User Modeling and User-Adapted Interaction 4: 1–19, 1994.

Metadoc: An Adaptive Hypertext Reading System

CRAIG BOYLE
Information Engineering, P.O. Box 180021, Austin, Texas 78718, USA;
e-mail: boyle@cactus.org

and

ANTONIO O. ENCARNACION
Department of Computer Science, Texas A&M University, College Station Texas 77843, USA

(Received: 21 February 1992; in revised form: 31 January 1994; accepted: 2 February 1994)

Abstract. Presentation of textual information is undergoing rapid transition. Millennia of experience writing linear documents is gradually being discarded in favor of non-linear hypertext writing. In this paper, we investigate how hypertext – in its current node-and-link form – can be augmented by an adaptive, user-model-driven tool. Currently the reader of a document has to adapt to that document – if the detail level is wrong the reader either skims the document or has to consult additional sources of information for clarification. The MetaDoc system not only has hypertext capabilities but also has knowledge about the documents it represents. This knowledge enables the document to modify its level of presentation to suit the user. MetaDoc builds and dynamically maintains a user model for each reader. The model tailors the presentation of the document to the reader. The three-dimensionality of MetaDoc allows the text presented to be changed either by the user model or through explicit user action. MetaDoc is more a documentation reading system rather than a hypertext navigation or reading tool. MetaDoc is a fully developed and debugged system that has been applied to technical documentation.

Key words: Hypertext, adaptation, user expertise, stretchtext, evaluation, online documentation

1. Introduction

1.1. Hypertext and Adaptive Hypertext

Hypertext (Nielsen, 1990) provides a means of flexibly organizing and presenting information. The writer of a document need not be rigid in his/her organization of text, since chunks of information can be linked. The reader manipulates the order of text presentation through link selection. In this sense, hypertext is an advance over the essentially linear nature of paper text or simple online documentation. However, hypertext is not an *active participant* in the reading process since the reader is required to adapt and select without assistance from the computer.

This work augments the notion of non-linear hypertext documentation. The addition of a user-modeling component through a system called MetaDoc (so called because it has knowledge about documents) allows the computer to actively participate in the reading process.

Conventional hypertext offers flexibility at the network level: traversal of links allows different nodes to be selected and a 'view' of the entire document to be presented. Another dimension of adaptivity is flexibility at the node level. Manipulation of the text inside a single node allows a greater degree of control over the information presented and hence a more readable document. Stretchtext (sometimes called replacement text) permits such node level flexibility (Nelson, 1971). MetaDoc uses stretchtext to implement its user modeling capability. Because of MetaDoc's heavy use of stretchtext, we refer to it as a three-dimensional writing style.

A user-modeling component augments a hypertext document by automatically adapting a document at the node level. A user model (Kobsa and Wahlster, 1989) contains a representation of the reader's knowledge. The model alters the amount of information presented by automatically stretching or contracting the text in a node – this is the essence of MetaDoc. The advantage of such an 'automatic stretchtext' system is that the reader no longer needs to adapt to a document. Instead, the document adapts to the reader and hence the computer actively participates in the reading process. Such active computer mediation (Reinking and Schreiner, 1985) should aid in the reading process.

An adaptive document allows different user-ability levels to be accommodated by a single document. It also allows for a reader to have an uneven scope of knowledge. To some extent, conventional hypertext allows the *user* to adapt to a document by browsing at the network level. *MetaDoc* adapts to the user through automatic node-level operations.

1.2. Goals of this Research

Our first goal is to build a hypertext document that automatically adapts to the ability level of the reader. Thus, only one document would be needed for all classes of potential readers. Each reader should feel that the document is personalized to his/her ability level. Ideally, no reader would feel the need to skip text (because the document is too trivial) or seek another document (because of complexity). Our domain is technical documentation, chosen because of the wide range of readers and its regular structure.

Our second goal is to evaluate the resulting work to discover whether such an adaptive document improves the productivity and the satisfaction of both domain-novices and domain-experts.

2. Related Work

In this section, we review previous work concerning stretchtext in several guises and briefly consider user modeling work upon which adaptive system behavior is based.

Ted Nelson, the hypertext pioneer, coined the term stretchtext in his article 'Computopia and Cybercrud' (Nelson, 1971). In this form of hypertext, 'the reader can control the amount of detail to suit himself, as he pulls on a throttle or some other control, additional words and phrases appear on the screen, and the rest move apart to make way; as he pushes the throttle in the other direction, words and phrases disappear, and the rest of the text slides back together.' Stretchtext allows the depth of information presented in a node to be varied. But there is more to it than just varying the number of words or the expansion and compression of a node. To minimize disorientation and maximize user satisfaction smooth transition between levels of stretch is essential.

VisiDoc, developed by Ken Bice in 1987 for on-line help on the TI Explorer Series, used the stretchtext metaphor (Schnase *et al.*, 1988; Texas Instruments, 1988). 'Stretching' text in whole paragraphs or sections (not words or phrases) gives the effect that a whole node had been replaced. Outline processors have similar limited stretchtext features. Since this stretchtext is very chunky, it resembles general hypertext with 'goto' links. In a hypermedia usability study, users found that stretchtext (or replacement buttons) is easy to understand and can improve the usability/readability of the document (Nielsen and Lyngbaek, 1989). These systems used stretchtext instead of 'goto' links.

