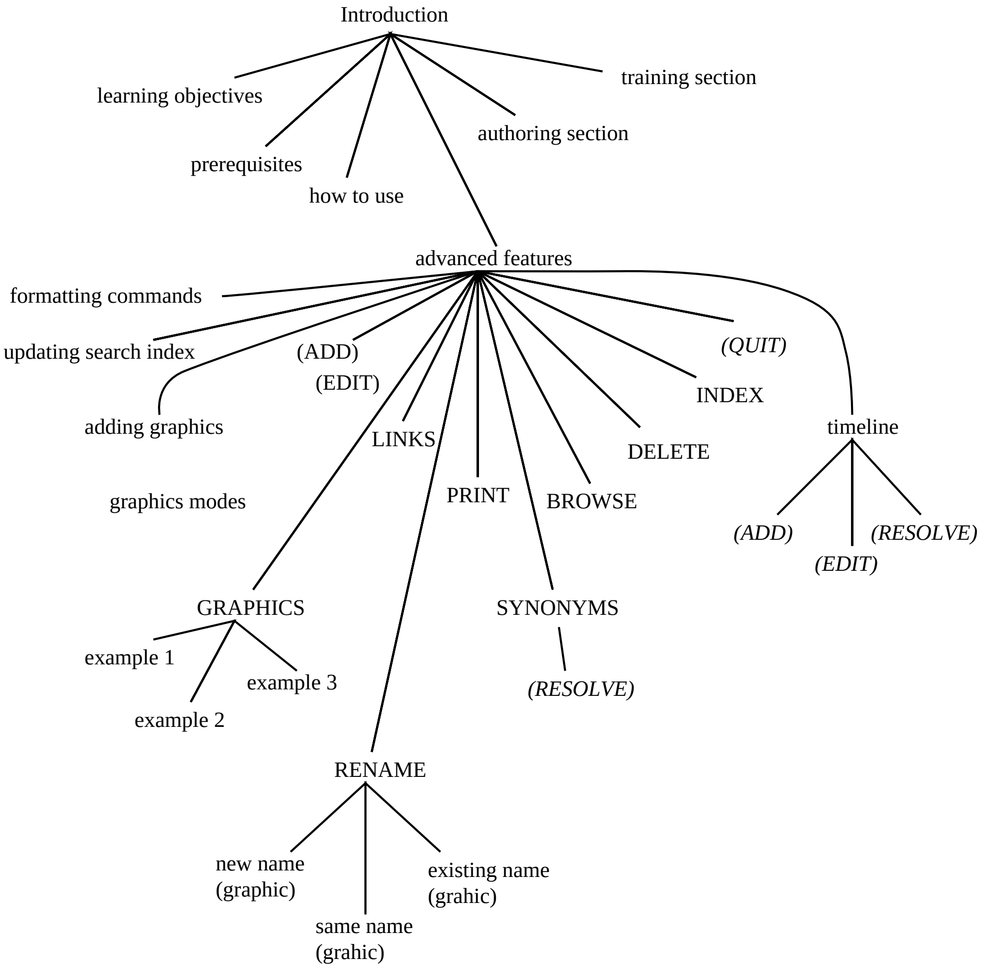
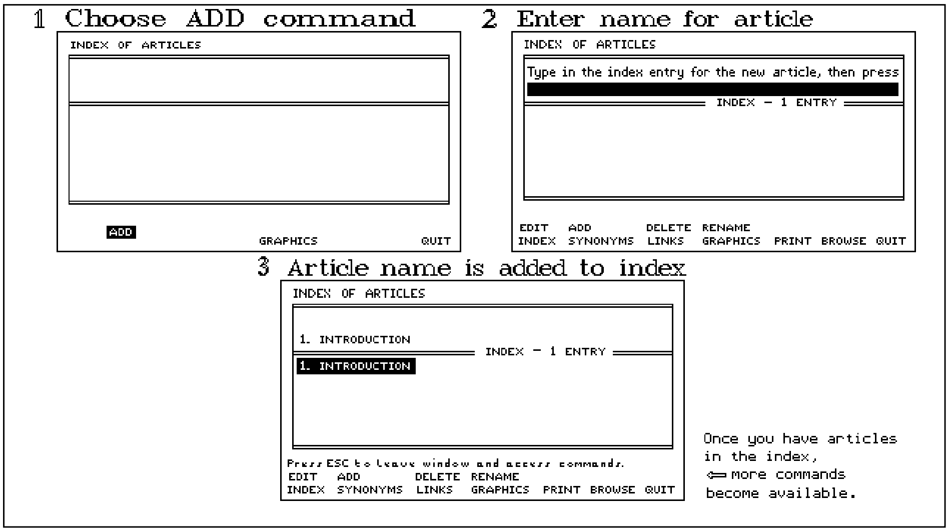


