AN EXPERT SYSTEM APPROACH TO THE EVALUATION OF
HYPERTEXT ENGINEERING: AN EXPERIMENT
WITH KNOWLEDGEPRO AND MAXTHINK'S

DISSERTATION

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Min Sun Pak, B.S., M.B.A.B.C.I.S.
Denton, Texas
December, 1991
AN EXPERT SYSTEM APPROACH TO THE EVALUATION OF
HYPERTEXT ENGINEERING: AN EXPERIMENT
WITH KNOWLEDGEPRO AND MAXTHINK'S

DISSERTATION

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Min Sun Pak, B.S., M.B.A.B.C.I.S.
Denton, Texas
December, 1991
ACKNOWLEDGEMENTS

First of all, Dr. Ana D. Cleveland, my major professor, is gratefully acknowledged for her helpful guidance and continual encouragement throughout the whole process of this research from the proposal to the conclusion. Without her genuine contributions, this research could not be successfully completed.

Dr. Donald B. Cleveland is also thanked for his constructive comments and criticisms during the formulation stage of the prototype expert system, Hypertext Evaluation Expert System (HEES). His efforts led the HEES to be a greatly enhanced prototype. Dr. Dewey E. Carroll deserves substantial credit for the research design. He expended valuable time and effort in validating the appropriateness of the research design for this study.

I would like to give special recognition to Dr. Norman Howden. He led directly to this dissertation topic by providing insightful suggestion and played a great role in developing the HEES as a human expert in hypertext. I would like to thank Dr. Tae Guk Kim for his advice. For their painstaking supports, I should express my deep gratitude to all my family: Kyung Hee, Sang Chull, and Sung Min.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>vii</td>
</tr>
<tr>
<td>SOFTWARE</td>
<td>viii</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Background of the Problem</td>
<td></td>
</tr>
<tr>
<td>Statement of the Problem</td>
<td></td>
</tr>
<tr>
<td>Purpose of the Study</td>
<td></td>
</tr>
<tr>
<td>Significance of the Study</td>
<td></td>
</tr>
<tr>
<td>Limitation of the Study</td>
<td></td>
</tr>
<tr>
<td>Assumptions</td>
<td></td>
</tr>
<tr>
<td>II. REVIEW OF THE LITERATURE</td>
<td>6</td>
</tr>
<tr>
<td>Expert Systems Shells and Building Expert Systems</td>
<td></td>
</tr>
<tr>
<td>Expert System Approaches to Evaluation and Assessment</td>
<td></td>
</tr>
<tr>
<td>Domain Knowledge (Hypertext Software)</td>
<td></td>
</tr>
<tr>
<td>III. HYPOTHESES</td>
<td>11</td>
</tr>
<tr>
<td>Hypothesis 1</td>
<td></td>
</tr>
<tr>
<td>Hypothesis 2</td>
<td></td>
</tr>
<tr>
<td>IV. METHODOLOGY</td>
<td>12</td>
</tr>
<tr>
<td>Experimental Design</td>
<td></td>
</tr>
<tr>
<td>Treatments</td>
<td></td>
</tr>
<tr>
<td>Subject Selection and Random Assignment</td>
<td></td>
</tr>
<tr>
<td>Instrumentation</td>
<td></td>
</tr>
<tr>
<td>V. DEVELOPING AN EXPERT SYSTEM</td>
<td>18</td>
</tr>
<tr>
<td>Selection of Application Domain</td>
<td></td>
</tr>
<tr>
<td>Selection of Resources</td>
<td></td>
</tr>
<tr>
<td>Knowledge Acquisition</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Two-Way Structure and Number of Subjects</td>
<td>13</td>
</tr>
<tr>
<td>2. Random Assignment and Treatments</td>
<td>14</td>
</tr>
<tr>
<td>3. Resources Utilized to Develop the HEES</td>
<td>25</td>
</tr>
<tr>
<td>4. Number of Participants at Each Location</td>
<td>114</td>
</tr>
<tr>
<td>5. Major Disciplines of Subjects</td>
<td>115</td>
</tr>
<tr>
<td>6. ANOVA for Testing Hypothesis 1</td>
<td>118</td>
</tr>
<tr>
<td>7. ANOVA for Testing Hypothesis 2</td>
<td>121</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

Figure | Page
---|---
1. Profile for Testing Hypothesis 1 | 117
2. Profile for Testing Hypothesis 2 | 120
SOFTWARE

Software developed for this dissertation, Hypertext Evaluation Expert System (HEES), is on file in the University of North Texas Library. A 3.5" HD/DS diskette contains the HEES. The Chapter V describes the HEES in detail.
CHAPTER I

INTRODUCTION

Background of the Problem

In the Machine Age, prompted by the Industrial Revolution, human beings tended to view everything, including themselves, in machine-like terms. Since the end of World War II, this mechanistic way of viewing nature has begun to be replaced by a less mechanistic way of viewing things.

With the emergence of the Information Age, there was a change of attitudes. Instead of viewing human beings in machine-like terms, some pioneers began to view machines in human-like terms. The assumption was made that machines could assume not only the physical, but also mental attributes of human beings, provided that such machines can track phenomena or objects and process information.

Memex (Bush 1945), which represented the idea of associative memory and modern hypertext, is an example of this nonmechanistic way of viewing:

The real heart of the matter of selection, however, goes deeper than a lag in the adoption of mechanisms .... Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. ... Having found one item, moreover, one has to emerge from the system and re-enter on a new path. ... The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by
the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain.

This way of representing the association of thoughts, ideas, or information is an area of study called associative information retrieval. Goffman's Indirect Method of information retrieval (Goffman 1968) embodied a similar concept in his alternative approach to file structure and searching. The Indirect Method was characterized as a "chaining type of file structure and search strategy" (Cleveland and Cleveland 1983) or "internally linked method" (Cleveland and Cleveland 1984). However, it is quite different from the Memex's approach.

A prime example of the Memex's approach is the fast developing software field known as hypertext. The term "hypertext" was coined by Ted Nelson, and it refers to a non-linearly and non-sequentially stored text.

Hypertext software engineering deals with the design and development of textual databases utilizing a series of links and nodes to associate units of information for referential or presentational purposes. For example, the NLS system (Gluck 1989, 172), developed by Doug Engelbart in 1968, allows users to manipulate symbols for concepts by rapidly displaying, storing, recalling, and operating.

Artificial intelligence (AI), another example of the nonmechanistic way of viewing things, can be defined as the science of making machines behave in a way that would
generally be accepted as requiring human intelligence (Prerau 1990, 2). Expert systems, one of the subfields of AI, are software that uses expertise and knowledge acquired from human experts to solve problems.

**Statement of the Problem**

Software evaluation usually requires sophisticated tools and aids, especially for leading-edge software like hypertext. An expert system approach may provide a tool for the evaluation of such advanced software.

In this study, the question asked is:

Can a prototype expert system, that acts in the evaluation of hypertext software, be developed? In other words, can users evaluate hypertext software with the aid of an expert system?

The object software is a prototype expert system called Hypertext Software Expert System (HEES), which was created and tested by the investigator of this study.

**Purpose of the Study**

The purpose of this study is to create the prototype expert system, HEES, and to examine its usability and usefulness in evaluating hypertext software.

**Significance of the Study**

The essence of hypertext engineering lies in the concept of associative chaining of information or documents. Recently, such associative information management approaches...
have been enhanced in hypertext engineering by integrating the software developments in various related fields, such as information storage and retrieval systems, database management systems, and expert systems.

However, users are inundated by a number of hypertext software packages that have differences in their functions, applications, terminologies, and general approaches. Evaluating such sophisticated hypertext software without any type of aid is analogous to judging direction without a compass or a map in an African jungle. Thus, due to their inherent capabilities, expert systems could be useful aids in evaluating hypertext software.

**Limitation of the Study**

Subjects were not chosen randomly but on a voluntary basis. Such non-randomness may fail to ensure the accurate representation of the population or the control of irrelevant variables. Thus, the extrapolation and generalization to whole population from the possible findings of this study may be limited.
**Assumptions**

1. It is assumed that there are no extreme differences in the cognitive ability of subjects over all the treatment groups and the control groups.

2. It is assumed that the behavior of the investigator of this study does not differ in each of the treatment groups or the control groups, except in those areas that distinguish between treatments.

3. It is assumed that the evaluation criteria for "hypertext engineering" differ from those for "hypertext software packages": the evaluation criteria for "hypertext engineering" may emphasize the function for building of associative chaining database while the evaluation criteria for "hypertext software packages" may emphasize usage itself or may include cost factors.
CHAPTER II

REVIEW OF THE LITERATURE

Expert System Shells and Building Expert Systems

Expert system development tools can be divided into two major categories: programming languages and expert system shells. The programming languages used for expert system applications (Waterman 1985, 80) are generally either problem-oriented languages, such as FORTRAN and PASCAL, or symbol-manipulation languages, such as LISP or PROLOG.

The expert system shells, which provide built-in facilities for building expert systems, include inference engine or full-featured knowledge engineering tools. Waldrop (1987, 158) noted:

Tools of a more recent vintagte are the expert-system "shells," which quickly made their first commercial appearance in 1984, and which quickly became one of the hottest segments of the AI market.

With the rapidly growing processing power and memory capacity, low-cost personal computers have contributed to the wider usage of the expert system shell in building expert systems. In the rest of this section, the literature about the expert system shells are reviewed.

EMYCIN (van Melle 1984) became an expert system shell from MYCIN which was invented for diagnosis of bacterial infections and prescriptions of treatments. The expert
system shell, EMYCIN, has been used to construct a number of prototype expert systems. Examples include: AUDITPLANNER for materiality judgements (Steinbart 1987); Knowledge-Based System for Individual Income and Transfer Tax Planning (Michaelsen 1982); and Expert System for Federal Tax Planning (Michaelsen 1984).

Similarly, KAS (Waterman 1985, 96) became an expert system shell from PROSPECTOR, which was developed to help geologists locate ore deposits and perform log analysis related to drilling operations, by removing all knowledge of geology and leaving only the inference engine and rule syntax.

Recently, full-featured general purpose shells have been developed for personal computers (PC). Examples include EXSYS, KEE, M.I, and KnowledgeMaker. EXSYS (EXSYS 1986), developed by EXSYS Inc., is capable of handling 5000 rules in a PC with 640K. CLAES (Pak and Nam 1989) is a prototype of commercial loan analysis expert system for business loan officer with rule-based application on an expert system shell, EXSYS.

KEE, which stands for Knowledge Engineering Environment, was developed by Intellcorp. The Lawrence Livermore Laboratory used the KEE to devise an expert system for fine-tuning its masspectrometer to give optimum performance (Waldrop 1987, 158-159). KnowledgeMaker was
invented by Knowledge Garden and M.I was developed by Texas Instruments’ Personal Consultant and Teknowlege.

Expert System Approaches to Evaluation and Assessment

At the eighth Expert Systems Conference organized by the British Computer Society Specialist Group on Expert Systems in 1988, Leung, Chandarana, and Mannock (1989) presented a prototype expert system to evaluate and improve the performance of a database system. The system allows a user to invoke and utilize its stored knowledge in a simple manner. With a justification interface, it makes visible to the user the chain of reasoning followed by the system. They concluded that a pure Prolog implementation, in the software evaluation expert system, is inadequate for efficiency reasons.

To evaluate the program partitioning and allocation problem, Aref (1988) developed an expert system by gathering known heuristics and other heuristics related to partitioning and allocation of algorithms in different application areas. As a knowledge base for the expert system, evaluation criteria were created to judge the performance of the heuristic approach.

A research (Gaudel 1987) on the automation of software test strategies found that the extended logic programming languages like SLOG and RAP can provide the resolution mechanism of PROLOG. SLOG was developed at Laboratories de Marcoussis and RAP was developed at the University of
Passau. Both of them were enriched by conditional rewriting, and dealt with functions and equality.

PROJCON (Underwood and Summerville 1981) is an expert system, developed using the EMYCIN, to help a project manager evaluate software development projects. It diagnoses the project's problems and their causes and explains its reasoning processes.

Domain Knowledge (Hypertext Software)

In rapidly developing domains or disciplines, it is usually difficult to find a generally accepted set of topology. Hypertext is not an exception. However, currently available or feasible hypertext systems tend to fall into one of the following categories (Gluck 1989, 5-6): large-literary systems, issue investigation systems, browsing systems, modified database management systems (DBMS), and general experimental systems.

HyperCard (Anzobin 1988), released by Apple Computer, Inc., is a mouse-oriented index card database being composed of stacks, backgrounds, cards, graphics, fields, buttons, the menu bar, menus, the message box, and scripts. HyperCard is primarily an interface product, providing the capability to interface the user to other software, to programs written in its macro language, or to external resources via modem (Howden 1990).

Project Xanadu (Nelson 1988), upgraded from Xanadu which has been revised several times by Ted Nelson since
1960, is designed to be the principal publishing utility of the future. It is to provide for deposit, delivery, and continual revision of linked electronic documents.

KnowledgePro (Franklin 1989), developed by Knowledge Garden, combines two cutting-edge technologies, hypertext and the expert system. Running on IBM PCs, KnowledgePro can link to texts, graphics, and many other expert system tools.

MaxThink's interrelated hypertext development tools (Devlin and Berk 1989), hereinafter referred to MaxThink's, include Houdini, Hyperez, and other products. Hyperez is a memory-resident system for authoring small hypertext applications containing fewer than 100 links. Houdini is the principal tool for building complex hypertext networks.
CHAPTER III

HYPOTHESES

One of the hypotheses in this study stated that an expert system approach to hypertext evaluation may result in a different kind of evaluation results from an evaluation without the expert system approach. The other hypothesis stated that the evaluations with the expert system approach produces consistent results within a statistical significance level.

Hypothesis 1

There is a significant difference between the hypertext software evaluation results by the control groups who don’t use the HEES and the results by the experimental groups who use the HEES.