Guide (Brown, 1987) is another commercial hypertext system that provides stretchtext, referred to here as replacement-buttons. According to Brown, documents with replacement-buttons can be presented in a form suitable to a wide range of readers. Since the text is replaced in-line, all material is seen in context.

The dynamic table of contents used in Superbook (Egan, 1989) , DynaText (Dynatext, 1990) and the IBM Operating System/2 on-line documentation (IBMb) display varying amounts of detail chapter and section headings. This is a very limited form of stretchtext.

User modeling is used by an application to explicitly store information about its users to improve interaction. The information stored is used to adapt computer behavior to the user, making the application easier to use and enlarging the population of potential users.

3. Metadoc

MetaDoc, as indicated in Figure 1, has conventional hypertext capabilities: users can traverse links between nodes, backtrack, follow the network linearly, and search for text. Links can be stretchtext operations concerning detail or expansion, glossary accesses and conventional 'jumps' (reference links).

MetaDoc's adaptive reading capability operates through the use of an Interactive Agent, allowing knowledge about the reader of the document to be stored and used to vary the level of detail in the document presented. If the user explicitly modifies the level of detail presented, then the user model is informed and may decide to vary presentation of future information. Thus, by noting previous user actions, it automatically calculates the best view of the document for the user. MetaDoc's Intelligent Agent stores information about the reader in the form of a user model. Figure 2 shows an overview of the MetaDoc system.

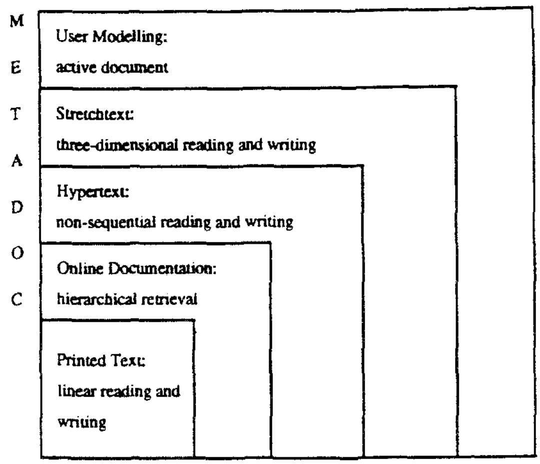

Fig. 1. MetaDoc's relation to other documentation forms

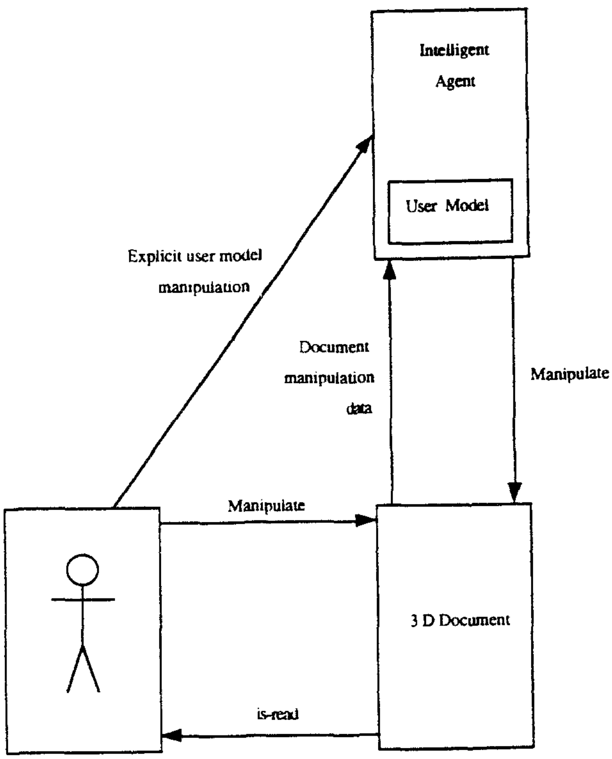

Fig. 2. An adaptive 3D document

MetaDoc matches the information presented to the information needs of the reader. The presentation styles preferred and amount of information required differ among readers. MetaDoc provides both automatic and manual control of the amount of information presented. Manual control is important as the user may wish to override the presentation decisions of MetaDoc.

MetaDoc is coded in ANSI-C under Microsoft Windows 3.0 on an IBM PS/2 model 80 with a VGA display. No existing hypertext environment was used. It is fully operational and is currently being used to present the contents of two chapters (with seventy paper pages) of the technical manual 'Managing the AIX operating system' for the IBM RT-PC (IBM, 1989). This manual was chosen because of the wide range of readership levels it is expected to address, ranging from inexperienced standalone system users to multi-user system managers. This ability range is typical of what MetaDoc can accommodate. MetaDoc can be applied to other documents without difficulty or recoding.