*Figure 3(c). Physical Map of HAT Database (part 3 of 3)*



1 Choose ADD command

2 Enter name for article

Type in the index entry for the new article, then press

3 Article name is added to index

Press ESC to leave window and access commands.

Once you have articles in the index, ⇐ more commands become available.

*Figure 4. Screen print from the training section: Add Command, page 4 (example of a "sequence snapshot")*

## 4 THE HYPERTIES AUTHOR TUTORIAL USABILITY STUDY

To evaluate the HAT in use and determine if the design goals had been achieved, we studied whether or not readers could adequately use The Author Tool after only one hour of HAT browsing. The purpose was not to determine if the HAT is the best approach for training Hyperties authors. Instead, we were interested in determining common errors of subjects who had used the HAT to learn The Author Tool; this information could be used to improve the tutorial. This error data could also be used to make hypotheses about the use of Hyperties for procedural learning.

Three individuals previewed the material. As a result, some critical problems were identified and corrected. In addition, we saw some indication of problems readers might have with the tutorial: misunderstanding how to move among sections of the index screen, feeling uncomfortable "going deep" into a thread of links, and getting fatigued before reaching some critical details.

### 4.1 The usability study

Eight subjects were obtained; these included two undergraduates, three graduate students, an information consultant, and two researchers affiliated with the University; their ages ranged from 21 to 48; three were female (a graduate student, the information consultant, and a researcher). Five subjects had specific projects in mind as a motivation for learning to use The Author Tool. Each of the eight subjects had previously used Hyperties for browsing, and seven had expressed a desire to learn how to use The Author Tool.

The procedure was quite simple: subjects were instructed to browse the HAT database, and informed that they would have to perform a short authoring task at the end. They were told that it should take about one hour to read the database. Subjects were also informed that they might be interrupted during their reading in order to elicit information about choices made.

Subjects read through the database in whatever order they wished. After their reading was complete, subjects were given a list of tasks to complete with The Author Tool:

- Start the Author, on a specific directory
- Edit the content of the introduction, adding a link
- Edit a new article
- Resolve links
- Import an article
- Change a heading
- Change a brief description

These tasks correspond to the typical approach one might take to learning the Hyperties Author Tool, i.e. creating a ‘toy’ hypertext.

After completion of the authoring task, subjects filled out a background questionnaire and a brief subjective questionnaire. The subjective questionnaire concerned impressions of the tutorial, as well as opinions on specific design issues. In addition, subjects were asked to make suggestions for improvement of the database.

### 4.2 Observations

For the first seven subjects, reading times ranged from 45 to 77 minutes, with an average of 57 minutes; the eighth subject was much more thorough and spent nearly two hours reading the database. For the authoring portion, times for six of the subjects ranged from 10 to 20 minutes, while one subject spent 30 minutes in this section and the final one spent 50 minutes; the average time was nearly 21 minutes. Subjects generally spent more than half their time exploring within the training section. Four of the subjects used the index at the end to insure that they had read all the articles in which they might be interested.