Hypothesis 2

$E_1 = E_2 = E_3 = ... = E_n$ where $E_n$, dependent variable for this experiment, refers to the hypertext software evaluation result by subject n using the expert system, HEES. In other words, the HEES works consistently as a tool for the evaluation of hypertext software.
CHAPTER IV

METHODOLOGY

Experimental Design

The Multiple-Group Posttest Design (Spector 1981, 44-45) was employed as an embryo for the experimental design of this study. In terms of post-test only design, this design is similar to the Post-Test Only Control Group Design (Campbell and Stanley 1963, 25-26) and to the Posttest-Only Design with Nonequivalent Groups (Cook and Campbell 1979, 96-98). The initial structure of this design is: \( \{X_1, O_1\} \{X_2, O_1\} \{X_3, O_1\} \ldots \{X_n, O_1\} \), where the X’s represent the treatment groups as the independent variable and the O’s represent observations of the dependent variable, the hypertext evaluation result by subject n.

The design is a balanced two-way structure within the context of the above initial structure. Two-way indicates that the design structure consists of treatments factor and experience factor. The subjects are divided into the three groups, depending upon the level of their experience with software (Table 1).

There are one control group and four treatment groups in each experience factor. In total, there are three
Table 1

TWO-WAY STRUCTURE AND NUMBER OF SUBJECTS

<table>
<thead>
<tr>
<th></th>
<th>Novice</th>
<th>Experienced with hypertext software</th>
<th>Experienced with other software</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>T3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Sub-total</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Total</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>48</td>
</tr>
</tbody>
</table>
Table 2
RANDOM ASSIGNMENT AND TREATMENTS

<table>
<thead>
<tr>
<th>Variations</th>
<th>Software first accessed</th>
<th>Training period</th>
<th>Random Number Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Groups</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5, 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6, 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7, 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8, 16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Treatment Groups</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>KnowledgePro</td>
<td>10 Minutes</td>
<td>1, 9</td>
</tr>
<tr>
<td>T2</td>
<td>KnowledgePro</td>
<td>30 Minutes</td>
<td>2, 10</td>
</tr>
<tr>
<td>T3</td>
<td>MaxThinks</td>
<td>10 Minutes</td>
<td>3, 11</td>
</tr>
<tr>
<td>T4</td>
<td>MaxThinks</td>
<td>30 Minutes</td>
<td>4, 12</td>
</tr>
</tbody>
</table>
control groups and twelve treatment groups. The total number of the subjects in the control groups is equal to the total number of the subjects in the treatment groups. Each cell of the treatment groups contains an equal number of observations, thus it is a balanced (orthogonal) design.

Treatments

The variations on the sequence of access to the hypertext software (e.g., first KnowledgePro or MaxThink’s) and the period of training session (ten minutes or thirty minutes session) differentiates the treatments (Table 2). For example, after training for ten minutes with a script (Appendix E), the subjects of the Treatment Group 1 (T1) first accessed KnowledgePro and then MaxThink’s to evaluate the hypertext software through the expert system, HEES.

The Treatment Group 4 (T4) was trained for thirty minutes with a script (Appendix F). Then, they first accessed MaxThink’s and later KnowledgePro. Consequently, each experience factor consists of four different treatment groups and one control group. The control groups were excluded from any treatment, including the guidance and utilization of the HEES.

Subject Selection and Random Assignment

Random selection of subjects is desirable to ensure their representativeness of the general population. However, a truly random representative human sample is
difficult to obtain. Forty-eight subjects in this study were selected from volunteers. In other words, sampling was the subjects' self-selection. Adair (1973, 49) described that self-selection is a common method of obtaining samples. A study (Black, Schumpert, and Welch 1972) noted that in a rotor performance experiment, volunteers had a greater accuracy and perseverance than did coerced subjects who were poorly motivated.

In this study, possible bias resulting from self-selection was minimized through random assignment. Following the paradigm of random assignment, which ensures controlling internal and external validity, the subjects were randomly assigned to one of the treatment groups or the control group within each experience factor. A random number generator (Appendix A), provided by a BASIC program utility, was used for the random assignment.

Based upon their experience of software usage, they were classified into three groups: novice; experienced with conventional database management software (spreadsheets, databases including online textual searching, etc.); and experienced with hypertext software. Level of experience was identified through a questionnaire (Appendix B). In order to protect the subjects' privacy, they were informed that any personal data would not be reported individually anywhere but used only for this experimental study in summarized-by-group formats.
Instrumentation

The main functions of instruments (Cleary 1977, 2-3) in an experiment are: presenting stimuli for systematic control of the independent variable; allowing the experimenter to control other potentially independent variables; measuring the dependent variable; controlling sequences in an experiment; recording data; and finally, analyzing data and presenting the results to the experimenter.

The prototype expert system, HEES, fills most of the above roles as the instrument in this experimental study. Data were collected from experimental groups through HEES. Following the guidelines of the HEES, the subjects of treatment groups responded to the questions prompted from the HEES. Then, the instrument recorded, processed the data, and generated the evaluation output using its backward chaining inference system.
The HEES was knowledge-engineered by the investigator of this study, who acquired domain knowledge and judgement expertise about hypertext engineering for the knowledge base. Using a modularity approach and IF-THEN rules, the knowledge base for the HEES was created. Modules consisted of a number of distinct but interrelated factors, and were integrated into one knowledge base.

The knowledge engineering process for the HEES consists of five highly interdependent and overlapping phases: selection of an application domain; selection of resources; knowledge acquisition; knowledge representation; implementation; and validation and verification. However, the whole process of the development is a cycle of reformulations, redesigns, and refinements.

The Selection of Application Domain

The phase for selection of application domain, includes the identification of the followings: problem to be solved
through an expert system; and feasibility within time, cost and technical constraints. The application domain for this study is the evaluation of hypertext engineering.

Hypertext engineering, as an application of software engineering in hypertext software, deals with the design and development of non-linear textual databases utilizing a series of nodes and links. The evaluation criteria for "hypertext engineering" may differ from those for "hypertext software packages." The evaluation criteria for "hypertext engineering" may emphasize the function for building of associative chaining database while the evaluation criteria for "hypertext software packages" may emphasize usage itself or may include cost factors. From a user's viewpoint, cost factors in evaluating software are very fluctuating because the price of software changes heavily in a short-term period. Even though cost is important, this study was concerned with the technical aspects and not with cost-benefit. A cost-benefit analysis in decision-making for purchase or software development would be a different level of evaluation.

The Selection of Resources

This phase includes the selection of the followings: software; human resources (knowledge engineers and domain experts); and hardware.

1. Expert System Development Software (Shell)

An expert system shell, KnowledgeMaker developed by
Knowledge Garden, was employed in a pilot study for this research. In addition to the backward chaining inference engine, the KnowledgeMaker provides excellent induction system which reads data in text or spreadsheet and finds the most relevant data to the value of the goal attribute. However, it was found that the shell is inappropriate to handle attributes with explanation in more than one line.

Thus, it was replaced by the other shell, EXSYS Professional developed by the EXSYS Inc. The EXSYS Professional can handle more narrative qualifiers (attributes). The EXSYS Professional, however, does not provide an induction system.

2. Utilizing a Single Expert in Building Knowledge Base

Prerau emphasized the advantages of utilizing a single primary expert in building knowledge base:

> It is generally advantageous for the project to select a domain where it will be possible to acquire the expertise for the system ... principally from a single expert. A project using a single expert avoids the problems inherent in dealing with multiple experts. Experts in the same domain or subdomain may disagree on their conclusions ... Even experts generally agree on their conclusions usually utilize somewhat different problem-solving techniques. Hence it is difficult to develop a consistent body of knowledge on a specific area of expertise by acquiring the knowledge partially from one expert and partially from another or several others (Prerau 1990, 114).

Prerau added more comments on using a single expert:

> Clearly there is even a greater problem in attempting to combine the knowledge and reasoning of two or more experts who often disagree on conclusions. ... If one leading expert says "Yes"
and another says "No," what should the expert system say (Prerau 1990, 192-193)?

Vedder also asked for students in his class to use a single expert in developing expert systems as term projects and to explain reasons if more than one expert were used (Vedder 1989, 4). Most of expert systems developed at the class under his supervision used a single expert (e.g., Forcasting Expert System, Vocational Advisor, ARCO Sales and Use Tax Advisor, Debugger, Commercial Loan Analysis Expert System, etc.).

Another examples of expert systems used a single expert are as the followings: the COMPASS expert system for the maintenance of a large telephone switching, which was initiated in the Knowledge-Based Department of GTE Laboratories' Computer and Intelligent Systems Laboratory (Prerau 1990, 33-34); and an expert system to review Medicare Claims for the Blue Cross and Blue Shield of South Carolina (Weitzel and Andrews 1988).

3. Hardware

An IBM or compatible microcomputer with an 80386 processor was selected to develop the HEES. For the EXSYSProfessional's color display, VGA monitor is preferred.

Knowledge Acquisition

The knowledge acquisition phase begins from the process of finding domain knowledge including facts, rules, heuristics, and procedures. Sources of domain knowledge may
consist of public domain knowledge which can be obtained from the literature and human expertise. This phase can be wrapped up with a specification of the knowledge of the expert system.

Schmoldt employed the combination of literature and human experts as the sources of knowledge which is desirable to exploit for an expert system for forest pest management (Schmoldt 1987, iv).

A methodology was developed for transforming current literature knowledge and the private knowledge of human experts into decision rules. Factors relevant to pest diagnosis decisions were identified. English-like decision rules were constructed, and human experts were asked to review these rules.

Schmoldt insisted "Because the literature is very descriptive and list-oriented, with few omissions, it proved to be a good place to begin enumerating these factors ... Several experts were then presented with these lists, and their comments were solicited (Schmoldt 1987, 90)."

The Hepatitis Knowledge Base, developed as a National Library of Medicine project for providing answers to a wide variety of biomedical questions about hepatitis rather than accessing to documents, is the other example of expert system based upon the combination of printed literature and human experts (Ralph and Micco 1990, 91-92, 115-117):

The first stage in the knowledge acquisition process was to limit the number of knowledge sources to be incorporated in the knowledge base. This was done by using review articles which are useful for identifying the most important sources because: they tend to be written by human
experts who were selected because of their expertise; they exclude low-quality material; they eliminate redundancy; they are based on experts contextual knowledge and are typically organized by applying a human expert’s schema for representing the knowledge in a topic area to a large, diverse body of literature.

PLEXUS is an expert system for referral about gardening (Vickery and Brooks 1987) developed at the University of London:

PLEXUS is another example of a system designed to augment the use of the printed literature. ... refer its users to human experts and to gardening associations in addition to printed reference works (Ralph and Micco 1990, 92).

Also, Rada described "An expert system for diagnosing a patient’s condition can incorporate information from many textbooks and strive to duplicate the decision making of a large professional community (Rada 1990, 67)."

Manning (1989, 9) noted "Often the terms knowledge systems, or knowledge-based systems, are used in a broader sense to include not only an expert’s knowledge but also public knowledge about a particular domain." Prerau (1990, 3) described "Expert systems are often called knowledge-based systems, and sometimes knowledge systems."

Knowledge Representation

The knowledge representation phase consists of the selection of the knowledge representation scheme, the selection of the knowledge engineering tool, and coding. As the most widely used schemes for knowledge representation in current expert systems, Waterman (1985, 20) listed rules,

"Knowledge Representation"

The knowledge representation phase consists of the selection of the knowledge representation scheme, the selection of the knowledge engineering tool, and coding. As the most widely used schemes for knowledge representation in current expert systems, Waterman (1985, 20) listed rules,
semantic nets, and frames while Prerau (1990, 18) pointed out production rules, frames, and object-oriented programming. The HEES adopted the production rules (IF-THEN rules).
CHAPTER VI

HYPERTEXT EVALUATION EXPERT SYSTEM (HEES)

Developing a prototype expert system for hypertext evaluation, HEES, is the main body of this study. This chapter describes the evaluation criteria of hypertext software, differences in tasks and users as preliminary considerations in evaluating software, detailed evaluation criteria, qualifiers, choices, and rules of the HEES. Resources utilized to develop the HEES are shown in the Table 3.