The linear version of this manual was rewritten in MetaDoc. The rewriting process involved breaking the text into nodes, determining inter-link nodes and intra-node stretchtext structure. The concepts explained in the manual needed to be understood and hierarchically classified by concept to facilitate user-modeling. Creation of this hierarchy could be done by a technical author with little background knowledge of MetaDoc.*

4. User Level and Levels of Information

The presentation styles preferred and the amount of information required are not the same for different readers. Thus, an objective of MetaDoc is to match the information needs and the information presented to the reader. To meet this objective, the readers and their information needs must be classified.

4.1. Users and Stereotypes

MetaDoc classifies readers with respect to their knowledge of Unix/AIX and general computer concepts into four possible categories, namely novices, beginners, intermediates or experts. This classification is based on UC, a classic user-modeling system for Unix users developed by Chin (Chin, 1986). We assume that readers' knowledge of Unix/AIX concepts and of general computer concepts has the most direct bearing on the comprehension of the prototype manual.

4.2. Concept Levels

Concepts in the manual are classified using the same scale as readers' knowledge. A concept is either a Unix/AIX concept, or a general computer concept. A concept

* Our discussions with technical authors show that they have a strong structural and semantic understanding of the documents they write. Consequently, hierarchy creation could be learned. However, inter-document hierarchies would be harder to create (because of multiple authors).

that is used both in Unix and in other operating systems is considered a general computer concept.

4.3. Presenting the Correct Level of Information

To present the correct level of information to the reader, MetaDoc varies the amount of explanation or detail information. MetaDoc uses internal information based on stereotypes about the difficulty of a concept and the reader's knowledge level.

A reader belonging to a classification whose level is lower than the difficulty of a given concept is assumed to be unfamiliar with this concept. Consequently, MetaDoc explains it by stretching text. Likewise, an expert reader does not need an explanation (since he would not find it informative), but would rather want more detailed information. MetaDoc would therefore display more detailed information for the expert.

5. The Metadoc Document

5.1. The Augmented Document

MetaDoc allows the reader to view different versions of a single node (with different amounts of information). The reader can manually move from one plane to another as well as have the system automatically calculate the best level suited for him/her. The text can be manipulated for more (or less) explanation or detail. More explanation includes definitions of key vocabulary, simpler and less technical versions, more examples, and supplemental background information. More detail includes lower-level concepts in the concepts hierarchy. The appendix shows two versions of the same node, one for the expert and one for the novice. In the second version, explanations are added to all technical concepts.

MetaDoc allows the reader to select which parts of the node to adjust, instead of adjusting the whole node. The reader can control the level of explanation or detailed information by clicking on stretchtext buttons. This allows selective stretching. To facilitate the comparison of readers' information needs with the information presented, concepts are associated with stretchtext buttons.

5.2. Writing Stretchtext

Ted Nelson (Nelson, 1971) mentioned that his 'slight experience trying to write stretchtext suggests that it is no harder to write, and perhaps easier, than ordinary prose', although he admitted that there is no firm evidence to back his claim.

Our experience showed that writing stretchtext, although not much more difficult, is all the more time consuming than writing ordinary prose. We rewrote an electronic copy of the manual in hypertext and rewrote the result in a three-dimensional form.

The principles that we followed in writing stretchtext are:

1. It is essential that the document read smoothly between the different levels of stretch. Additional text should conform nicely to the existing text when more stretch information is requested, and also if less is requested. Seamless stretchtext is important in order to to maintain the user's view of the document as being personalized.
2. Text cues must be retained between different levels of 'stretch' to minimize reader confusion. Loss of familiar 'landmarks' between levels of 'stretch' forces the reader to backtrack and re-read the node. A 'chunky' stretchtext has the same effect on the reader.
3. There should be common node identifiers for both novice and expert readers, to facilitate discussion by providing a common reference. Having sufficient commonalities between the different 'stretch' versions facilitates node identification among different readers.
4. The stretchtext should be ordered. For example, the reader can move from the most detailed version to the least detailed by directing the 'throttle' in one direction or vice-versa.

Currently MetaDoc documents are written using a text editor. A custom markup language similar to that of the Interleaf (Interleaf, 1990) desktop publishing program is used for formatting and link information. We plan to allow MetaDoc documents to be authored in Interleaf.

6. Stretchtext: Three-Dimensionality in Metadoc

Three-dimensionality in the form of stretchtext has different forms. In this section, we will consider the different dimensions of stretchtext, the types of stretchtext and the methods of stretchtext presentation.

6.1. Dimensions of Stretchtext

The information presented in a node can be varied in multiple ways. A simplistic approach would be to vary the number of words in a node, but there is little correspondence between the amount of words and the amount of information. The readability of the text (similar to the Flesch reading levels) and the presentation style (technical reference type versus procedural style) are other factors that can be varied.

In MetaDoc, we chose to vary the information presented in terms of either explanation or amount of detail. Better explanations for a concept are provided in MetaDoc by presenting a definition of key vocabulary; presenting a simpler, less technical version; and presenting supplemental background information, which is a prerequisite for understanding. More detail provides lower-level information in the hierarchy of concepts.