In general, subjects seemed to enjoy the database, although they frequently expressed fatigue near the end of the training section. While browsing, there was a group of readers who followed a very systematic approach, accessing links in order as they appeared in the text. Those who had the most experience with Hyperties seemed to be the most uncomfortable going deep along a branch of articles. For example, the basic sequence in placing an article in the hypertext uses the commands Add-Edit-Resolve (Figure 5). Within the tutorial, this complete path has a minimum of 7 articles, with 9 or 10 being more likely; in addition, at each stage, there are numerous length-one paths to view. While following this path, three subjects mentioned “I need to go back and read GET first”. This feeling of disorientation might have been avoided if the timeline had appeared first, so the reader would have known what to expect.

Although Hyperties browsers tend to be systematic in their choices, this should not

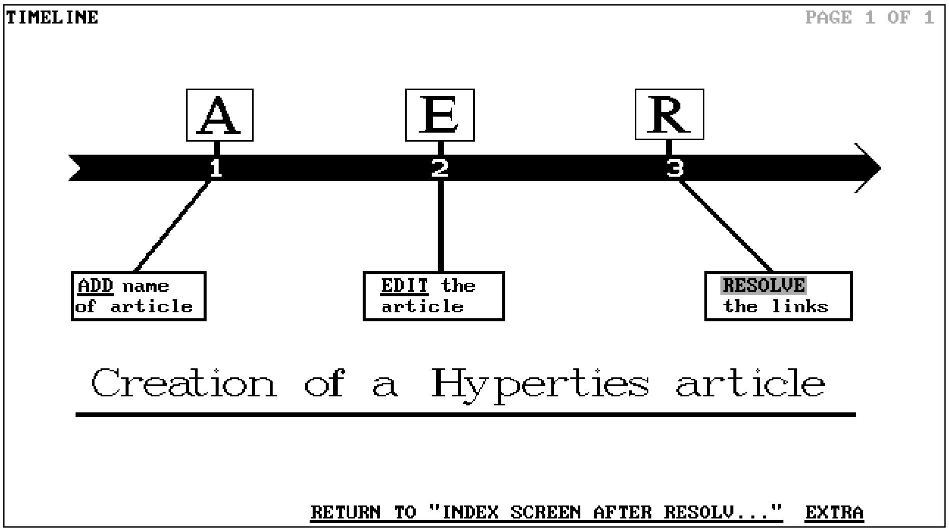


*Figure 5. The Timeline article*

imply that their individual reading sequences are alike. Upon examining partial paths that the subjects took through the hypertext, certain common elements can be recognized, but no two readers had identical browsing patterns.

### *4.2.1 Paths*

The main sections were presented in the following order on the second page of the **Introduction**: the authoring section, the training section, and the advanced concepts portions of the training section. Six of the eight read in that order. One read the training portion before the authoring section. The remaining subject read the authoring section, and then the advanced features section instead of the initial training portion. This was not reflected in the performance of these two.

In the training portion, the first article is the index screen. Two commands are selectable here: ADD and QUIT. Once ADD has been chosen, EDIT and INDEX also become selectable. There were four distinct paths through this section taken by the subjects:

| | | |
|---|---|---|
| 4 | Add - Edit | Subjects 1, 2, 3, 4 |
| 1 | Quit - Add - Edit | Subject 6 |
| 1 | Add - Index - Edit | Subject 5 |
| 1 | Add - Quit - Index - Edit | Subject 7 |

Subject 8 did not select any of the links from the index screen; instead, she returned to the **Introduction** and then proceeded to the **Advanced Features** screen.

The edit screen presented numerous options for paths. Five commands were accessible: HEADING, DESCRIPTION, CONTENT, GET, and INDEX. The first four are the most important; the final one elaborates on the error-recognition message presented by Hyperties.

The next step in the basic loop, represented by the **Resolve** article, is only accessible from **Content** or **Index**. Again, there were four distinct paths:

| | | |
|---|---|---|
| 5 | Heading - Description - Content - Resolve | Subjects 1, 2, 3, 7, 8 |
| 1 | Get - Heading - Description - Content - Resolve | Subject 5 |
| 1 | Heading - Get - Description - Content - Resolve | Subject 6 |
| 1 | Heading - Description - Get - Content - Resolve | Subject 4 |

What is interesting about these paths is that they are identical, except for the position of the GET node. The most common path was a "depth-first" attack on the links, where links are selected based on their screen position, and followed until closure was reached. This was also true of the most common sequence at the index screen.