Table 3
RESOURCES UTILIZED TO DEVELOP THE HEES

<table>
<thead>
<tr>
<th>Knowledge Engineer:</th>
<th>Min Sun Pak, the author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Expert:</td>
<td>Dr. Norman Howden</td>
</tr>
<tr>
<td>Advisory Committee:</td>
<td>Dr. Ana D. Cleveland, Chair</td>
</tr>
<tr>
<td></td>
<td>Dr. Donald B. Cleveland</td>
</tr>
<tr>
<td></td>
<td>Dr. Dewey E. Carroll</td>
</tr>
<tr>
<td>Expert System Shell:</td>
<td>EXSYS Professional</td>
</tr>
<tr>
<td>Hardware:</td>
<td>IBM or its compatible with an 80386 processor</td>
</tr>
</tbody>
</table>
Evaluation Criteria of Hypertext Software

1. Hypertextability

1.1. Dynamics of Linking

1.1.1. Number and Type of Links Supported

1.1.1.1. Text Editor Linking
1.1.1.2. Linking by Matrix Outlining
1.1.1.3. Key Word Linking
1.1.1.4. File Linking
1.1.1.5. Hierarchical Links
1.1.1.6. Referential Links
1.1.1.7. Cluster Linking

1.1.2. The Intuitiveness of Linking Procedure

1.2. User's Manipulations of Nodes

1.2.1. Capability of Editing Nodes (e.g., moving, copying, deleting, inserting)

1.2.2. Capability of Modifying and Arranging both Nodes and Links Together (e.g., dividing, merging, sorting, etc.)

1.2.3. Displaying Graphics

1.2.4. Optional Functionalities for Node Manipulation

1.2.4.1. Typed Nodes Defining a Set of Data Structure and Valid Operation

1.2.4.2. Annotation on Nodes

1.2.4.3. Multiple Users' Access for Concurrent Authoring
1.2.4.4. Versioning
1.2.4.5. Creating Graphics
1.2.4.6. Sound
1.2.4.7. Animation

1.3. Traversing
   1.3.1. Backward Traversing
   1.3.2. Browsing Map
   1.3.3. Multiple Projection Windowing

1.4. File Transformation from Other Applications (e.g., ASCII text, Word Perfect 5.1., Lotus 1-2-3, dBASE IV, graphics) into Hypertext

2. The Speed of Operation in Link Activation

3. Printing Hypertext (the Concatenated Nodes)

4. Characteristics Generic to Most Software
   4.1. Ease of Learning and Use
      4.1.1. User-Friendly Features
         4.1.1.1. Help Menu
         4.1.1.2. Visible or Audible Response to User’s All Action with the System
         4.1.1.3. Error Messages to a User’s Error
         4.1.1.4. Menu Selection Procedure
         4.1.1.5. Direct Manipulation
         4.1.1.6. Color Display
         4.1.1.7. Tutorial Introduction or Demonstration
      4.1.2. The Complexity of Use
4.1.2.1. Progressive Disclosure
4.1.2.2. The Availability of Any Readily Apparent Metaphors
4.1.2.3. The "Look and Feel" of the Command Structure

4.2. Search of Character String
4.3. Documentation
4.4. Vendor Supports
  4.4.1. Support Services through Hot Line (e.g., Toll Free Phone Number, 24 Hours and 7 Days Phone Service, Online Electronic Bulletin Board System)
  4.4.2. Policies Alerting Users to Changes and Enhancements
  4.4.3. Warranty

4.5. Software Maturity
  4.5.1. Age of Software on the Market
  4.5.2. Existence of Active User Group
  4.5.3. Literature (Other Than Advertising or Product Announcing) about the Software

5. Extra Functionality
Differences in Tasks and Users: Preliminary Considerations in Evaluating Software

1. Tasks

A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.

Two general functions are usually associated with hypertext environments: (1) the ability to retrieve information in a large information space, and (2) the ability to create non-sequential text (Knuth 1990).

Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind (Halasz 1988).

People having different tasks will use hypertext systems in different ways, so different hypertext mechanisms are needed to support different tasks (Nielsen 1989).

Task.1: Authoring

Authoring centres on the information creating and linking aspects. That is, hypertext systems can provide a means to support the construction of both individual and community information bases.

Systems designed primarily for authoring are generally characterized by the well-developed tools for creation and modification of the networks (Halasz 1988).

Intermedia was described as an environment "to link information together, create paths through a body of related material, annotate existing texts, and create notes that direct readers to either
bibliographic data or the body of the reference text (Yankelovich 1988)."

Task.2: Browsing

Browsing through traversing or graphical browsers is one of prevalent information retrieval in hypertext.

The ability to explore large bodies of information easily and naturally is the important issue to these experts. ... several mechanisms (e.g. graphical browsers, sophisticated search facilities, etc.) which are intended to support these exploratory tasks (Knuth 1990).

Systems designed primarily for browsing are generally characterized by relatively well-developed tools for "information presentation" and "exploratory browsing" (Halasz 1988).

2. Users

Individual difference is one of the major preliminary considerations in evaluating software. It is hard to expect that one software satisfies everybody.

Thus, it is necessary to identify yourself. What is best for one group of users may not be best for another, since different people will perform very differently (Nielsen 1990).

User.1: Not Familiar with Any Hypertext Software.

User.2: Familiar with at Least One Hypertext Software.

3. Different Evaluation Results

Depending on types of users and tasks, there should be different results in evaluating software.

Because of these two observations (differences in both users and tasks), there is little hope for a single, universal hypertext user interface design which will be optimal to everybody (Nielsen 1990).
Detailed Evaluation Criteria

1. Hypertextability

Hypertextability is the capability of hypertext software to construct a desired level of associative chaining of information and idea. Hypertextability consists of the followings:

Dynamics of Linking;

User's Manipulations of Nodes;

Traversing;

File Transformation from Other Applications (e.g. ASCII text, Word Perfect 5.1., Lotus 1-2-3, dBASE IV, graphics) into Hypertext;

Capability of Displaying or Creating Graphics;

and, Capability of the Simultaneous Production of Sound or Animation.

1.1. Dynamics of Linking

Dynamics of linking can be represented by the power and flexibility of linking methodology of hypertext software. Links connect nodes in the hypertext software by computer-supported relationships that permit rapid, easy movement across the network of nodes. There are great variety of links in hypertext systems. Dynamics of linking can be measured by two factors: number and type of links supported and the intuitiveness of linking procedure.

Create and modify links (Importance rate 4.68/5.00; rank 2/37) (Knuth 1991).
Units of Information Interconnected by Links. Users navigate in a hypermedia database by selecting links in order to travel from unit to unit (Aks cyn 1988);

User's Manipulation of Nodes and Links. The user can easily create new nodes and new links to new nodes (for annotation, comment, elaboration, etc.) or to existing nodes (for establishing new connections) (Conklin 1987);

Information-Chunk Sizes and Link-Attachment Locations. Several systems simply link nodes of equivalent size to each another, much like sending you from one card to another in the card catalog. Other systems provide links from many different subsets of information within the node and display those links in windows on the screen. In some systems, the links are part of the node or attached to the text information. In other systems, the links simply attach themselves to the basic information structure and are not altered by changes to the content of the node. ... Link and Node Classifications. A richer environment is created when the links or information chunks are classified by type (Gluck 1989);

Rearranging Links. Allow to rearrange links at the cost of added complexity (Hardman 1989);

Links. Software methodology that facilitates building of and access to nodes via links. Links connect nodes in the hypertext software by computer-supported relationships that permit rapid, easy movement across the network of nodes. There are great variety of links in hypertext systems. Here are some types of links (Horn 1989);

Associative Links. Hypertext connects nodes through nodes. What is most important is that these links are modifiable. Both the designers and user of many hypertext systems may add or change existing links to reflect their own knowledge structures. ... Organizational Structure. The structure of a hypertext determines and describes the system of links or relationships between the nodes or information units. The most important aspect of the characteristics is that the structure is modifiable, both by the developer and by the browsers (Jonassen 1989);
Whether links are tied to text (and move when it is edited) or to a position on the screen (and must be rebuilt if text is edited) (Myers 1990);
The ability to create, delete, store, and retrieve nodes, links, and anchors (Schnase 1988);
Power and flexibility of knowledge representation and inference and control techniques (Harmon 1988).

1.1.1. Number and Type of Links Supported

What types of links should there be? A predefined set of link types or Link types defined by user types. ... Should a link have internal structure? (Akscyn 1988);

Link Types. Can links have types? (Conklin 1987);
Ease of creating links (Myers 1990);
Multiple Links (Schnase 1988).

1.1.1.1. Text Editor Linking

The user links topics by typing text using text editor.

Explicit Intra-Document Links. Links of this class explicitly connect two parts of the same document, and include footnotes, *see also* cross-references, and pointers to figures, illustrations, or other non-textual components. When printed documents are converted to electronic ones it is essential to exploit this sort of knowledge by capturing the explicit intra-document links first (Glusko 1989).

1.1.1.2. Linking by Matrix Outlining

In a hierarchical organization, the user displays all possible parent (input) or child (output) topics to a specific topic on windows by touching a function key, then marks on the necessary topics to be linked to the current topic as a parent node or child node.

Matrix Outlining (MaxThink 1989).
1.1.1.3. Key Word Linking

The user lists the unique words in a file using a user-initiated function key and marks any necessary key word, then the system link the key words to all topics which contain the key word.

Keyword Links. They are created by the system and permit the users to find the location of specific words in the text. They aid in search for strings of information (Horn 1989).

Key Word Linking (MaxThink 1989).

1.1.1.4. File Linking

File linking is to link a topic to a file.

Explicit Inter-Document Links. These are connections between documents, which are easy to identify because they follow presentation conventions in the printed document (Glusko 1989);

"To" link. A connector whose source anchor is an element contained in a node. A link from content to structure. Following a "to" link causes what is presently displayed to be replaced by the contents of the destination node (Schnase 1988);

File Linking (MaxThink 1989).

1.1.1.5. Hierarchical Links

Hierarchical Links is organizational links which implement hierarchical tree linkages within the hypertext network. Terms are parent, child, and sibling.

Organizational links include tables of contents and other such hierarchical structuring (Horn 1989).

Hierarchies. In addition to the standard node and link networks, there is some special support for hierarchical organizations (Halasz 1988).
Cross-Hierarchical Links: the ability to directly connect pieces of information that cannot be connected easily on paper (Hardman 1989).

1.1.1.6. Referential Links

Referential Links. They connect points or regions of text and are non-hierarchical. In the terminology of some in the field, referential links go from link source (also called link point or link icon) to link destination (also called link region) (Horn 1989).

1.1.1.7. Cluster Linking

Define arbitrary aggregates (groups) of nodes. {Importance rate 3.67/5.00; rank 20/37} (Knuth 1990);

How can nodes be aggregated into larger structures? (Akscyn 1988);

Attributes. Can user-designated attribute/value pairs be associated with nodes or links? (Conklin 1987);

Paths. Can many links be strung together into a single version? (Conklin 1987);

Discrete Information Trails. Some systems have the capability to package chunks and link into discrete trails of information, also called paths or chains. Several systems also allow creation of customized webs or folders of information that re-use or share the nodes of information among various independent trails (Gluck 1989, 9-13);

Grouping of Information on the Screen. Strong grouping of information on the screen can aid comprehensibility (Hardman 1989);

Paths. Some hypertexts provide default routes or pre-determined paths through the hypertext. Path may be developed to adapt the information needs, different backgrounds, and different information gathering styles (Jonassen 1989);

Web Identification (Begory 1990);

Cluster Links. They are links that enable a user to organize a group of shorter pieces of
information as they proceed through the project (Horn 1989).

1.1.2. The Intuitiveness of Linking Procedure

The intuitiveness of linking procedure can be measured by the extent to which linking procedure is simple, intuitive, and consistent throughout the system.

The Intuitiveness of linking procedures (Howden 1990);

Ease in specifying links (Kearsley 1988);

Simplicity in Traversal. Navigation should be simple, intuitive, and consistent throughout the system (Kearsley 1988);

Ease of creating links (Myers 1990).

1.2. User's Manipulations of Nodes

Create and edit nodes {Importance rate 4.60/5.00; rank 3/37} (Knuth 1990);

Information "chunked" into small units. They are variously called notecards, frames, nodes, etc. Units may contain textual information, graphics, and so on. ... Creating, editing, and linking units, users build information structures for various structures. What is the appropriate data model for a node? ... What size should a node be? (Akscyn 1988);

A Network of Textual Nodes. The database is a network of textual (and perhaps graphical) nodes which can be thought of as a kind if hyperdocument. ... User's Manipulation of Nodes and Links. The user can easily create new nodes and new links to new nodes (for annotation, comment, elaboration, etc.) or to existing nodes (for establishing new connections) (Conklin 1987);

Programming Capabilities. Several current hypertext systems allow the user to build and execute a series of customized instructions for information manipulation or user interaction. These instructions may initiate. ... Formatting tables of data (Gluck 1989);
Nodes. The nodes are implemented using an extensible type hierarchy, with at least the text and graphics node types being provided with the basic system (Halasz 1988);

Nodes. A network of nodes which may be text and/or graphics. Nodes are the part of the hypertext network where the text or other media are located. A node may contain one idea, one sentence, or a whole documents as long as a book or chapter (Horn 1989);

Node. The container for information, being a chunk of information, in a hypertext. ... Named Node (A character string may be associated with nodes in order to provide organizational assistance to the user, facilitate searching, or facilitate filtering. ... The ability to create, delete, store, and retrieve nodes, links, and anchors (Schnase 1988);

Whether presentation must be compiled (Myers 1990).

1.2.1. Editing Capability of Nodes (e.g., moving, copying, deleting, inserting)

Text Editor. What editor is used to create and modify the contents of nodes? (Conklin 1987);

Wordprocessing. Annotation or modification of the contents of a particular information chunk in a hypertext requires invocation of computer program modules designed for such purposes. In hypertext systems this text editor is generally a set of wordprocessing instructions either built into the general environment of the system or readily invoked from it (Gluck 1989, 9-13);

Range of editing functions available (i.e. copying, moving, insertion, deletion, etc.) (Kearsley 1988);

Power and ease of use of included word processor (Myers 1990).

1.2.2. Capability of Modifying and Arranging both Nodes and Links Together (e.g., dividing, merging, sorting)

DIVIDE Linking. ... JOIN Linking (MaxThink 1989);
Sorting Data. ... Concatenating a subset of nodes (Gluck 1989, 9-13);

Easy to modify. ... Easy to maintain (Maintainability) (Hennell 1987, 81-91).

1.2.3. Displaying Graphics

Nodes capable of displaying bit-mapped graphics (Importance rate 4.56/5.00; rank 4/37). ...
Nodes capable of displaying vector graphics (Importance rate 4.17/5.00; rank 10/37) (Knuth 1990);

A Network of Textual Nodes. The database is a network of textual (and perhaps graphical) nodes which can be thought of as a kind of hyperdocument. ... Pictures or Graphics: is some form of pictorial or graphical information supported in addition to text? ... Graph-based: does the system support nonhierarchical (cross-referenced) links? (Conklin 1987);

Using Graphical Representations: uses of graphical representations can be extended beyond those possible in paper-based information (Hardman 1989);

Text only or graphics. ... Requirement for / support for EGA/VGA (Myers 1990);

Graphics (Shneiderman 1989, 35-74);

Graphics for notations (Heap 1988).