6.2. Types of Stretchtext

There are four possible types of stretchtext, based on the placement of the new text relative to the original:

- prefix – the additional text appears at the beginning of the original text.
- embedded – the additional text becomes embedded inside the old text.
- appended – the additional text is appended at the end.
- replacement – the new text completely replaces the original text.

We favor the embedded and appended stretchtext because they are less confusing to the reader. These two types of stretchtext preserve the position of text cues between 'stretch' operations.

6.3. Presenting Stretchtext

The degree of stretch is selected through mouse operations on expansion and detail buttons. It is important to give the user an idea of a button's operation (i.e., stretch, glossary or jump). This becomes more important in a system that employs user modeling to minimize the noise going to the user modeling component. We decided to use a context-sensitive mouse cursor, similar to KMS (Akscyn *et al.*, 1987), to make the button types evident to the reader.

The granularity of stretchtext is an important issue. MetaDoc can handle any granularity of stretchtext from fine (word level) to coarse (paragraph or section level). We favor fine granularity of stretchtext because it is less confusing to the reader and shows smoothness between the different levels of stretch.

Theoretically, the number of stretchtext levels is unlimited, although this is not practical. In MetaDoc we allow recursive stretchtext buttons; an embedded stretchtext button may 'contain' an append button, which 'contains' embed and append buttons, etc. A stretchtext button may 'contain' a jump link.

7. The Adaptive Document

MetaDoc is an active participant in the reading process. MetaDoc keeps track of the level of user knowledge and dynamically adjusts the amount of information to the user level.

The default rules used to determine the depth of information presented to the user are:

- Explanation of concepts associated with higher levels are automatically provided for lower level users.
- Explanation of concepts associated with lower levels are unnecessary for higher level users and are suppressed.
- Higher level details not necessary for understanding a certain concept are suppressed for lower level users.
- Details of equal or lower level concepts are automatically displayed for higher level users.

8. Architecture of Metadoc

8.1. OVERVIEW

The main components of MetaDoc are the 3D Document and the Intelligent Agent. The 3D Document component has the most interaction with the user. It determines the final form of the node presented to the user and receives commands from the user. The Intelligent Agent component dynamically keeps track of the user knowledge level. The Domain Concepts component bridges the gap between the Intelligent Agent component and the 3D Document component. Figure 3 shows the architecture of MetaDoc in detail.

8.2. 3D DOCUMENT COMPONENT

The 3D Document component provides the interface to the user. It is composed of the Document Presentation Manager and the Base Document.

At any time, the Document Presentation Manager presents a portion of the Base Document (node) and determines its form for display to the user. The user, after reading the node, issues one of the document manipulation commands, and thereby 'jumps' to a regular or glossary node, or to a stretchtext operation. For stretchtext operations and 'jumps' to the glossary, the Document Presentation Manager notifies the Intelligent Agent component to update the User Model. 'Jump' to a regular node is not transmitted to the Intelligent Agent since a regular node includes both explanation and detail information.

The Base Document includes regular nodes and glossary nodes containing the definition of key items. Whenever the user performs document manipulation, the Base Document is updated to reflect the status of the manipulation. Domain concepts are associated with buttons. The Domain Concepts Component keeps track of user manipulation of concepts. The Intelligent Agent component of MetaDoc functions as an assistant to the user in determining the correct level of information provided. The Document Presentation Manager refers to the Intelligent Agent component only when it cannot determine any previous user action on a particular piece of text. The user's actions are more important than the agent's, and cannot be overridden. For example, if the user clicked on a button for more detail, detail information associated with the button will always be shown until the user performs another action on the button. Other buttons associated with the same concept will exhibit the same behavior, provided that they have not been manipulated by the user. In determining the level of information presented, the Document Presentation Manager checks the following in order: button status, concept status, and the Intelligent Agent component.