From **Resolve**, it is possible to eventually arrive at the **Advanced Features** screen, which serves the dual purpose of reviewing of the basic process and presenting advanced commands and topics. There was more individual variation of choices at this screen than at either the initial index screen or the edit screen. It is impossible to tell whether this is a result of the number of links, fatigue of the readers, their interest, or other factors. A few patterns of use were observed. The **Timeline** link was the first highlighted term in the article. Five subjects indeed viewed the timeline first. Two subjects seemed to be very systematic about their approach to this screen. One subject read all the links within the text in order, and then proceeded to read the command links on the graphic in order (left to right, top row before bottom row). Another first viewed the timeline, then examined the new commands from left to right, then proceeded to read the advanced topics (those not on the screen mock-up, but listed at the end of the text).

### *4.2.2 Other browsing issues*

The level of engagement with the material varied between subjects. One subject took notes on several topics. Another subject was actively trying to "make sense" out of the material in the training portion, such as trying to figure out the various resolve options on the basis of their names. A different subject engaged in obvious "metacognitive monitoring"; that is, he would frequently return to the **Learning Objectives** article to see if he had covered all the important material. In addition, after reading about an access method, this subject would go to the Index to see if that type of article appeared in the hypertext; he was particularly disappointed that there was no article entitled **Summary**. This subject was also the only one to use the **Table of Contents**, directly as a result of this method of attack. It appeared that graphics such as the sequence snapshots resulted in a more active level of interaction. Otherwise, subjects were fairly passive.

In general, the subjects felt comfortable in browsing by following links and thus did *not* take advantage of the access methods (history, search, table of contents) provided by Hyperties. Only two subjects used the **History** function; one used it to recover when he felt lost, while another used it to return directly to a higher level in a thread of articles. Likewise, only two used the **Search** command. One wanted to see if there were more articles about synonyms, since he was confused about the topic. The other had a different purpose: upon completion, he mentioned that the HAT would be useful for reference while working on a hypertext, and decided to look up the word "import" to show how he intended to use it.

At only a few places did subjects take advantage of the fact that not all links in a hypertext have to be read. For example, at the RESOLVE stage, one subject figured out how each of the resolve options worked without reading them. On the basis of the names, some subjects decided to not read all of the advanced commands, since their meaning appeared to be straightforward.

Some subjects appeared to be overwhelmed when a large number of choices appeared on the screen. This was most pronounced at the **Advanced Features** screen, as all eleven commands on the index screen were highlighted, along with four more links on the second page of the article. The readers were beginning to feel comfortable with the basic concepts at this point, but tended to be fatigued. Thus, the appearance of fifteen links tended to be intimidating.

### *4.2.3 Task*

When attempting the authoring task, every subject remembered the basic sequence of Add-Edit-Resolve. Overall, subjects were successful in using The Author Tool to accomplish the required tasks. Detailed reports on each subject are available in Reference [24].

It is possible to examine the errors made in light of the syntactic/semantic model. In this model, there are two basic types of errors: syntactic and semantic, and there are two types of semantic errors—improper mappings from task-concept semantic representations to the computer concept, and improper mappings from the task (problem solution) to the task-concept semantic representation. One subject spent up to five minutes looking for an IMPORT command within the edit screen, when the command is called GET. He knew it had to be on the edit screen, but he did not remember the exact word; this was clearly a syntactic error. Another syntactic error occurred when a subject tried to add a link within a brief description; this is perfectly reasonable semantically, but not within the system.