1.2.4. Optional Functionalities for Node Manipulation

1.2.4.1. Typed Nodes Defining a Set of Data Structure and Valid Operation

What types of nodes should there be? (Akscyn 1988);

Node Types. ... Most hypermedia system support just one type of node as a location to store any type of editable substance. ... Some systems, however, provide special types of nodes to help manage the structure of a hypermedia graph. ... The advantage of system supported node types is the ability to process nodes based on their type
(e.g., view only nodes of a specific type (Begory 1990);)

Link and Node Classifications. A richer environment is created when the links or information chunks are classified by type (Gluck 1989, 9-13);

Nodes. The nodes are implemented using an extensible type hierarchy, with at least the text and graphics node types being provided with the basic system (Halasz 1988);

Typed Node. Nodes may be typed in order to define a set of data structures and valid operations on nodes of a given type. A typed node is a member of a class of nodes (Schnase 1988, 4-49).

1.2.4.2. Annotation on Nodes

Annotate or comment on nodes created by others {Importance rate 4.20/5.00; rank 9/37} (Knuth 1990);

Collaboration. ... How can annotation supported? (Akscyn 1988);

Annotation via pop-up windows (Horn 1989);

Capability to switch between author and user mode to test ideas (Kearsley 1988);

Annotation (Schnase 1988, 4-49);

Annotation. Users leave their marks. This is the ability of the user to make additions to documents. These additions might take the form of "margin notes" that are kept together with document (Shneiderman 1989, 35-74).

1.2.4.3. Multiple Users' Access for Concurrent Authoring

Concurrent multi-user access {Importance rate 3.92/5.00; rank 16/37} (Knuth 1990);

Concurrent Multiusers. Can several users edit the hyperdocument at the same time? (Conklin 1987);

Sociability: allow multiple users to access, update and communicate {Expected} (Gaines 1988);
Multiple Users. A system that allows more than one user to edit information chunks or create links at the same time is called a concurrent multiuser system. Concurrent multiuser systems permit extensive sharing within a system but require special software or hardware features to prevent one user from inadvertently or maliciously affecting another user's webs, trails, links, or nodes (Gluck 1989, 9-13);

Distribution. Single-user or multi-user central server with concurrent control (Halasz 1988);

Multi-User Access. Access in some systems is provided to multiple users simultaneously, usually through a distributed network of computers (Jonassen 1989);

Concurrent Access. Allowing more than one user to access hypertext or the components of hypertext at the same time (Schnase 1988);

How information be jointly authored started by multiple users? (Akscyn 1988).

1.2.4.4. Versioning

Version control (Importance rate 3.80/5.00; rank 18/37) (Knuth 1990);

Versions: can nodes or links have more than a single version? (Conklin 1987);

Versioning: maintaining a history of changes to the contents of a node or set of nodes (Schnase 1988);

Recreatability allow users to recreate past perspectives. {Desirable} (Gaines 1988);

History of Changes. Another feature of some systems is version storing. These systems keep track of modification to the chunks, links, and webs, forming a change history that allows reconstruction of earlier states of versions (Gluck 1989, 9-13);

How can versioning be supported? (Akscyn 1988).

1.2.4.5. Creating Graphics

The system which has tools allowing the user
create graphic within the tools themselves may provide excellent user interface (Harmon 1988).

1.2.4.6. Sound

Controlling the simultaneous production of sound in a hypermedia system (Gluck 1989, 9-13).

1.2.4.7. Animation

Controlling the simultaneous production of animation in a hypermedia system (Gluck 1989).

1.3. Traversing

Ease of navigating within and between screens (Myers 1990);

By following links and opening windows successively to examine their contents (Conklin 1987);

Navigating to other nodes or document systems (Gluck 1989).

1.3.1. Backward Traversing

Retracing previous moves or commands made by the user, to return to a prior location or state of the system (Jones 1989);

Backtracking: ability to allow to go back along the path they took to reach the current screen in the hypertext. ... Going Back to Main Contents Screen: the availability to go directly back to the beginning of the hypertext (Hardman 1989);

"Back" Link: the ability to follow any link that points to or into a node as if it were bi-directional (Schnase 1988).

1.3.2. Browsing Map

Browsing map is to show a network diagram, specifying current node, goal nodes, and links.

Graphical browsers automatically generated by the system (Importance rate 3.92/5.00; rank 16/37) (Knuth 1990);
A browser displays some or all of the hyperdocument as a graph, providing an important measure of contextual and spatial cues to supplement the user's model of which nodes he is viewing and how they are related to each other and their neighbors in the graph. ... By navigating around the hyperdocument using a browser that displays the network graphically. ... Graphic Browser: is there a browser which graphically presents the nodes and links in the hyperdocument? (Conklin 1987);

Support for Browsing (Begory 1990);

Navigational Charts. To avoid disorientation and overload, some systems provide some assistance to the user for monitoring the current location within system and the relationship of the current information to the rest of the system's information. These are called navigational charts, graphical map, or browser (Gluck 1989, 9-13);

Guidability: indicate suitable paths through the hyperbase {Desirable} (Gaines 1988);

Overviews. Information access and structure editing is accomplished using "browsers" containing a graphical map of the network (Halasz 1988);

Graphical Browser (Schnase 1988, 4-49);

Location aids: maps, net displays, capability to return to previous screens, capability to repeat (Myers 1990).

1.3.5. Multiple Projection Windowing

Multiple overlapping windows {Importance rate 3.48/5.00; rank 25/37} (Knuth 1990, 91-107);

How should nodes be presented on the display? 1. Each node in a separate window, with multiple overlapping windows, perhaps of different sizes. 2. Single linear display, where each node is expanded in place. 3. Two nodes (linked each other at one screen, each taking up a full half of the screen. ... One unit per window. Units of information are displayed one per window. Systems vary in the number, size and arrangement of windows they permit (Akscyn 1988);
Nodes appear on the screen in windows. The size of the window is not necessarily tied to the size of the node. ... Multiple Node Presentation. Often a hypermedia user will want to see the contents of a new node before leaving the node they are currently viewing. The most common approach to this issue is to present each node in a window whose size is something less than that of the entire screen. Sometimes the size is a fixed portion of the screen but most often the user can change the size of each window and display as many nodes as they like (Begory 1990);

Windows Corresponding to Nodes. Windows on the screen correspond to nodes in the database on a one-to-one basis, and each has a name or title which is always displayed in the window. However, only a small number of nodes are ever "open" (as windows) on the screen at the same time.

Window System Operations. Standard window system operations are supported. Windows can be repositioned, resized, closed, put aside as small window icons. Clicking with the mouse on the icon of a closed window causes the window to open instantly (Conklin 1987);

Windows (Gluck 1989, 9-13);

Information-Chunk Sizes and Link-Attachment Locations. Several systems simply link nodes of equivalent size to each another, much like sending you from one card to another in the card catalog. Other systems provide links from many different subsets of information within the node and display those links in windows on the screen. In some systems, the links are part of the node or attached to the text information. In other systems, the links simply attach themselves to the basic information structure and are not altered by changes to the content of the node (Gluck 1989, 9-13);

Multiple windows (Halasz 1988);

Information Representation. Hypertext not only presents information, but it also represents the information in a multidimensional way. This representation of information is based upon a network of ideas and organizational structure that organizes the network (Jonassen 1989);
Range of screen formatting commands available (Kearsley 1988);

Overlapped Windows (Schnase 1988);

Screen formats. . . . Windows (Shneiderman 1989, 35-74);

Node Presentation (Begory 1990).

1.4. File Transformation from Other Applications (e.g., ASCII text, Word Perfect 5.1., Lotus 1-2-3, dBASE IV, graphics) into Hypertext

Import text, graphics, etc. from other applications (Importance rate 4.84/5.00; rank 1/37) (Knuth 1990);

Integration: provides for materials to carry gateways linking materials. Standard media require external indexes or directories to carry such linkage information (Essential). . . .

Programmability: provide for the incorporation of other programs (Desirable) (Gaines 1988);

Sharing Among Systems or Users. Some systems permit copying and sharing of customized collections of information with other users, systems, or databases as to improve the portability and compatibility of chunks and links among different systems (Gluck 1989);

Capability to import existing text or graphic files (Kearsley 1988);

Ability to import text and/or graphics (Myers 1990);

Import: allow the contents of a file to be brought into a node and incorporated into a hypertext (Schnase 1988, 4-49);

Integration. Sharing the wealth. To be useful, a hypertext system must be able to load information to and from other programs (such as text editors, database managers, spreadsheets) (Shneiderman 1989, 35-74);
Integration (Compatibility of input and output between stages/steps and tasks; Integration strategy with other software tools) (Heap 1988);

Input methods: should be efficient, with provisions for obtaining input by interface with another system if the data exists elsewhere (QED 1985, 9-23);

System interface: ability of tools to send and obtain data from other programs and data bases; hardware the tools run on (Harmon 1988);

External Databases & Interfaces (Meador 1984).

2. The Speed of Operation in Link Activation

The speed of operation in hypertext software may be measured in terms of link activation response time.

Rapid response time. (Importance rate 4.48/5.00; rank 6/37) (Knuth 1990);

The speed of operation (Howden 1990);

How fast should the system respond when following a link? "We believe that fast system response to selecting a link is the most important parameter of a hypermedia system. Newell (1977) has argued that response faster than 0.25 seconds will not benefit users (Akscyn 1988);

Link Activation Response Time (Begory 1990);

Current response times for most systems are in the one to two second range (Conklin 1986);

Performance. Performance of a hypertext system is measured in terms of response time. Users generally prefer faster response time, even down to a tenth of a second. However, fast response time encourages the users to take less time to think, which can lead to errors or disorientation. Response times of less than a tenth of a second can cause confusion because screen changes may be difficult to perceive. Furthermore, novices sometimes prefer a slower pace so they can read at a leisurely pace (Shneiderman 1989);

Runtime speed (Harmon 1988);
Performance (Heap 1988);

Efficiency (Hennell 1987, 81-91).

3. Printing Hypertext (the Concatenated Nodes)

Print user defined sequences of nodes {Importance rate 3.80/5.00; rank 17/37}. ... Print system defined sequences of nodes. {Importance rate 3.44/5.00; rank 26/37}. ... WYSIWYG display (Importance rate 3.96/5.00; rank 15/37) (Knuth 1990);

WYSIWYG stands for What-You-See-Is-What-You-Get. It is a phrase that describes a type of interface in which the ultimate product of the computer application, such as a graphic image or a text, is displayed upon the screen exactly as it will appear in its final output form (Jones 1989, 129);

How can material from a database be converted to paper form? (Akcyn 1988);

Print Facilities (Schnase 1988, 4-49);

Printing. Because hypertext involves the creation of nonlinear documents, translating hypertext into printed output (a linear process) is a significant challenge. When a hypertext document is printed, some convention for how to handle links is needed (Shneiderman 1989, 35-74);

Printing the concatenated nodes (Gluck 1989, 9-13);

Report Generator: ability to produce high quality formal reports quickly and easily (Meador 1984);

Output Versatility. Output reports should be versatile in scope and prepared on stock paper whatever possible (QED 1985, 9-23);

4. Characteristics Generic to Most Software

4.1. Ease of Learning and Use

If the system provides users for menu-driven procedure (rather than command languages) with default options, help menu with function key, function-keys-window, or feedback, then the system is good in ease of use and ease of learning. Feedback means that the system responds to user
error with error message and direction. ... Ease of learning: learning time must be minimal (Shneiderman 1989, 35-74);

If the system supports direct manipulation of the displayed information can result in easier learning and more complete remembering of interface conventions, and fewer errors (Jones 1989, 108);

Easy to understand (Comprehensibility). ... Easy to maintain (Maintainability). ... Easy to modify. ... Easy to read (Readability) (Hennell 1987, 81-91);

Ease of learning (Myers 1990);

User interface. Ease and flexibility of user interface development utilities, including graphics. Good graphics can be very important to developing effective user interfaces, and tools that allow developers to use graphics are very desirable (Harmon 1988);

Ease of Use (Meador 1984).

4.1.1. User-Friendly Features

There is no precise meaning to this overused term, user_friendly. In general, it refers to a computing system whose command interface is designed to simplify the task of learning and using the computer. If at any point in the system the user can request an explanation of what is currently happening in the system, and what the results of a particular action may be and selections on a menu provide a description of their function then the system is very much user friendly (Jones 1989, 128);

If the command structure gives the user a feeling of control over the system (e.g., the user feels comfortable in starting, proceeding, getting help, and terminating the operation of the system whenever necessary) and all user actions cause some visible or audible response or change in the system, even if that response is only an indication that input is not being accepted at that time, and the user is able to interrupt the system at any time and re-direct its activity then
the system is very much user-friendly (Jones 1989, 83-84);

If the system is "convenient methods of input" and is "tolerant and forgiving of mistakes", and produces "helpful and easy-to-understand error messages", and has "on-line assistance", and produces its "results in a natural, convenient format and medium" then the program would be said to be "user-friendly" (Schneider 1987, 394-398).

4.1.1.1. Help Features

Help features: the set of features provided as part of a computing system to present guidance to the user in the functioning of the system (Jones 1989);

It is, also called "help mode", simply on-line documentation that the user can browse through, to locate the information needed to solve a problem or correct an error (Schneider 1987, 395).

4.1.1.2. Visible or Audible Response to User’s All Action with the System

Supplying Feedback: feedback to the reader should be supplied whenever possible (Hardman 1989).

4.1.1.3. Helpful and Easy-To-Understand Messages to a User’s Error

If the error messages are user oriented and are directly at informing the user, in simple nontechnical terms, of what caused the error and how to recover then the system is much user_friendly. ... The system which provides meaningful and helpful response to a user’s error in interfacing with the system without abnormal termination is very much user_friendly. The meaningful and helpful response includes displaying an appropriate error message, explaining why the operations cannot be executed, and, if possible, supplying details about how to correct it (Schneider 1987, 395).