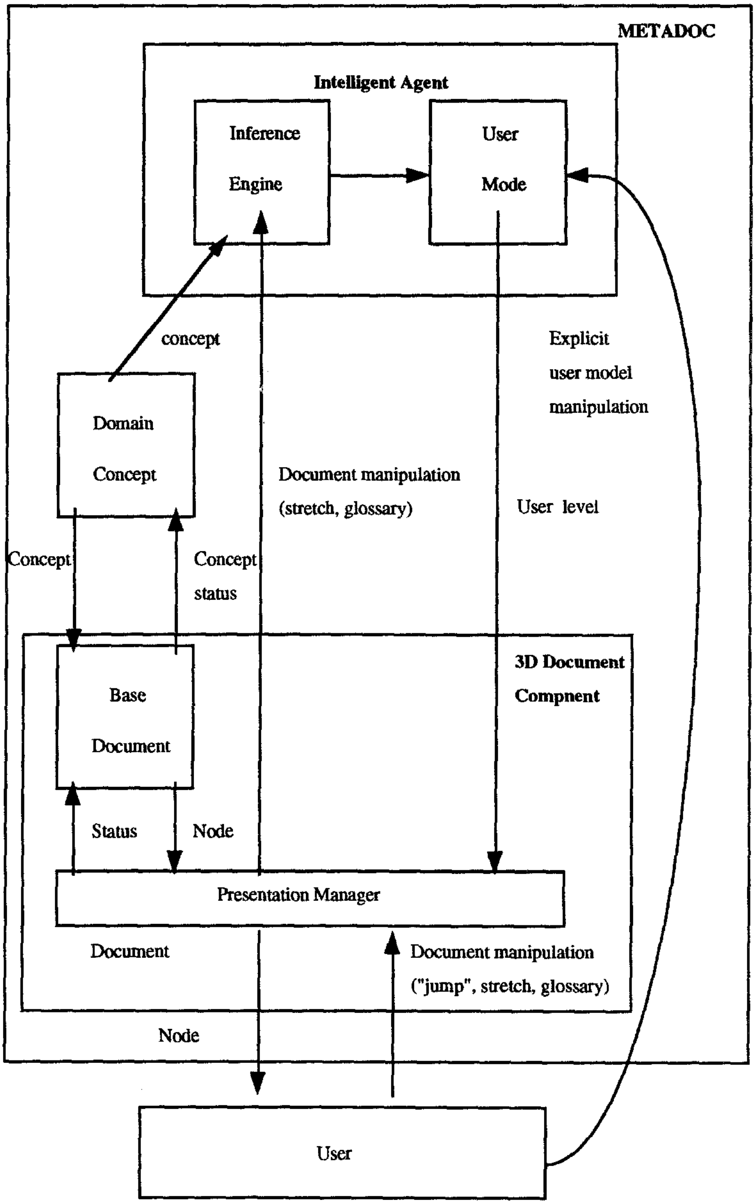

Fig. 3. Functional description of MetaDoc

8.3. Intelligent Agent Component

The Intelligent Agent component plays the central role in automatically matching the presented information depth to the user level. It unobtrusively keeps track of the user actions during the session to determine the correct user level.

The Intelligent Agent component uses a user model, consisting of both long-term and short-term elements, to represent the user level. The long-term model, which keeps track of the knowledge state of the user between sessions, is based upon an initial enrollment and the short-term model. The short-term model maintains the immediate user actions within a session.

8.4. Explicit Modeling

Compared to a natural language dialog (a common domain for user-modeling systems), the simple and efficient hypertext interface allows a low bandwidth of information for user modeling. To overcome this limitation, explicit user modeling is used. Users are requested to indicate their experience with computers before their first session. However, to skip the enrollment, users can choose the default user model.

Based on the enrollment, a model of the user is constructed. Two expertise levels, which correspond to the user's presumed knowledge of AIX/Unix and general computer concepts, are maintained for each user. In this effect, dual stereotypes are maintained for each user.

Similarly, AIX/Unix and general computer concepts in the MetaDoc document were classified into different levels. Concept levels and 'concept islands' were artificially created to organize the actual AIX/Unix and general computer concepts. Stereotypically, the actual concepts were grouped into 'concept islands' which form concept levels. Thus, AIX/Unix and general computer concepts are singly stereotyped. Special types of concepts and concept-islands are triggers (similar to Rich (Rich, 1983)) and required (Finin, 1988).

Lack of familiarity with a 'required' concept island forces a reclassification to a lower level, while knowledge of a 'trigger' concept island belonging to a higher level initiates a promotion after a certain threshold is exceeded. The same mechanism holds true at the concept level; ignorance of a 'trigger' concept indicates ignorance of the whole concept island. Concepts and concepts islands may be 'triggers' and 'required' for zero or more concept islands or levels, respectively.

The user is given the option of explicitly changing the user model within the session by specifying which concepts should be explained and which should be shown with more detail. The ability to explicitly change the user model allows users to feel in control of the session (Korfhage, 1985).

8.5. Implicit Modeling

Unobtrusive, implicit user modeling is used throughout the session to refine the user model. A request for more explanation about a concept indicates lack of familiarity with the concept. Requests for more detail imply an understanding of the concept. The stretchtext command for less explanation implies an understanding of a concept; a stretchtext command for less detail implies unfamiliarity with the concept.

Aside from stretchtext operation, 'jumps' to the glossary for definition purposes has the same effect as performing a stretchtext operation for more explanation. 'Jumps' to ordinary nodes do not have an effect on the user model, since these nodes contain both explanation and detail information.

On account of the narrow bandwidth of information for user modeling in this domain, a certain threshold must be set to balance the effect of noise against correct information. Therefore, requesting once for more explanation is not yet considered to be a lack of familiarity with a concept; however, a second request pertaining to the same concept would be.

9. Evaluation of Metadoc

9.1. Hypotheses

Guthrie (Guthrie and Kirsch, 1987) showed that there are two principle reading tasks: comprehension and location of specific information. MetaDoc was evaluated with respect to both of these issues.

The experiment primarily compared user performance between hypertext and MetaDoc. Tests on stretchtext were also conducted, since quantitative experiments on stretchtext are not available. The primary experimental hypothesis predicted that MetaDoc users would have better reading performance than stretchtext and hypertext users; i.e., MetaDoc users would have more correct answers.

Additionally, it was predicted that adaptive documents were more efficient to use. That is, MetaDoc users would spend less time, visit less number of nodes and perform fewer operations in answering the reading comprehension questions.