Four of the subjects were not able to get from the resolution summary screen to the index; instead, they returned to the edit screen. The most obvious explanation is this: the default action is to return to the edit screen, while an extra step is necessary to select the index. This would be a syntactic error. For one of the subjects, however, the error did not occur until his third editing sequence. This seems to be a clear-cut example of the first type of semantic error, as he forgot the precise mapping to the computer concepts. Many subjects misunderstood the directions to “change the title of any article” and used the RENAME command instead of changing the HEADING at the edit screen. This improper use of commands can be seen as an example of the second type of semantic error. The subjects were familiar with Hyperties articles, but not necessarily the internal “structure” of an article.

## 5 BEYOND CAI: A NOTION OF HYPER-ACTIVITY

Based on our experience, we believe that it is reasonable to use a hypertext framework to facilitate some types of procedural learning. Much more research needs to be done to determine the most appropriate uses; however, after examining patterns of usage, errors, and difficulties, we offer the following suggestions.

The use of the word *hyper-active* expresses the unique strengths that hypertext can provide for user training. The unique characteristics of hypertext, not offered by traditional computer-assisted training, are non-linearity, connectivity, and reversibility. These combine to give the learner a greater level of control over the presentation of material. The term *hyper-activity* in this context should imply a sense of interaction within the framework of a hypertext network. It differs from a traditional branch-and-bound tutorial in that users can choose which node to view next. Of course, in a system with preset links, the author has ultimate control over the material. However, with the flexibility of hypertext, users are given a greater control over the presentation order. In addition, users are more aware of how their actions affect what is viewed next.

In terms of the syntactic/semantic model of user behavior [23], hyper-activity seems most appropriate for linking semantic task knowledge to semantic computer concepts. The best match is when the task is essentially linear, and at each stage has multiple options which can be explored. The main issue for authors is keeping the reader moving along the linear path. The key feature seems to be having graphical overviews, which enable subjects to see a sub-task as part of the whole task. For lower-level, syntactic details, learners really need hands-on experience in order for the training to be memorable; at such times, a simulation is necessary.

These recommendations are derived from our observations. Unless there are multiple options at some of the stages, the flexibility of hypertext does not add much. Users of the HAT all found the **Timeline** to be beneficial. Syntactic details were easily forgotten, we hypothesize that this is because the reader was not directly responsible for causing the action.

The Hyperties Author Tutorial represents an appropriate use of hypertext for this procedural learning situation. There is one fundamental process which must be followed: name the article (with the add command), edit it, then resolve the links in it. At each of these stages, however, there are numerous options that can be exercised. Consider these examples: at the edit screen, there is no one correct order for accessing the heading, description, or content. The same is true of resolving references: there are five options, but there is no canonical order for learning about them. A final example is that of the (advanced) index screen, where the commands and other topics can be explored under complete user control. These examples express the concept of a *hyper-active* training style.

## 6 RECOMMENDATIONS

Based on the limited nature of our study, it would be premature to offer more than tentative suggestions. More studies should be done, with other topics, more subjects, differing methods, etc. We hope that the recommendations provided here will lead others to reflect on the use of hypertext for procedural learning in the interim.

### 6.1 For the improvement of the HAT

For the HAT to serve as a tutorial for new authors, there are some changes that should be made. A key issue is the placement of the timeline. One previewer suggested adding a graphic overview to present the basic process for adding articles to the database. At his suggestion, the **Timeline** (Figure 5) article was added to the hypertext, accessible from the **Index Screen After Resolving References** article. Originally, this article served as a summary for the add-edit-resolve process. However, five subjects suggested that it be presented before the main portion of the tutorial; in other words, as an advance organizer.

It might then serve to direct readers along that path, and allay fears they might have about “going too deep”. As a related issue, it might be wise to redraw the timeline in order to clarify that the process is a cycle, i.e. that it is important to return to the index before continuing to edit articles.

Links added between the authoring and training sections would better associate the semantic task and computer concepts, without sacrificing the system-independence of the authoring section. At the same time, Hyperties ‘syntax’ would be reinforced.