4.1.1.4. Menu Selection Procedure

Menu is a list of choices which is displayed on the screen for selection of one or more of the
alternatives by the user. The use of menus removes the requirement that the user remember and enter a set of statements to command the system to perform some action (Jones 1989);

Menu-selection interface (Akscyn 1988);

Ease of use. By the use of menus (rather than command languages) and default options, it is possible for a novice to begin using a system immediately) (Shneiderman 1989, 35-74);

Input methods: should be efficient, with provisions for obtaining input by interface with another system if the data exists elsewhere. (QED 1985, 9-23);

If the system provides the combination of menu selection and concrete desktop metaphor then it facilitates learning and performance. Notes. Interaction via menu selection is based on recognition reducing substantially the memory load of the (novice) user whereas using a command language requires recall of command codes for execution actions. For describing and naming in order to convey functions and structure of the system to the user, a "metaphor world" is used. The description via a metaphor world is related to an existing mental model which the user has already built up from other systems (Streitz 1987, 78-80);

Default value: when a series of choices or selections are presented to the user, the designation of one of these, or a different one altogether, as the choice if the user does not make an explicit selection (Jones 1989, 114);

Default values (options): a value (option) set by the system to be used unless the user specifies otherwise. It can reduce the user's burden. (Schneider 1987, 397)

4.1.1.5. Input Method for Direct Manipulation

Direct Manipulation is a method of interacting with a computing system in which the object of the interaction is continuously displayed. Interaction is accomplished with some form of input device rather than a command language and the effects of the interaction upon the object are immediately displayed (Jones 1989);
Direct manipulation paradigm. A single point-and-click without intermediate menu selection reduces the average time per operation to less than half what it is with typical menu-selection interfaces (Akscyn 1988);

Mouse driven (Halasz 1988);

Direct manipulation. ... Minimal number of user errors. Direct manipulations—the use of the mouse or touch—eliminates the possibility of errors from incorrectly typed commands). (Shneiderman 1989, 35-74);

Input devices (Gluck 1989, 9-13);

Mouse selection (Schnase 1988, 4-49);

Support for or requirement of mouse. ... Whether usable without a mouse, and how easily (Myers 1990);

Input methods: should be efficient, with provisions for obtaining input by interface with another system if the data exists elsewhere (QED 1985, 9-23);

If the system provides a single point-and-click without intermediate menu selection interface, then response time is good. Notes. A single point-and-click reduces average time per operation to less than half what it is with typical menu-selection interfaces (Akscyn 1988);

The system which facilitates progressive disclosure is not so much of the complexity of use for novice users (Jones 1989, 29).

4.1.1.6. Color Display

Control of color (text, background) (Kearsley 1988);

Support for screen colors (Myers 1990).

4.1.1.7. Tutorial Introduction or Demonstration

4.1.2. The Complexity of Use

The complexity of use (Howden 1990);
The system which facilitates "progressive disclosure" is not so much of "the complexity of use" (Jones 1989);

The system which allows the user to avoid the wordiness of either of the proceeding user aids or a tutorial introduction or some sections of the tutorial introduction is very much "easy of use". Giving the user the ability to skip over sections of the tutorial that are too basic for him will help to keep from boring more advanced users who nevertheless may need introduction to certain features. (Jones 1989, 105)

Non-complex. (Hennell 1987, 81-91);

Program Logic Complexity. Software package programming techniques should dictate that programming be performed in a logical, straightforward manner. (QED 1985, 9-23).

4.1.2.1. Progressive Disclosure

It is a way to reduce the apparent complexity of the system by presenting the user with only the information relevant to him at any point in time. Menu interfaces and icons (where data is represented in a compact format, and details are only presented upon explicit request) are methods of progressive disclosure (Jones 1989, 29);

Hiding Information: Inessential information can be kept hidden until requested by the reader (Hardman 1989).

4.1.2.2. The Availability of Any Readily Apparent Metaphors

A metaphor provides a simple explanation of the system which can be used to predict the system’s behavior. If the system provides the combination of menu selection and concrete desktop metaphor then it facilitates learning and performance. If with metaphor provided by the system users can predict what does that mean without explanation, the system can be said ease of learning and use.
A metaphor suggests that the computer is like something with which the user is already familiar (Rubinstein 1984, 43);

Icons displayed on the screen, as the metaphor, may represent a number of documents, file folders, and software programs. Icon is a simplified pictorial representation of an object, location, or function, usually having a physical resemblance in certain key or distinctive features. Icons may be used to replace commands in systems equipped with a pointing device, such as a mouse, that can be used to select the icon. (Jones 1989, 117)

The availability of any readily apparent metaphors that might appear in the software (buttons, books, files, etc.) (Howden 1990);

Link Icons. Windows can contain any number of link icons which represent pointers to other nodes in the database. The link icon contains a short textual field which suggests the contents of the node it points to. Clicking on a link icon with the mouse causes the system to find the referenced node and to immediately open a new window for it on the screen (Conklin 1987);

Links. The links are labeled but not typed, and are anchored using icons within the contents of both the source and destination nodes (Halasz 1988);

Buttons. Interface tools that facilitate the creation of arbitrary linkages in the text with buttons and (frequently) the easy manipulation of chunks of text and media through windows. Buttons are specific locations in the hypertext or on other media that permit the user to jump along a link to another node, usually with the click of a mouse or the pressing of a key. In one sense, buttons are the user-visible manifestations of links. ... Nodes may have different display metaphors, such as cards, pages, or windows (Horn 1989);

Spatiality: provide extensive spatial metaphors. (Expected) (Gaines 1988);

Interaction via menu selection is based on recognition reducing substantially the memory load of the (novice) user whereas using a command language requires recall of command codes for
execution actions. For describing and naming in order to convey functions and structure of the system to the user, a "metaphor world" is used. The description via a metaphor world is related to an existing mental model which the user has already built up from other systems (Streitz 1987, 78-80);

If the user's model accurately reflects the effects of the system, then he will be more successful in learning and using the system, and likely will perceive the system as easy to use (Jones 1989, 16).

4.1.2.3. The "Look and Feel" of the command structure

The "look and feel" of the command structure (Howden 1990);

"Look and Feel" is a phrase invented by Jack Lusso and Doug Derwin for the discussion of software user interface protection. Some have speculated that "look" refers to the appearance of the screen displays (i.e. the visual layout of words and pictorial features of the screen) and "feel" refers to how the program interacts with the user when performing its functions (i.e., the sequence of functions that occur when the user selects an option made available on the screen) (Samuelson 1989);

Menu Layout: the ability to provide for visually distinct menu heading from menu items. The currently selected menu item should be recognizable (Hardman 1989);

If the command structure gives the user a feeling of control over the system (e.g., the user feels comfortable in starting, proceeding, getting help, and terminating the operation of the system whenever necessary) and all user actions cause some visible or audible response or change in the system, even if that response is only an indication that input is not being accepted at that time, and the user is able to interrupt the system at any time and re-direct its activity then the system is very much user-friendly. (Jones 1989, 83-84);

4.2. Search of Character String

Search of character string can be measured by the
ability to search the character data for occurrences of a particular string of characters.

- Full string search of the entire hypertext {Importance rate 3.92/5.00; rank 16/37}.
  
  - ... Boolean search of the entire hypertext. {Importance rate 3.60/5.00; rank 22/37} (Knuth 1990);

  How can the user search for the information other than browsing? (Akscyn 1988);

  - String Search: can the hyperdocument be searched for strings (including keywords) ? (Conklin 1987);

Character String Searching. Many systems provide the ability to search the character data for occurrences of a particular string of characters. Depending upon the system, string search capabilities range from being very simple to quite sophisticated. The simple search mechanisms are only able to find the particular set of characters wherever they exist in the file. The more complex mechanisms can search for the string in specific locations, such as at the beginning of a word, sentence, or paragraph, or employ wild card characters and other sophisticated pattern matches (Gluck 1989, 9-13);

- Search/query, query-based access and full text string match or keyword matching (Halasz 1988);

Search Strategies: novel hypertext search strategies need to be explained to readers (Hardman 1989);

- Availability of search/replace functions for making changes (Kearsley 1988);

Searching is primarily a back-end function associated with the browser component of a hypertext system (content search, structure search, and spatial search) (Schnase 1988, 4-49);

4.3. Documentation

Documentation (Myers 1990);

Standards and Documentation. Procedural standards, interface standards, presentation and layout
standards, organization of documentation, and appropriate standards. (Heap 1988; 7.)

Documentation: for user, programmer, operations (Meador 1984).

4.4. Vendor Supports

Customer Support (Myers 1990);

Quality Assurance. ... Product Support and Lifespan. ... Maintenance procedures. ... Training facilities. ... Vendor/Supplier Characteristics. ... Market Credibility. ... Business Credibility. ... Training materials and training type (residential/in-house). ... Product Portfolio. ... Company Viability (Heap 1988);

Training: novice/advanced, systems/user. ... Support: consultant, hot line. ... Vendor viability: size of company, revenues (Meador 1984).

4.4.1. Support Services through Hot Line (e.g., Toll Free Phone Number, 24 Hours and 7 Days Phone Service, Online Electronic Bulletin Board System)

Consultancy support (Heap 1988);

Support: consultant, hot line (Meador 1984);

Training and vendor support: documentation, courses, consulting availability, and vendor experience with particular domains and hardware (Harmon 1988).

4.4.2. Policies Alerting Users to Changes and Enhancements

Policy on Enhancements, revisions, and new releases (Heap 1988).

4.4.3. Warranty

Warranty period (Heap 1988).

4.5. Software Maturity

Product Maturity: elapsed time since the product was launched, market share of the product and of rival products, number of active users, existence
of active user group, and development resources (Heap 1988).

4.5.1. Age of Software on the Market

4.5.2. Existence of Active User Group

4.5.3. Literature (other than advertising or product announcing) about the Software

5. Extra Functionality

The capability of the software package to be expanded to do extra functions.

Extensible: the functionality of a hypertext system may be extended by the user through use of a programming language that is integrated with the system (Schnase 1988, 4-49);

How can the functionality of the system be extended (Aks cyn 1988);

Extensibility: allow users to add new data and new perspectives (Expected) (Gaines 1988);

Expandability. It refers to the capability of the software package to be expanded to have more capability or to handle larger files. ...

Modifiability. Even if the modifiability of a package is not immediately critical to a user, it will be critical, over time, as the users of the package demand more capability (QED 1985, 9-23);

Adaptability: the tool can be used for more than one step/task (Heap 1988);

Expert System Function. A computer program that is designed to emulate the reasoning and decision making processes of a human expert in some procedural domain (Jones 1989);

Spreadsheet System Function. Spreadsheet is an electronic replacement for the traditional financial modeling tools: the accountant's columnar pad, pencil and calculator. In some ways, spreadsheet programs are to those tools what wordprocessors are to typewriters. Spreadsheets offer dramatic improvements in ease of creating, editing, and financial models. (Que' 1987, 12)
Database System Function. Database System provides the ability to create, read, write and maintain database files. The conventional database handles records with fixed length and limited scope. Examples of a database system include DBASE III, Multiplan and RBase.
Qualifiers

/* Qualifier 1
Name: USER
Q> < Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE
1> not familiar with any hypertext software.
2> familiar with at least one hypertext software.

/* Qualifier 2
Name: TASK
Q> < A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>
YOUR INTENDED PRIMARY TASK IS
1> browsing: using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition) or accessing large, complex information sources in various ways depending on diverse viewpoints.
authoring: using hypertext software as a tool for building full text database or an idea organizer.

/* Qualifier 3
Name: SPEED
Q> < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on a button (link source) and the show-up of linked node (link destination).>
THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS
1> slower than 2 second and faster than 0.25 second (poor response time).
2> faster than 2 second and slower than 1 second (fair response time).
3> faster than 1 second or slower than 0.25 second (good response time).

/* Qualifier 4
Name: PRINTING
Q> PRINTING HYPERTEXT (the concatenated nodes) IS
1> not available.
2> available.

/* Qualifier 5
Name: SEARCH
Q> CHARACTER STRING SEARCH IS
1> not available.
2> available.
/* Qualifier 6
Name: "LOOK and FEEL"
Q> < "Look" refers to the appearance of the screen displays (i.e., the visual layout of words or pictorial features of the screen). "Feel" refers to how the program interacts with the user when performing its functions (i.e., the sequence of functions that occur when the user selects an option made available on the screen).>
"LOOK AND FEEL" OF THE COMMAND STRUCTURE IS
1> clumsy.
2> fair.
3> smoothly functioning.

/* Qualifier 7
Name: METAPHORS
Q> < A metaphor provides a simple explanation of the system which can be used to predict the system’s behavior. It suggests that the computer is like something with which the user is familiar. Icons displayed on the screen, as the metaphor, may represent a number of documents, file folders, or software programs.>
ANY READILY APPARENT METAPHORS ARE
1> not available.
2> available.

/* Qualifier 8
Name: DOCUMENTATION
Q> DOCUMENTATION IS
1> poorly organized and inconvenient to use.
2> fairly well organized and convenient to use.
3> well organized and very convenient to use.

/* Qualifier 9
Name: HELP FEATURES
Q> ONLINE HELP FEATURES ARE
1> not available.
2> available.

/* Qualifier 10
Name: RESPONSE
Q> VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS
1> not available.
2> available.

/* Qualifier 11
Name: ERROR MESSAGE
Q> HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE
1> not available.
2> available.

/* Qualifier 12
Name: MENU SELECTION
Q> MENU SELECTION PROCEDURE IS
1> not provided.
2> provided without default option.
3> provided with default option.

/* Qualifier 13
Name: DIRECT INPUT
Q> INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS
1> not provided.
2> provided.

/* Qualifier 14
Name: COLOR DISPLAY
Q> COLOR DISPLAY IS
1> not available.
2> available.

/* Qualifier 15
Name: USER-FRIENDLY
Q> USER-FRIENDLY FEATURES ARE
1> poorly supported.
2> fairly well supported.
3> well supported.

/* Qualifier 16
Name: TUTORIAL
Q> TUTORIAL INTRODUCTION OR DEMONSTRATION IS
1> not provided.
2> provided.