If, as Reinking (Reinking and Schreiner, 1985) suggested, textual manipulations – especially computer-mediated – increase reading comprehension, then subjects using MetaDoc should perform better than stretchtext and hypertext users in reading comprehension tasks. In other words, three-dimensional documents facilitate reading comprehension. Otherwise, there would be no significant difference in the performance of subjects if adaptive documents had no effect.

The secondary experimental hypothesis predicted that MetaDoc users would perform better than stretchtext and hypertext users in search and navigation tasks.

9.2. Design

The design was 2 × 3 factorial, between-subjects. The independent variables were the readers' expertise level and the system or medium used in presenting information.

The systems compared were the hypertext-only version, the stretchtext version and MetaDoc. MetaDoc was used as the base system. The hypertext-only version used the same system as MetaDoc, but the user modeling and stretchtext features were disabled. The hypertext document was taken from the original manual and re-written in hypertext by fully expanding the detail and explanation elements of each node. The stretchtext version was also based on MetaDoc and utilized the same document, but the user modeling was disabled. Users were still allowed to ask for more or less detail and move between nodes as with regular hypertext.

The dependent variables were the time (in seconds) spent in finding the correct answer, the number of correct answers, the number of nodes visited, and the number of operations (or user commands) used. The time spent in finding the correct answer also included the time spent answering the question. The answers given by the subjects were either right or wrong. The number of nodes visited includes revisits to the same nodes, backtracking and visits to the glossary nodes. This includes the number of stretchtext commands, search commands, and nodes visited. Stretchtext commands were composed of expand and compress text operations and UNDOs of previous stretchtext commands. The number of search operations included new search and continuance of the same search.

9.3. Subjects

The subjects in this experiment were students from the Department of Computer Science. The experts in this experiment belong to the technical support group for the departmental computers. No incentives were offered except for the opportunity to learn more about Unix and AIX.

9.4. Apparatus and Materials

The experiments were carried out on an IBM PS/2 model-80 type 8580–311 computer, equipped with ten megabytes of main memory, a 14-inch IBM PS/2 color display type 8514 and an IBM mouse. MetaDoc requires a VGA monitor, a two-button mouse and Windows 3.00 running in 386-enhanced mode.

Materials used were three booklets containing the experimental questions.

9.5. Procedure

Subjects were randomly assigned to one of the three systems. The distribution was balanced between all combinations of system and expertise level. Subjects were introduced to MetaDoc, then trained on their system to an objective competence standard, using a test document and verbal instruction. After the subject had browsed through the actual document for two minutes, the booklet of experimental questions was given to the subject. Five search and navigation questions preceded the eight reading comprehension questions. The subject was allowed three minutes to find the answer in the search and navigation questions and then five minutes for the reading comprehension question. For each question, the subject was allowed three tries in finding the correct answer. For the search and navigation questions, the

TABLE I. Summary of Results

	Hypertext		Stretchtext		MetaDoc	
	Expert	Novice	Expert	Novice	Expert	Novice
Reading comp. time	1780	1930	1250	1780	810	1420
Mean search time	755	725	645	530	555	575
Read. comp. correct	5	3	6.5	7	7	7
Search correct	3.5	2.5	3.5	3	3.5	3.5

Note: Times are in seconds; maximum number of correct answers is 8.

subject simply pointed out the location of the answer. The subject orally provided the answer in a few phrases or sentences for the reading comprehension questions. If the answer was vague, the subject was required to explain. The correct answer was provided if the subject failed after three attempts.

9.6. RESULTS

Two-way between-subjects Analysis of Variance (ANOVA) was the primary statistical test used. Table 1 summarizes the results.

9.6.1. Time

The main effects of both the system used and the expertise level on reading comprehension time were very significant at the 1 percent level (see line 1 in Table I). Applying Tukey's Studentized Range test to the data indicated that hypertext and stretchtext users had significantly higher mean times than MetaDoc users; however, no significant difference was found between stretchtext and hypertext users. The mean times of the novice and expert subjects also differed at the 1 percent level of significance. Interaction effects were not significant.

For search and navigation questions (see line 2 in Table I), the main effects of the system used were very significant at the 1 percent level; i.e., the mean times of the hypertext, stretchtext and MetaDoc users were different at the 1 percent level of significance. Tukey's Studentized Range test indicated that hypertext users had significantly higher mean search times than the stretchtext and MetaDoc users, although no significant difference was found between stretchtext and MetaDoc users. No significant main effect was found at the expertise level. An interaction effect between system and expertise was not significant.

9.6.2. Number of correct answers

For reading comprehension (see line 3 in Table I), the main effects of the system used on the number of correct answers (hits) was very significant at the 1 percent level. Tukey's Studentized Range test indicated that hypertext users correctly answered significantly fewer reading comprehension questions than the stretchtext and MetaDoc users, although no significant difference was found between stretchtext and MetaDoc users. MetaDoc users answered more questions correctly than stretchtext users. No significant main effect was found at the expertise level. An interaction effect between the system and expertise was not significant.