Shortly before the evaluation, the advanced commands and some advanced topics were added at the end of the add-edit-resolve loop. This screen was also directly accessible from the **Introduction**, through the **Advanced Features** link. The intention was to give readers the opportunity to continue exploring once they were familiar with the basics. In reality, readers were often fatigued at this stage, and seeing myriads of highlighted terms tended to overwhelm them; in addition, most readers seemed to feel obligated to read all the links. A better design would entirely separate the advanced features from the initial loop of basics, while maintaining a link between the two. Perhaps the **Introduction** could suggest to readers to try out their newly obtained knowledge in The Author Tool in order to cultivate their basic skills, before returning and reading the advanced topics. The summary could also point to the advanced topics. If there were a simulation capability in Hyperties, the end of the basic loop would be an ideal place to simulate The Author Tool, Another possible place is after readers have been introduced to the add-edit-resolve loop, to reinforce the concepts. It would also be interesting to port the tutorial to the Sun version of Hyperties, since on that platform, the Author and Browser can run side-by-side. Such a move would help determine whether the simulation would be useful, also, it would make it possible to test the usability of the HAT as an on-line manual.

Several of the subjects did not choose the term INDEX off the edit screen, or from the resolution summary screen. When asked about this, most expressed the sentiment that “it’s probably just the index that I’ve already seen.” However, that was not true. In each location, the term links to a different article: at the edit screen, an explanation describes when it is appropriate to select index within that context; at the index screen, the explanation tells why the selector bar is returned to the index window; and at the resolution summary screen, the **Index After Resolving References** article presents a summary of the basic steps. Usually, it is a bad idea to use the same phrase to link to different locations within Hyperties, for precisely this reason. In the HAT, it was necessary, for the command has different implications at each location. A method to lead the reader to see these differences would be beneficial. One way to address this problem would be to give more direct clues in the text, in effect to say “you should select the INDEX command now . . . ” Nielsen [25] presents the idea of footprinting, whereby links are “checked off” after they have been read. In our situation, adding checkmarks to the screen would have reduced fidelity of the screen mock-ups to the actual tool; furthermore, it would have cluttered the screen a great deal—there is simply no room in the bottom line of the screen mock-up for any more text.

### 6.2 For the improvement of the Hyperties Browser

As a result of this study, some deficiencies in Hyperties were observed. Those discussed here can also be considered relevant to hypertext systems in general.

Multiple indices would have been helpful in the HAT. Since the Hyperties index indicates which articles have been read, subjects most familiar with Hyperties would check their progress by looking at the index. However, there was no natural way to separate the two sections of the tutorial within the index. Therefore, a reader could observe “I seem to have read about a third of it”, but still not know if he had read the entire authoring section. Nielsen [25] refers to this as a context-in-the-large problem. Separate indices for the authoring section and the training section would have reduced this problem, by giving the reader a flavor of how much content had been read. An iconic representation of how much of the database had been viewed is another idea. A bar graph, similar to a fuel gauge, may not be compatible with the view that hypertext is “non-linear”. An alternative idea is a pie chart; slices of pie representing different sections could be of differing sizes, indicative of their size relative to the whole hypertext. An unresolved problem with this idea is that there would be no way for the reader to determine how much of the unread content had been skipped on purpose.

In this hypertext, a simulation that allowed tutorial users to practice steps while they read would have been beneficial. While reading about the resolve options, one subject said it would make more sense if she could try them out and see what each did. Another mentioned “now would be a good time to test”; his suggestion for improvement of the database was to provide feedback tests on some of the important functions. This is impossible to do in the current Hyperties Browser, since it only allows a reader to make embedded menu choices. However, a new version of Hyperties (Hyperties 3.0), due to be released in late 1990, allows an author to branch to external programs, such as simulations. This feature would have been appropriate for the HAT.

### 6.3 For future researchers

We initiated this inquiry to discover whether or not the HAT was effective as a one-hour tutorial for Hyperties authors. To fully evaluate the tutorial, and to evaluate the efficacy of hypertext for procedural training, many other studies should be conducted.