/* Qualifier 17
Name: PROGRESSIVE
Q> < The apparent complexity of the system can be reduced by progressive disclosure which is presenting the user with only the information relevant to him or her at any point in time. Menu interfaces and icons (where data are represented in a compact format, and details are only presented upon explicit request) are methods of progressive disclosure.>
PROGRESSIVE DISCLOSURE IS
1> poorly provided.
2> fairly well provided.
3> well provided.

/* Qualifier 18
Name: EASE
Q> EASE OF LEARNING AND USE:
1> not easy.
2> easy.
3> very easy.

/* Qualifier 19
Name: SOUND
Q> SOUND IS
1> not available.
2> available.
/* Qualifier 20
Name: DISPLAY_GRAPHIC
Q> DISPLAYING GRAPHICS IS
1> not available.
2> available.

/* Qualifier 21
Name: TRANSFORMATION
Q> FILE TRANSFORMATION FROM OTHER APPLICATIONS (e.g., ASCII text, Word Perfect 5.1., Lotus 1-2-3, dBASE IV, graphics) INTO HYPERTEXT IS
1> not available.
2> available.

/* Qualifier 22
Name: TRAVERSING
Q> TRAVERSING IS
1> poorly supported.
2> fairly well supported.
3> well supported.

/* Qualifier 23
Name: NODES
Q> USER'S MANIPULATION OF NODES IS
1> poorly supported.
2> fairly well supported.
3> well supported.
/* Qualifier 24
Name: INTUITIVENESS
Q> LINKING PROCEDURE'S INTUITIVENESS IS
  1> poor.
  2> fair.
  3> good.

/* Qualifier 25
Name: LINK TYPES
Q> NUMBER AND TYPE OF LINKS ARE:
  1> poorly supported.
  2> fairly well supported.
  3> well supported.

/* Qualifier 26
Name: DYNAMICS
Q> DYNAMICS OF LINKING IS
  1> poorly supported.
  2> fairly well supported.
  3> well supported.

/* Qualifier 27
Name: HYPERTEXTABLE
Q> HYPERTEXTABILITY OF THE SOFTWARE IS
  1> poorly supported.
  2> fairly well supported.
  3> well supported.
/* Qualifier 28
Name: BACKTRACKING
Q> < Backward traversing is the navigating to go back along the path they took to reach the current position in the hypertext.>
BACKWARD TRAVERSING IS
1> not supported.
2> supported.

/* Qualifier 29
Name: BROWSING MAP
Q> < Browsing map is the graphical representation of both current location within the system and the relationship to the rest of the system by links.>
BROWSING MAP IS
1> not provided.
2> provided.

/* Qualifier 30
Name: WINDOW
Q> < Often users want to see the contents of a new node before leaving the node they are currently viewing. The most common approach to this issue is to present each node in a window whose size is something less than that of the entire screen. Multiple projection (or overlapping) windowing allows a number of windows which contain a node respectively to be open on the screen at the same time.>
MULTIPLE PROJECTION WINDOWING IS
1> not provided
2> provided.

/* Qualifier 31
Name: EDITING NODES
Q> CAPABILITY OF EDITING NODES (e.g., moving, copying, 
deleting, inserting) IS
1> not enough.
2> enough.

/* Qualifier 32
Name: MODIFYING
Q> CAPABILITY OF MODIFYING AND ARRANGING BOTH NODES AND 
LINKS ALTOGETHER (e.g., dividing, merging, sorting) IS
1> not enough.
2> enough.

/* Qualifier 33
Name: TYPED NODES
Q> TYPED NODES (defining a set of data structure and valid 
operation) IS
1> not available.
2> available.

/* Qualifier 34
Name: ANNOTATION
Q> ANNOTATION ON NODES IS
1> not available.
2> available.

/* Qualifier 35
Name: MULTIPLE USER
Q> MULTIPLE USERS' ACCESS (for concurrent authoring) IS
1> not available.
2> available.

/* Qualifier 36
Name: VERSIONING
Q> VERSIONING (maintaining a history of changes on nodes) IS
1> not available.
2> available.

/* Qualifier 37
Name: EDITOR LINKING
Q> TEXT EDITOR LINKING (linking topics by typing text using
text editor) IS
1> not available.
2> available.

/* Qualifier 38
Name: MATRIX OUTLINING
Q> In a hierarchical organization, the user may display
all possible parent (input) or child (output) topics to a
specific topic on windows by touching function keys. Then
the user may mark on the necessary topics to be linked to
the current topic as a parent or child node. This, matrix outlining, is one of the powerful way of linking.

MATRIX OUTLINING IS
1> not available.
2> available.

/* Qualifier 39
Name: KEY WORD LINKING
Q> < Users may list all the unique words in a file by touching a function key and mark on the necessary topics for linking. Then, the system automatically links the key words to all the topics which contain the key words. This, key word linking, is also one of the powerful way of linking.

KEY WORD LINKING IS
1> not provided.
2> provided.

/* Qualifier 40
Name: FILE LINKING
Q> FILE LINKING (linking a topic to a file or linking between files) IS
1> not provided.
2> provided.

/* Qualifier 41
Name: HIERARCHY LINKS
Q> HIERARCHICAL LINKING (implementing hierarchical tree
linkages within the hypertext having parent, child, or sibling nodes) IS
1> not provided.
2> provided.

/* Qualifier 42
Name: REFERENTIAL
Q> REFERENTIAL LINKING (linking topics without hierarchical relationship) IS
1> not provided.
2> provided.

/* Qualifier 43
Name: CLUSTER LINKING
Q> CLUSTER LINKING (linking a group of shorter pieces of information, called path or web) IS
1> not provided.
2> provided.

/* Qualifier 44
Name: SERVICES
Q> SUPPORT SERVICES THROUGH HOT LINE (e.g., toll free phone number, 24 hours phone service including weekend, online bulletin board system) IS
1> not available.
2> available.

/* Qualifier 45
Name: POLICY

Q> POLICIES ALERTING USERS TO CHANGES AND ENHANCEMENTS ARE
1> not available.
2> available.

/* Qualifier 46
Name: WARRANTY

Q> WARRANTY PERIOD IS
1> less than 1 year.
2> equal to or greater than 1 year.

/* Qualifier 47
Name: SUPPORT

Q> VENDOR SUPPORT IS
1> not enough.
2> enough.

/* Qualifier 48
Name: EXTRA FUNCTION

Q> EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS
1> not supported.
2> supported.

/* Qualifier 49
Name: AGE OF SOFTWARE

Q> AGE OF SOFTWARE ON THE MARKET IS
1> less than 1 year.
2> equal to or greater than 1 year.
3> not known.

/* Qualifier 50
Name: USER GROUPS
Q> USER GROUPS ARE
1> not active.
2> active.
3> not known.

/* Qualifier 51
Name: LITERATURE
Q> LITERATURE (other than advertising or product announcing) ABOUT THE SOFTWARE IS
1> not published.
2> published.
3> not known.

/* Qualifier 52
Name: MATURITY
Q> SOFTWARE MATURITY:
1> not matured.
2> matured.
3> not known

/* Qualifier 53
Name: COMPLEXITY
Q> COMPLEXITY:
1> very complex.
2> complex.
3> not complex.

/* Qualifier 54
Name: GENERIC
Q> < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>
CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE
1> poorly supported.
2> fairly well supported.
3> well supported.

/* Qualifier 55
Name: ANIMATION
Q> ANIMATION IS
1> not available.
2> available.

/* Qualifier 56
Name: CREATE_GRAPHIC
Q> CREATING GRAPHICS IS
1> not available.
2> available.
/* Qualifier 57
Name: NODE OPTIONAL
Q> OPTIONAL FUNCTIONALITIES FOR NODE MANIPULATION (e.g.
typed nodes, sound, annotation, multiple users' access,
creation of graphics, animation) ARE
1> poorly supported.
2> fairly well supported.
3> well supported.
Choices

/* Choice 1
Name: EXTREMELY POOR
C> This hypertext software is extremely poor for you to do your primary hypertext task (browsing or authoring).

/* Choice 2
Name: QUITE POOR
C> This hypertext software is quite poor for you to do your primary hypertext task (browsing or authoring).

/* Choice 3
Name: POOR
C> This hypertext software is poor for you to do your primary hypertext task (browsing or authoring).

/* Choice 4
Name: FAIR
C> This hypertext software is fair for you to do your primary hypertext task (browsing or authoring).

/* Choice 5
Name: GOOD
C> This hypertext software is good for you to do your primary hypertext task (browsing or authoring).
/* Choice 6

Name: EXCELLENT

C> This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).

/* Choice 7

Name: EXTREMELY EXCELLENT

C> This hypertext software is extremely excellent for you to do your primary hypertext task (authoring or browsing).
Rules of HEES

/* Rule 1

Name: FRIENDLY.1

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER’S ALL ACTION WITH THE SYSTEM IS {not available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER’S ERRORS ARE {not available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {not provided.}

and: COLOR DISPLAY IS {available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 2
Name: FRIENDLY.2

IF:
< Individual difference is one of the major preliminary
considerations in evaluating software. What is best
for one group of users may not be best for another
since different people perform very differently. There
is little hope for a single, universal hypertext
software which will be optimal to everybody. Thus, it
is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {not available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH
THE SYSTEM IS {available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S
ERRORS ARE {available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-
point-and-click by mouse, arrow keys) IS {not
provided.}
and: COLOR DISPLAY IS {available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not
provided.}

THEN:
USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 3
Name: FRIENDLY.3

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}
and: COLOR DISPLAY IS {available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:
USER-FRIENDLY FEATURES ARE {fairly well supported.}
RULE NUMBER: 4

RULE: FRIENDLY.4

IF:

< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {not available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {not available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}

and: COLOR DISPLAY IS {not available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 5
Name: FRIENDLY.5

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {not available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {not available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {not provided.}
and: COLOR DISPLAY IS {available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:
USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 6 
Name: FRIENDLY.6 

IF:  
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {not available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {not available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {not provided.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}
and: COLOR DISPLAY IS {not available.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
/ * RULE NUMBER: 7
Name: FRIENDLY.7

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {not available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}
and: COLOR DISPLAY IS {not available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:
USERS-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 8

Name: FRIENDLY.8

IF:

< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {not available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER’S ALL ACTION WITH THE SYSTEM IS {not available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER’S ERRORS ARE {not available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {not provided.}

and: COLOR DISPLAY IS {not available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 9
Name: FRIENDLY.9

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}

and: COLOR DISPLAY IS {available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {fairly well supported.}
/* RULE NUMBER: 10
Name: FRIENDLY.10

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {not available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}
and: COLOR DISPLAY IS {not available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:
USER-FRIENDLY FEATURES ARE {poorly supported.}
RULE NUMBER: 11
Name: FRIENDLY.11

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {not available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}

and: COLOR DISPLAY IS {available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 12
Name: FRIENDLY.12

IF:
< Individual difference is one of the major preliminary
considerations in evaluating software. What is best
for one group of users may not be best for another
since different people perform very differently. There
is little hope for a single, universal hypertext
software which will be optimal to everybody. Thus, it
is necessary to identify yourself.>
YOU ARE {not familiar with any hypertext software.}
and: ONLINE HELP FEATURES ARE {available.}
and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH
THE SYSTEM IS {available.}
and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S
ERRORS ARE {available.}
and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-
point-and-click by mouse, arrow keys) IS {not
provided.}
and: COLOR DISPLAY IS {not available.}
and: MENU SELECTION PROCEDURE IS {not provided.}
and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not
provided.}

THEN:
USER-FRIENDLY FEATURES ARE {poorly supported.}
/* RULE NUMBER: 13
Name: FRIENDLY.13

IF:
< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {provided.}

and: COLOR DISPLAY IS {not available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {fairly well supported.}
/* RULE NUMBER: 14
Name: FRIENDLY.14

IF:
  < Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>
  YOU ARE (not familiar with any hypertext software.) and:
  ONLINE HELP FEATURES ARE (available.) and:
  VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS (not available.) and:
  HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE (available.) and:
  INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS (provided.) and:
  COLOR DISPLAY IS (available.) and:
  MENU SELECTION PROCEDURE IS (not provided.) and:
  TUTORIAL INTRODUCTION OR DEMONSTRATION IS (not provided.)

THEN:
  USER-FRIENDLY FEATURES ARE (fairly well supported.)
/* RULE NUMBER: 15

Name: FRIENDLY.15

IF:

< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody. Thus, it is necessary to identify yourself.>

YOU ARE {not familiar with any hypertext software.}

and: ONLINE HELP FEATURES ARE {not available.}

and: VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS {not available.}

and: HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE {available.}

and: INPUT METHOD FOR DIRECT MANIPULATION (e.g., a single-point-and-click by mouse, arrow keys) IS {not provided.}

and: COLOR DISPLAY IS {available.}

and: MENU SELECTION PROCEDURE IS {not provided.}

and: TUTORIAL INTRODUCTION OR DEMONSTRATION IS {not provided.}

THEN:

USER-FRIENDLY FEATURES ARE {poorly supported.}
Rules number 16 through 984 are contained in a 3.5" HD/DS diskette, which is on file in the University of North Texas Library. Owing to their heavy volume, equivalent to about 900 pages, these rules are not printed in this chapter.

*****
/* RULE NUMBER: 986

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {fairly well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time).}

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well
and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is good for you to do your primary hypertext task (browsing or authoring).
- Confidence=8/10

/* RULE NUMBER: 987 */

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>
THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (poor response time).}

and: <Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY {e.g., expert system function, spreadsheet function, database management system function} IS {not supported.}

THEN:

This hypertext software is good for you to do your primary hypertext task (browsing or authoring).