For search and navigation questions (see line 4 in Table I), the main effects of both the system used and the expertise level were not statistically significant. MetaDoc users answered more questions correctly than the users of both systems, and stretchtext version users had more correct answers than the hypertext version users. Expert users answered more questions correctly than novice users.

9.6.3. Number of nodes

In both search and reading comprehension questions, there was no significant difference in the number of nodes visited which can be attributed to the system used or to the expertise level.

9.6.4. Number of operations

In both search and reading comprehension questions, there were no significant differences in the number of operations which can be attributed to the system used or to the expertise level.

9.7. Comparison of MetaDoc and the Stretchtext Version

To test whether MetaDoc users – who answered more questions – had to perform more stretchtext operations than the stretchtext users, the number of stretchtext operations performed by users in both systems was compared. Stretchtext operations included both text expansion and compression operations.

The main effects of the system used on the number of stretchtext operations was significant at the 5 percent level for the reading comprehension questions. The difference between the means in the number of stretchtext operations performed by novice and expert users was not significant. Expert users performed more stretchtext operations.

For search and navigation questions, the main effects of both system and expertise were not significant. MetaDoc users performed slightly fewer stretchtext operations than the stretchtext version users. Expert users performed more stretchtext operations than novices.

9.8. Discussion of Results

The results of this experiment, in terms of the four dependent measures to indicate reader performance, are consistent with the primary and secondary hypotheses.

The results from the reading comprehension tasks are consistent with the primary hypothesis: users of adaptive documents spent less time answering the reading comprehension questions than stretchtext and hypertext users and also had significantly more correct answers. An appropriate explanation for this finding could be found in what Reinking (Reinking and Schreiner, 1985) suggested in the first place – that computer-mediated text manipulations enhance reading comprehension.

The timing results from the search and navigation questions are consistent with the secondary hypothesis – users of adaptive documents spent less time (although not statistically significant) in answering search and navigation questions. Interestingly, novices were consistently *faster* than experts in search and navigation, perhaps indicating a fixation on keyword searching. This contrasts with the superior performance of experts in reading comprehension.

Both primary and secondary hypotheses are not confirmed by the results of the number of nodes visited. Hypertext subjects used the string-search function more often than MetaDoc and stretchtext subjects. Through the string-search function, hypertext users visited a large number of nodes but spent more time reading the nodes than MetaDoc and stretchtext users. MetaDoc subjects used a browsing strategy most often to find information within the document. Although browsing requires an understanding of the contents of the document, MetaDoc users were able to visit a large number of nodes since they spent less time reading the nodes. Perhaps on account of the readability of the MetaDoc document, the MetaDoc subjects preferred browsing to the string-search function.

MetaDoc had greater impact on novice users than experts. Results were more significant in reading comprehension than in search and navigation. This conclusion is consistent with previous reading research, showing reading aids to be more significant to novice than expert readers (Reinking and Schreiner, 1985).

10. Metadoc in Perspective

Many tools have been devised to aid hypertext users. These tools fall into two categories: those which aid navigation and orientation and those which aid reading. Navigation and orientation tools include graphical browsers as implemented in Intermedia (Yankelovich *et al.*, 1985; Yankelovich *et al.*, 1989) and NoteCards (Halasz *et al.*, 1987), as well as in bookmarks (Bernstein, 1988), hierarchical adaptive indexing (Frisse, 1987; Frisse *et al.*, 1989) and fisheye views (Furnas, 1986).

MetaDoc has no intended relation to any of these navigation and orientation tools. However, we believe that the philosophy behind the adaptive three-dimensional writing style will reduce the possibility of disorientation and navigation problems. By presenting information at the appropriate detail level, users will have less need to browse and consequently are less likely to become lost. Thus, MetaDoc prevents rather than cures one source of ‘lost in space’ problems.

Reading tools are often trail and path oriented, and typically suggest a path through a hypertext which will lead the user to pertinent information in an ordered sequence. Particular paths may be oriented towards different user-ability levels and may present a number of different options at different nodes. Examples of this approach include Zellwegger's active paths (Zellwegger, 1989) .

Some of MetaDoc's personalization capabilities can be simulated through the use of path-based mechanisms. MetaDoc is more sophisticated, however, in that it:

- represents and dynamically alters a model of each user;
- includes a concept-based understanding of the document;
- transparently (to the user) modifies the level of presentation in terms of detail and expansion;
- manipulates text rather than links.

Perhaps the greatest difference is that MetaDoc modifies the entire document to suit the readers ability. It has no explicit notion of guiding the user through the document by forcing a reading order, though. MetaDoc documents could be read either linearly or non-linearly, depending on the readers requirements.

11. Conclusion

MetaDoc is a system rather than a tool. Adaptive documentation is the core of MetaDoc rather than a usability-related tool. MetaDoc seeks to fundamentally improve the way information is presented, rather than to cure known problems.

MetaDoc provides an environment in which the user can read a hypertext document that will adapt to his/her needs. MetaDoc does not take the entire control away from the user. The user can adapt the degree of detail or explanation as needed. MetaDoc can help improve reader performance by enhancing the comprehensibility of the document.