The present study did not focus on whether or not the HAT was the best method for training Hyperties authors. In order to do that, studies could compare the HAT with other training methods for The Author Tool. One type of media comparison could compare the database with the Hyperties manual; however, since the text is quite different, this might not be a fair comparison. To maintain consistency of content, one could compare the HAT with a printed version; however, this becomes difficult for readers to follow the links, especially with the three distinct uses of the term INDEX. Yet another idea is to compare its use to that of a short training videotape. It might seem that the videotape would win; however, since a videotape would be even less interactive than the HAT, it might not. In addition, more realistic authoring tasks could be given. A simple way would be to give tasks which check more aspects of the authoring process, for example, requiring that they make various types of links besides *new entry*. A long-term study where subjects work on their own projects may be more illuminating.

It is important to determine the best function of the HAT. One treatment group could read the basic section, then experiment briefly with the actual tool, and then read the advanced sections. Another treatment group could explore the tool first, then browse the database, and finally use the tool again. Would the second group form incorrect habits as a result of their initial undirected exploration? Such an interaction may change their approach to the topics in the database.

It could be argued that the suitability of the HAT was solely as a result of the self-referential character of the material. However, we believe this does not reflect the true situation. The Hyperties interface is unobstrusive enough to support a similar presentation of other software packages, and other types of procedures. The visual reinforcement provided by the screen mock-ups was probably more important than the self-referential quality. It may even be that the self-reference hindered learning of syntactic details.

There are many basic issues surrounding reading from hypertext that still need to be determined. In this study, we observed that some readers tended to be systematic in their reading, while others were not. What is interesting is that the most novice browsers were the least systematic. Are there optimal reading styles for non-linear text? Or, more basic, how does comprehension of non-linear text proceed? It seems that the eventual reaction to an environment that tries to promote free-form access is to force familiar sequential styles upon it. Of course, many other factors not controlled in this study could be responsible for this effect. It would be interesting to see if reading patterns are affected by cognitive learning styles, by personality, by experience with the system or with the topic, by the writing style used in the database, or perhaps by a totally different factor. The answers to these questions will require much more in-depth study.

Numerous questions remain about the efficacy of hypertext and hypermedia for procedural learning. We should take Carroll's challenge seriously: do hypermedia provide unique strengths, or will we merely be migrating existing approaches?

As a practical matter, experimenters who study the usability of hypertexts should take advantage of think-aloud strategies. When reviewing observations days after a subject finished browsing, it is beneficial to determine why certain choices were made; unless the subjects were vocal about their reasons, there is no way to make these conclusions. Another helpful feature is the automatic logging of selections made by the subjects, as we have done in other studies [4]. Even without verbalization, patterns of usage can lead to insight about the system.

## 7 CONCLUSION

In conclusion, although there are areas that need improvement, the HAT is usable for training prospective Hyperties authors. Subjects enjoyed using the HAT. Although subjects' paths through the hypertext were similar, they did approach the material in idiosyncratic manners. The training section is good as a hypertext because of the control the reader has over the presentation material. The feeling of *hyper-activity* is an asset for this material, because of the flexibility at each step along the basic loop. If even more advanced material were included, the benefits would be even greater, as not all features are needed by all authors.

This Hyperties-based tutorial offers flexibility not afforded by a linear version of the manual or even by traditional computer-assisted instruction: the use of graphics to display procedures, easier access to information about each command, a more realistic feel, and the ability for the reader to set the pace and direction of exploration. Whether or not this is more effective has yet to be investigated; however, it has been shown to be usable.

## REFERENCES

1. J. Smith, S. Weiss, and G. Ferguson, 'A hypertext writing environment and its cognitive basis', in *Proceedings of Hypertext '87*, pp. 195–214, (1987).