- Confidence=8/10

/* RULE NUMBER: 988

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}
and: PRINTING HYPERTEXT (the concatenated nodes) IS 
   {available.}
and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}
and: < The speed of operation in hypertext can be measured 
in terms of link activation response time which is the 
length of time between user's pointing on button (link 
source) and the show-up of linked node (link 
destination).> 
THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS 
{faster than 2 second and slower than 1 second (poor 
response time). }
and: < Characteristics generic to most software include ease 
of learning and use, character string search, "Look and 
Feel" of the command structure, documentation, vendor 
support, and software maturity.>
CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well 
supported.}
and: EXTRA FUNCTIONALITY (e.g., expert system function, 
spreadsheet function, database management system 
function) IS {supported.}

THEN:

This hypertext software is good for you to do your 
primary hypertext task (browsing or authoring). 
- Confidence=8/10

/* RULE NUMBER: 989
IF:
< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (fair response time).}

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {fairly well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function,
spreadsheet function, database management system function) IS {not supported.}

THEN:

This hypertext software is good for you to do your primary hypertext task (browsing or authoring).
- Confidence=8/10

*/ RULE NUMBER: 990

IF:
< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user’s pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (fair
response time). }

and: <Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well fairly supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).
- Confidence=8/10

/* RULE NUMBER: 991

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS
and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time).}

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {poorly supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).
- Confidence=8/10

/* RULE NUMBER: 992

IF:

< A major preliminary consideration in evaluating
software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time).} 

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {poorly supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system
function) IS {not supported.}

THEN:

This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).
- Confidence=8/10

/* RULE NUMBER: 993 */

IF: < A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>
YOUR INTENDED PRIMARY TASK IS {browsing: using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition) or accessing large, complex information sources in various ways depending on various viewpoints.}
and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}
and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}
and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS
Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {not supported.}

THEN:

This hypertext software is extremely excellent for you to do your primary hypertext task (browsing or authoring).
- Confidence=9/10

/* RULE NUMBER: 994

IF: < A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}
and: PRINTING HYPERTEXT (the concatenated nodes) IS
{available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured
in terms of link activation response time which is the
length of time between user’s pointing on button (link
source) and the show-up of linked node (link
destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS
{faster than 2 second and slower than 1 second (good
response time). }

and: < Characteristics generic to most software include ease
of learning and use, character string search, "Look and
Feel" of the command structure, documentation, vendor
support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {fairly
well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function,
spreadsheet function, database management system
function) IS {supported.}

THEN:

This hypertext software is extremely excellent for you
to do your primary hypertext task (browsing or
authoring).

- Confidence=9/10
/* RULE NUMBER: 995

IF:
< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS (browsing: using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition) or accessing large, complex information sources in various ways depending on various viewpoints.)

and: PRINTING HYPERTEXT (the concatenated nodes) IS (not available.)

and: HYPERTEXTABILITY OF THE SOFTWARE IS (well supported.)

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user’s pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS (faster than 2 second and slower than 1 second (good response time).)

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>
CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is extremely excellent for you to do your primary hypertext task (browsing or authoring).

- Confidence=9/10

/* RULE NUMBER: 996

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {not available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user’s pointing on button (link
source) and the show-up of linked node (link destination).

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS
greater than 2 second and slower than 1 second (good response time).

and: < Characteristics generic to most software include ease
of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor
support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well
supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function,
spreadsheet function, database management system
function) IS {supported.}

THEN:

This hypertext software is extremely excellent for you
to do your primary hypertext task (browsing or
authoring).
- Confidence=9/10

/* RULE NUMBER: 997

IF:
< A major preliminary consideration in evaluating
software is the task which you plan to do with the
software. Many hypermedia systems have been designed
to support a specific task. Even generic systems are
usually designed with a target task domain in mind.>
YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}

and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time).}

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {not supported.}

THEN:

This hypertext software is extremely excellent for you to do your primary hypertext task (browsing or
IF:
< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>
YOUR INTENDED PRIMARY TASK IS {browsing: using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition) or accessing large, complex information sources in various ways depending on various viewpoints.}
and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}
and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}
and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user’s pointing on button (link source) and the show-up of linked node (link destination).>
THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time).}
and: < Characteristics generic to most software include ease

- Confidence=9/10
of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is extremely excellent for you to do your primary hypertext task (browsing or authoring).
- Confidence=10/10

/* RULE NUMBER: 999

IF:

< A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.>

YOUR INTENDED PRIMARY TASK IS {authoring: using hypertext software as a tool for building full text database or an idea organizer.}

and: PRINTING HYPERTEXT (the concatenated nodes) IS {available.}

and: HYPERTEXTABILITY OF THE SOFTWARE IS {well supported.}
and: < The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS {faster than 2 second and slower than 1 second (good response time). }

and: < Characteristics generic to most software include ease of learning and use, character string search, "Look and Feel" of the command structure, documentation, vendor support, and software maturity.>

CHARACTERISTICS GENERIC TO MOST SOFTWARE ARE {well supported.}

and: EXTRA FUNCTIONALITY (e.g., expert system function, spreadsheet function, database management system function) IS {supported.}

THEN:

This hypertext software is extremely excellent for you to do your primary hypertext task (browsing or authoring).

- Confidence=10/10
Data Collection and Analysis as a Procedure of Validation and Verification for an Expert System Development

The validation and verification for the prototype expert system, HEES, was based on the input and feedback of the knowledge engineer (the author of this dissertation), the human expert in hypertext domain (Dr. Norman Howden), the chair of the advisory committee (Dr. Ana D. Cleveland), and the advisory committee members (Dr. Donald B. Cleveland and Dr. Dewey E. Carroll). Prerau (1990, 300) defined the validation and verification for an expert system development:

Validation refers to determining whether the right system was built, that is whether the system does what it was meant to do and at an acceptable level of accuracy. Validating an expert system involves confirming that the expert system performs the desired task with a sufficient level of expertise. Verifications refers to determining whether the system was built right, that is, whether the system implementation correctly corresponds to its specification. Therefore verifying an expert system means confirming that the program accurately implements the acquired knowledge as documented.

Data were collected at three different locations: Dallas/Fort Worth metroplex area (DFW.MP), Houston metroplex area (Hou.MP), and College Station (Co.St.) in Texas (Table 4). The Information Science Building Room 206 of the
University of North Texas (UNT) was the main site for the data collection from 38 subjects in the Dallas/Fort Worth metroplex area for four months (May through August 1991). Major sources of subjects in the area were the followings: students who took a course in Hypercard authoring at UNT; the Hypercard User Group at UNT; attendants at conferences of the North Texas Association of Artificial Intelligence which consisted of DFW area universities and industries; participants in the VAX bulletin board system. An IBM PS/2 microcomputer with an 80386 processor was used as the hardware for collecting data.

In June 1991, the author visited Texas A&M University in College Station with a letter from Dr. Ana D. Cleveland to Dr. Craig Boyle. Two subjects participated in the experiment to generate data at a lab of the Computer Science Department, Texas A&M University. In July 1991, the author returned to College Station and three subjects joined the experiment at the Student’s Memorial Center, Texas A&M University. An IBM compatible microcomputer with an 80386 processor was used for both visits.

In July 1991, the author travelled to Houston, Texas for additional data collection. One subject participated in the experiment at the Clearlake campus library of University of Houston; one subject at the Looscan Branch of Houston Public Library; and three subjects at a computing lab of the main campus of University of Houston. All the experiments
in the Houston area used an IBM Compatible microcomputer with an 80386 processor.

Table 4

NUMBER OF PARTICIPANTS AT EACH LOCATION

DFW.MP: 38
Hou.Mp: 5
Co.St.: 5

Total 48

The major disciplines represented by the subjects were computer science, information systems, information science, natural sciences, humanities, business, education, library science, engineering, pharmacy, and medical science (Table 5). The subjects' experience levels with software were classified into three groups, using the questionnaire (Appendix B.): experience with at least one hypertext software (HPT); experience with conventional database management software (CDB); and no experience with the above software (NVC).
Table 5

MAJOR DISCIPLINES OF SUBJECTS

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer science:</td>
<td>7</td>
</tr>
<tr>
<td>Information system:</td>
<td>6</td>
</tr>
<tr>
<td>Information science:</td>
<td>3</td>
</tr>
<tr>
<td>Natural sciences:</td>
<td>9</td>
</tr>
<tr>
<td>Humanities:</td>
<td>8</td>
</tr>
<tr>
<td>Education:</td>
<td>5</td>
</tr>
<tr>
<td>Business:</td>
<td>4</td>
</tr>
<tr>
<td>Library science:</td>
<td>2</td>
</tr>
<tr>
<td>Engineering:</td>
<td>2</td>
</tr>
<tr>
<td>Pharmacy:</td>
<td>1</td>
</tr>
<tr>
<td>Medical science:</td>
<td>1</td>
</tr>
</tbody>
</table>

Total 48

Based upon their experience levels, all the subjects were randomly assigned to one of the control groups or to one of the four different treatment groups by the random number generator (Appendix A). The treatments consisted of one of the following options: 10 minutes training session and first access KnowledgePro with the aid of the HEES; 30 minutes training session and first access KnowledgePro with the aid of the HEES; 10 minutes training session and first access MaxThink’s with the aid of the
HEES; and 30 minutes training session and first access MaxThink's with the aid of the HEES.

The subjects evaluated two hypertext software: KnowledgePro (KP) and MaxThink's (MT). The results were converted to a percentage-type data (KP/MT*100 on the Figure 1 and Figure 2) in order to be processed by the data-analytic tool, the Analysis of Variance (ANOVA) for the factorial design with PROC ANOVA on SAS (Freund and Littel 1982, 67-74). SAS (Statistical Analysis System) has become one of the most prevalent statistical software packages since its introduction by North Carolina State University.

**Testing Hypothesis 1**

Hypothesis 1 stated "There is a significant difference between the hypertext software evaluation results by the control groups who don't use the HEES and the results by the experimental groups who use the HEES."

As shown in Figure 1, the profiles for testing Hypothesis 1 are not parallel. It indicates that there is an interaction effect. In other words, the effectiveness of the random assignment of the subjects into the control groups and the experimental groups varies from one
Figure 1

PROFILES FOR TESTING HYPOTHESIS 1

700 C

Legend

400 C

C: control group

T: treatment group

= =

250 C

= =

200 C

190

180

170

160

150 T C

140 C

130 C C

T T

120 C C

110 100

(KP/ Mt)*100

90 T

80 T

70 T T C

60 T T C

50 T C

HPT

CDB

NVC
experience level to the next. This interaction effect ratifies the factorial design, rather than randomized block design, for this experimental research.

The randomized block design is appropriate for the profiles without interaction effect, which means that extraneous factors to the experiment can be minimized or removed by blocking. The factorial design is to minimize the impact of extraneous factors by randomization, instead of blocking (Brightman, 207-223).

Table 6
ANOVA FOR TESTING HYPOTHESIS 1

<table>
<thead>
<tr>
<th>Degree of freedom</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>between groups:</td>
<td>1</td>
</tr>
<tr>
<td>within group:</td>
<td>46</td>
</tr>
<tr>
<td>total:</td>
<td>47</td>
</tr>
<tr>
<td>F value generated by the data:</td>
<td>6.51</td>
</tr>
<tr>
<td>F value from the Fisher table at the 0.01 level of significance:</td>
<td>4.05</td>
</tr>
</tbody>
</table>
As shown in Table 6 and Appendix C, the F value (variance ratio) generated by the data, 6.51, exceeds the F value obtained from the Fisher table with the degree of freedom (1, 46) at the 0.01 level of significance, 4.05. It means that there is a significant difference between the control groups and the treatment groups in the evaluation of hypertext software. Thus, Hypothesis 1 is not rejected at the 0.01 level of significance.

**Testing Hypothesis 2**

Hypothesis 2 stated that

\[ E_1 = E_2 = E_3 = \ldots = E_n \]

where \( E_n \), dependent variable for this experiment, refers to the hypertext software evaluation result by subject \( n \) using the expert system HEES. In other words, the HEES works consistently as a tool for the evaluation of hypertext software.

The profiles in Figure 2 are not parallel so that the effectiveness of treatments, the combination of different training session and the order of access to software, varies from one to the other. Thus, the factorial design, instead of the randomized block design, is appropriate.
Figure 2

PROFILES FOR TESTING HYPOTHESIS 2

Legend

1: Treatment group 1
2: Treatment group 2
3: Treatment group 3
4: Treatment group 4

(KP/MT) * 100

<table>
<thead>
<tr>
<th></th>
<th>HTP</th>
<th>CDB</th>
<th>NVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7
ANOVA FOR TESTING HYPOTHESIS 2

<table>
<thead>
<tr>
<th>Degree of freedom</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>between groups:</td>
<td>3</td>
</tr>
<tr>
<td>within group:</td>
<td>44</td>
</tr>
<tr>
<td>total:</td>
<td>47</td>
</tr>
</tbody>
</table>

F value generated by the data: 3.90
F value from the Fisher table at the 0.01 level of significance: 4.24

As shown in Table 7 and Appendix D, the F value (variance ratio) generated by the data, 3.90, is less than the F value obtained from the Fisher table with the degree of freedom (3, 44) at the 0.01 level of significance, 4.24. It means that there is no significant difference among the treatment groups in the evaluation of hypertext software. Thus, Hypothesis 2 is not rejected at the 0.01 level of significance.
CHAPTER VIII

CONCLUSION

Both Hypothesis 1 and Hypothesis 2 were not rejected at the 0.01 level of significance. It indicates that the expert system approach to the evaluation of hypertext engineering may be usable and useful. However, this research was conducted with the limitation and the assumptions as stated in the Chapter I: the limitation in random sampling; and the assumptions about the cognitive capability of subjects, the behavior of the investigator, and the evaluation criteria of hypertext engineering. In other words, the finding of this study is empirical and may be useful within the limitation and the assumptions.

Further research should be conducted to generalize the results of this research: sampling by volunteering for data collection may be enhanced by stratified sampling beyond the geographical boundaries across the nation; cognitive factors may be employed as experimental factors; and cost factors may be integrated as evaluation criteria.