References

Akscyn, R., D. McCracken and E. Yoder: 1987, 'KMS: A Distributed Hypermedia System for Managing Knowledge In Organizations', *Proceedings of Hypertext'87.* Chapel Hill, North Carolina.

Bernstein, M: 1988, 'The Bookmark and the Compass: Orientation Tools for Hypertext Users', *ACM SIGOIS Bulletin* **9** (4), 34–45.

Brown, P. J.: 1987, 'Turning Ideas into Products: The Guide System', *Proceedings of Hypertext'87.* Chapel Hill, North Carolina.

Chin, D.: 1986, 'User Modelling in UC: the Unix Consultant', *Proceedings of the CHI-86 Conference.* Boston.

Dynatext Corp: 1990, One Richmond Square. Providence, Rhode Island.

Egan, D, J. R. Remde, T. K. Landauer, C. C. Lochbaum and L. M. Gomez: 1989, 'Behavioral Analysis of a Hypertext Browser', *Proceedings of CHI 89.* Addison Wesley.

Finin, T.: 1988, 'Default Reasoning and Stereotypes', *International Journal of Expert Systems* **1** (2), 131–158.

Frisse, M. E.: 1987, 'Searching for Information on a Hypertext Medical Handbook', *Proceedings of Hypertext'87.* ACM Press, Baltimore, MD, pp. 57–66.

Frisse, M.F., S. B. Cousins: 1989, 'Query by Browsing: An Alternate Hypertext Information Retrieval Method', *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care.* IEEE Computer Society Press.

Furnas, G. W.: 1986, 'Generalized Fisheye Views', *Proceedings of the 1986 ACM Conference of Human Factors in Computing Systems.* pp. 16–23.

Guthrie, J. T. and I. S. Kirsch.: 1987, 'Distinctions Between Reading Comprehension and Locating Information in Text', *Journal of Educational Psychology* **79**, 220–227.

Halasz, F. T. Moran, and R. Trigg: 1987, 'NoteCards in a nutshell', *Proceedings of the CHI '87 Conference.* Toronto, Canada, pp. 45–52.

IBMa: IBM Advanced Workstations Division, Austin, Texas.

IBMb: IBM Entry Systems Division, Austin, Texas.

Interleaf Inc.: 1990, Waltham, MA.

Kobsa, A. and W. Wahlster: 1989, *User Models in Dialog Systems.* Berlin, Springer-Verlag.

Korfhage, R. R.: 1985, 'Intelligent Information Retrieval: Issues in User Modelling', *Conference paper in Expert Systems in Government Symposium.* IEEE Computer Society Press, Washington, DC.

Nelson, T.: 1971, 'Computopia and Cybercrud', in: Levien (ed.): *Computers in Instruction.* The Rand Corporation.

Nielsen, J.: 1990, *Hypertext and Hypermedia.* Academic Press, NY.

Nielsen, J. and U. Lyngbaek: 1989, 'Two Field Studies of Hypermedia Usability', *Proceedings Hypertext2 Conference.* York, United Kingdom, pp. 29–30.

Reinking, D. and R. Schreiner: 1985, 'The Effects of Computer-mediated Text on Measures of Reading Comprehension and Reading Behavior', *Reading Research Quarterly,* Fall 1985, 536–552.

Rich, E.: 1983, 'Users Are Individuals: Individualizing User Models', *International Journal of Man-Machine Studies* **18**, 199–214.

Schnase, J., J. Leggett, C. Kacmar and C. Boyle: 1988, 'A Comparison of Hypertext Systems', *TAMU Technical Report 88–017.* Dept. of Computer Science, Texas A&M University, College Station, TX.

Texas Instruments (1988), Austin, TX

Yankelovich, N., N. Meyrowitz and A. van Dam: 1985, 'Reading and Writing the Electronic Book', *IEEE Computer,* Oct. 1985, 15–30.

Yankelovich, N., B. J. Haan, N. Meyrowitz, and S. M. Drucker: 1988, 'Intermedia: The Concept and the Construction of a Seamless Information Environment', *IEEE Computer,* Jan. 1988, 81–96.

Zellweger, P.: 1989, 'Scripted Documents: A Hypermedia Path Mechanism', *Proceedings of Hypertext'89.* Seattle, Washington.

Appendix

MetaDoc

General System Structure

The AIX Operating System has three parts:

* The AIX Virtual Resource Manager (VRM)

* The AIX Operating System kernel

* The shell

User Model:
Unix/AIX:
expert
Gen. Computer:
expert

An expert's view of a MetaDoc node.

MetaDoc

General System Structure

The AIX Operating System (a group of programs that act as interface between the user and computer) has three parts:

* The AIX Virtual Resource Manager (VRM), a set of programs that manages the resources of the computer (main storage, disk storage, display stations, and printers).

* The AIX Operating System kernel, a set of programs that send instructions to the VRM. It is a set of programs tha control, using the VRM, the system hardware (the physical components of the system).

* A shell is often called an interface or a command interpreter. It is the part of the operating system that allows access to the kernel.

User Model:
Unix/AIX:
novice
Gen. Computer:
novice

A novice's view of the same MetaDoc node