2. L. DeYoung, 'Hypertext challenges in the auditing domain', in *Proceedings of Hypertext '89*, pp. 169–180, (1989).
3. G. Fischer, R. McCall, and A. Morch, 'Janus: Integrating hypertext with a knowledge-based design environment', in *Proceedings of Hypertext '89*, pp. 105–117, (1989).
4. B. Shneiderman and G. Kearsley, *Hypertext Hands-On! An introduction to a new way of organizing and accessing information*, Addison-Wesley, Reading, MA, 1989.
5. D.P. Ausubel, 'Cognitive structure and the facilitation of meaningful verbal learning', in *Readings in the Psychology of Cognition*, eds., R.C. Anderson and D.P. Ausubel, Holt, Rinehart, and Winston, New York, 1965.
6. D. P. Ausubel, 'In defense of verbal learning', in *Readings in the Psychology of Cognition*, eds., R.C. Anderson and D.P. Ausubel, Holt, Rinehart, and Winston, New York, 1965.
7. D. H. Jonassen, 'Hypertext principles for text and courseware design', *Educational Psychologist*, **21**(4), 269–292, (1986).
8. M. Scardamalia, C. Bereiter, R. S. McLean, J. Swallow, and E. Woodruff, 'Computer-supported intentional learning environments', *Journal of Educational Computing Research*, **4**(1), 51–68, (1989).
9. G. Solomon, 'Heard it, read it, and saw it on the grapevine', *Electronic Learning*, 18–19, (May 1989).
10. G. P. Landow, 'Relationally encoded links and the rhetoric of hypertext', in *Proceedings of Hypertext '87*, pp. 331–344, (1987).
11. W. Beeman, K. Anderson, G. Bader, J. Larkin, A. McClard, P. McQuillan, and M. Shields, 'Hypertext and pluralism: From lineal to non-lineal thinking', in *Proceedings of Hypertext '87*, pp. 67–88, (1987).
12. G. Crane, 'Challenging the individual: The tradition of hypermedia databases', *Academic Computing*, 22–38, (January 1990).
13. N. V. Hammond and L. J. Allinson, 'Development and evaluation of a cal system for non-formal domains: The hitch-hiker's guide to cognition.', *Computers in Education*, **12**(1), 215–220, (1988).
14. N. V. Hammond and L. J. Allinson, 'Travels around a learning support environment: Rambling, orienteering or touring?', in *CHI'88 Conference Proceedings: Human Factors in Computing Systems*, eds., E. Soloway, D. Frye, and S.B. Sheppard, pp. 269–273, (1988).
15. T. Mayes, M. Kibby, and H. Watson, 'Strathtutor: The development and evaluation of a learning-by-browsing system on the macintosh.', *Computers in Education*, **12**(1), 221–229, (1988).
16. C. Neuwirth and D. Kaufer, 'The role of external representations in the writing process: Implications for the design of hypertext-based writing tools', in *Proceedings of Hypertext '89*, pp. 319–341, (1989).
17. N. Streitz, J. Hannemann, and M. Thuring, 'From ideas and arguments to hyperdocuments: Travelling through activity spaces', in *Proceedings of Hypertext '89*, pp. 343–364, (1989).
18. A. Newmark, 'The election of 1912: A hypertext simulation', *Electronic Learning*, 54–55, (March 1989).
19. J. M. Carroll and S. A. Mazur, 'Learning to use an office system with an on-line tutorial', Technical Report 47740, IBM, (1984).
20. B. Shneiderman, 'Reflections on authoring, editing, and managing hypertext', in *The Society of Text*, ed., Ed Barrett, pp. 115–131, MIT Press, Cambridge, MA, 1989.
21. N. A. Stanton and R. B. Stammers, 'Learning styles in a non-linear training environment', in *Hypertext: State of the Art*, ed., R. McAleese, pp. 114–120, Intellect Limited, Oxford, 1990.
22. J. Psotka, 'Advancing the mind/machine interface: Qualitative simulations, hypertext, and natural language processing', in *Proceedings of the 33rd Annual Meeting of the Human Factors Society*, (October 1989).
23. B. Shneiderman, *Designing the user interface: Strategies for effective human-computer interaction*, Addison Wesley, Reading, MA, 1987.
24. T. Jones, *Hypertext for instruction: Design, evaluation, and theory*, Master's thesis, University of Maryland, 1990.
25. J. Nielsen, 'The art of navigating through hypertext', *Communications of the ACM*, **33**(3), 296–310, (1990).