In terms of software engineering, the studies should be conducted to enhance the procedure of validation and verification; and the number of hypertext software to test the HEES should be extended to include all available
hypertext software. Also, it is desirable to upgrade the product of this study, HEES, to cope with rapidly changing trends of hypertext software engineering; and the criterion of link activation response time should be periodically surveyed with recent products of hypertext software.
Algorithmic Languages. Programming languages which focus on data and algorithmic solution, such as FORTRAN, COBOL, PASCAL, and C.

Attribute. A property associated with some other object, such as display frame, a graphic, or a line of text, which modifies the object. Examples of attributes are color, highlighting, blinking, borders, fonts, and shapes (Jones 1989).

Backward Traversing. The method for returning to previously explored nodes and ultimately to the starting point or region. See also "backtracking."

Backtracking. Retracing previous moves or commands made by the user, to return to a prior location or state of the system (Jones 1989).

Bit-map Displays. The establishing of correspondence and a physical connection between a block of computer memory and the display, such that each pixel (picture element) in the display corresponds to one or more bits in the block of memory (Jones 1989).

Browsing Map. A network diagram specifying current node, goal node, and links.

Button. A portion of a hypertext that signals the beginning of a link.

Command Language. A computer language that is designed specifically for the encoding of procedural instructions for controlling the functions of the computer (Jones 1989).

Database System. A system providing the ability to create, read, write and maintain database files. The conventional database handles records with fixed length and limited scope. dBASE III and RBase are examples of the conventional database system.

Default Value. When a series of choices or selections are presented to the user, the designation of one of these, or a different one altogether, as the choice if the user does not make an explicit selection (Jones 1989, 114). Schneider (1987, 397) defines it as a value.
Direct Manipulation. A method of interacting with a computing system in which the object of the interaction is continuously displayed. Interaction is accomplished with some form of input device rather than a command language and the effects of the interaction upon the object are immediately displayed (Jones 1989).

Domain Knowledge. Knowledge about the problem domain. For instance, knowledge about the hypertext engineering in an expert system for evaluating hypertext software.

Expert System. A computer program that is designed to emulate the reasoning and decision making processes of a human expert in some procedural domain (Jones 1989). It is an advanced computer program that can, at high level of competence, solve difficult problems requiring the use of expertise and experience. It accomplishes this by employing knowledge of the techniques, information, heuristics, and problem-solving processes that human experts use to solve such problems. Expert systems thus provide a way to store human knowledge, expertise, and experience in computers. That is a way to clone human experts at least to some degree. Knowledge-based systems (KBS) or knowledge systems is often used as its synonym.

Expert System Shell. A software package that provides facilities to aid in expert system development. It is, often, called skeletal systems which do not contain domain knowledge but only the inference engine and support facilities. Support Facilities for expert system include aiding tools such as knowledge base editor, explanation facilities, interface, or debugging.

File Linking. Link a topic to a file, then display the file.

Frame. One of knowledge representation scheme. It is a structure containing information about a single entity, for example, a concept, item or class. A frame consist of slots which contain the value of the attributes of frame entity.

Help Facility. The set of features provided as part of a computing system to present guidance to the user in the
functioning of the system (Jones 1989). See also "on-line assistance."

Heuristic. "Rule of thumb" based upon experience and judgement. A set of heuristics provides guidelines for action, but not a specific formula for action (Jones 1989). Usually, heuristics can not be proved formally, and they may not be correct in all cases.

Hierarchy. An organization composed of distinct levels, with components at one level being superior to, or inclusive of, components at lower levels; and subordinate to, or included by, components at higher levels. Examples are corporate charts and family trees (Jones 1989).

Hypertextability. The capability of hypertext software to construct a desired level of associative chaining of information and ideas.

Icon. A simplified pictorial representation of an object, location, or function, usually having a physical resemblance in certain key or distinctive features. Icons may be used to replace commands in systems equipped with a pointing device, such as a mouse, that can be used to select the icon (Jones 1989, 117).

Induction System. One of reasoning process which decides what one needs to know about a problem from what one already knows.

Inference Engine. A component of expert system which provides the system control. It applies the expert domain knowledge to what is known about the present situation to determine new information about the domain. This process will, hopefully, lead to the solution of the problem.

Jump. The act of moving across a link.

Key Word Linking. A way of linking which lists the unique words in a file, then links all topics containing a specified word to a new topic containing that selected word.

Knowledge Base. A component of expert system which stores the facts and heuristics of domain experts, as obtained, usually by knowledge engineers. It also includes expert techniques on how and when to use these facts and heuristics.
Knowledge Engineering. The process of developing an expert system. This term is sometimes confined to the task of acquiring knowledge for an expert system from experts. However, this term, in this study, will be used broadly to indicate all of the technical aspects of developing expert systems.

Knowledge Engineers. The people who develop expert systems and perform the knowledge engineering.

Link. A connection between two ideas or information in hypertext.

Matrix Outlining. One way of linking which displays all possible input or output links to a topic, then mark links or break actual links to the current topic.

Menu. A list of choices which is displayed on the screen for selection of one or more of the alternatives by the user. The use of menus removes the requirement that the user remember and enter a set of statements to command the system to perform some action (Jones 1989).

Metaphor. A thing which provides a simple explanation of the system which can be used to predict the system's behavior. It "suggests that the computer is like something with which the user is already familiar (Rubinstein 1984, 43)." Icons displayed on the screen, as the metaphor, may represent a number of documents, file folders, and software programs.

Multiple Projection Windowing. A series of stratified division of a computer screen into several windows. See "Window."

Node. The beginning or end of a link. It is an idea or collection of ideas often represented visually by a button.

On-Line Assistance. Also called "help mode", simply on-line documentation that the user can browse through, to locate the information needed to solve a problem or correct an error (Schneider 1987, 395). See also "Help facilities."

Progressive Disclosure. A way to reduce the apparent complexity of the system by presenting the user with only the information relevant to him at any point in time. Menu interfaces and icons (where data is represented in a compact format, and details are only
presented upon explicit request) are methods of progressive disclosure (Jones 1989, 29).

Production Rule. One of the knowledge representation scheme. It represents a set of knowledge using If-Then rules. A set of rules can specify how the expert system should react to the changing data without requiring detailed knowledge about the flow of control in advance.

Response Time. Length of time between the completion of some action by the user (for example, the entry of a command) and the first indication by the system and communicated to the user that the system is carrying out the command (Jones 1989).

Rule. See "Production Rule."

Semantic Net. A scheme for knowledge representation. It was originally developed for use as psychological models of human memory but is now a standard representation method for expert systems. A semantic net consists of nodes and arcs. Nodes mean objects, concepts, or events while arcs describe the relations between nodes.

Spreadsheet System. An electronic replacement for the traditional financial modeling tools (the accountant's columnar pad, pencil and calculator). In some ways, spreadsheet programs are to those tools what wordprocessors are to typewriters. Spreadsheets offer dramatic improvements in ease of creating, editing, and financial models (Que' 1987, 12).

Symbolic Programming Language. One of computer programming languages which uses mostly logical operators to manipulate symbols, focusing on knowledge and heuristic solutions, such as LISP and PROLOG.

User_Friendly. In general, it refers to a computing system whose command interface is designed to simplify the task of learning and using the computer (Jones 1989, 128).

What-You-See-Is-What-You-Get (WYSIWYG). A phrase that describes a type of interface in which the ultimate product of the computer application, such as a graphic image or a text, is displayed upon the screen exactly as it will appear in its final output form (Jones 1989, 129).
Window. A rectangular area on the display, which has been separated from the rest of the display in some manner, usually by a distinctive border surrounding the region, and which may be treated independently of the rest of the display screen (Jones 1989).
APPENDIX A

BASIC PROGRAM FOR RANDOM NUMBER GENERATION
The IBM Basic
Version A4.00 Copyright IBM Corp. 1981, 1988
60225 Bytes free

Ok
LOAD"\heesbsc\rng.bsc"
Ok
LIST
10 RANDOMIZE TIMER
15 FOR I=1 TO 68
20 X = RND
30 Y = X * 16
40 Z = Y + 1
50 A = INT(Z)
60 PRINT A;
65 NEXT I
70 END
Ok
RUN
11 12 12 6 10 5 10 11 1 6 3 12 4 9 2 3 8 13 13 3 10 1
13 9 15 13 11 5 6 1 1 6 8 10 15 5 3 6 9 3 15 5 2 4 10
16 8 8 14 12 9 2 1 6 2 3 8 4 6 5 1 12 15 1 1 11 14
Ok
ILIST 2RUNC 3LOAD" 4SAVE" 5CONTc 6,"LPT1" 7TRONc 8TROFFc9KEY 0SCREEN
Ok
RUN
6 10 8 2 13 9 13 3 14 11 8 11 15 7 15 1 16 9 11 10
4 7 1 6 15 1 8 10 10 10 3 9 6 1 8 7 12 7 10 5 10 10
12 2 16 2 3 16 2 13 8 8 8 8 9 4 1 11 4 7 15 13 12 9
Ok
RUN
7 10 13 1 2 6 1 14 6 7 10 6 16 16 3 5 2 2 11 7 7 4 1
13 6 3 4 16 13 16 16 11 2 16 16 11 6 5 6 7 16 15 11 15
2 7 10 9 6 9 10 1 8 7 1 1 3 12 9 2 4 8 2 8 7 4 1
Ok
RUN
13 5 15 12 1 11 12 1 3 9 3 2 3 2 12 6 8 12 2 2 5 12
10 4 7 3 10 10 11 12 4 8 6 16 9 1 4 12 7 7 5 2 2 7 6
5 6 2 3 9 8 14 2 8 2 14 13 14 12 2 1 6 10 1 11
Ok
RUN
5 9 14 3 3 15 6 1 6 2 7 16 8 3 8 11 6 16 16 9 8 10 1
9 13 12 12 15 12 3 11 2 10 8 13 10 7 7 1 11 15 15 13 1
16 3 14 7 10 14 2 5 11 12 13 7 16 1 3 13 12 10 6 7 2
Ok

ILIST 2RUNC 3LOAD" 4SAVE" 5CONTc 6,"LPT1" 7TRONc 8TROFFc9KEY 0SCREEN
 Ok
QUESTIONNAIRE

In order to protect the subjects' privacy, any personal data, obtained by this questionnaire, will not be reported individually but used only for the experimental study in summarized-by-group formats. According to the level of computer software usage, the subjects will be divided into the following groups: novice group; experienced group with hypertext software; and, experienced group with other software (databases, spreadsheets, or online textual databases).

1. Have you taken any courses which taught hypertext software?
   (_____Yes, _____No)

   If yes, please list the course titles.

2. Have you used any hypertext software package to construct databases or to author texts?
   (_____Yes, _____No)

   If yes,
   a) how many files did you create approximately?

   b) please list the names of the hypertext software packages you used?
3. Have you taken any courses which taught software packages for database managements, spreadsheets, or online textual databases?
   (_____Yes, _____No)

   If yes,
   a) please list the course titles.
   b) please list the name of software packages used in the courses.

   (e.g., DBase III, Lotus 123)

4. Have you used any software packages for database managements, spreadsheets, or online textual databases?
   (_____Yes, _____No)

   If yes, please list the names of the software packages?

5. Do you have any experience with other computer software packages than the above packages?
   (_____Yes, _____No)

   If yes, please list the names of the software packages?

Your Initial ____________________
APPENDIX C

SAS RUN FOR TESTING HYPOTHESIS 1
SAS(R) LOG OS SAS 5.18 VS2/MVS JOB IH3I STEP SAS 22:03 WEDNESDAY, AUGUST 28, 1991

NOTE: COPYRIGHT (C) 1984, 1988 SAS INSTITUTE INC., CARY, N.C. 27512, U.S.A.
NOTE: THE JOB IH3I HAS BEEN RUN UNDER RELEASE 5.18 OF SAS AT UNIVERSITY OF NORTH TEXAS (02152001).
NOTE: CPUID VERSION = FF SERIAL = 010152 MODEL = 3081.
NOTE: SAS OPTIONS SPECIFIED ARE:
SORT=5

YOUR SERVICE AGREEMENT HAS EXPIRED FOR THE FOLLOWING PRODUCT(S):
*SAS/OR
PLEASE CONTACT YOUR COMPUTING INSTALLATION'S USER SERVICE PERSONNEL OR INSTALLATION'S SAS REPRESENTATIVE.
(*) - DENOTES EXPIRATION ERROR, OTHERWISE, EXPIRATION WARNING)

1    TITLE WEE;
2    DATA XPRMNT;
3    INPUT GRP $ XPRNC $ EVLTN;
4    LABEL GRP=TREATMENTS
5    XPRNC=EXPERIENCE
6    EVLTN=EVALUATION;
7    CARDS;

WARNING 520: OBSOLETE FORM OF STATEMENT OR TEXT82 OPTION INCORRECT.

NOTE: THE DATA SET WORK XPRMNT HAS 48 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
NOTE: THE DATA STATEMENT USED 0.05 SECONDS AND 552K.

56 PROC SORT DATA=XPRMNT; BY GRP XPRNC;
NOTE: 3 CYLINDERS DYNAMICALLY ALLOCATED ON SYSDA FOR EACH OF 3 SORT WORK DATA SETS.
NOTE: THE DATA SET WORK XPRMNT HAS 48 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
NOTE: THE PROCEDURE SORT USED 0.28 SECONDS AND 952K.

57 PROC MEANS MEAN STDERR. BY GRP XPRNC;
58 OUTPUT OUT=FACTMEAN MEAN=GRNDMEAN;
NOTE: THE DATA SET WORK FACTMEAN HAS 8 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
NOTE: THE PROCEDURE MEANS USED 0.10 SECONDS AND 856K AND PRINTED PAGE 1.

59 PROC PRINT DATA=FACTMEAN;
NOTE: THE PROCEDURE PRINT USED 0.08 SECONDS AND 820K AND PRINTED PAGE 2.

80 PROC PLOT DATA=FACTMEAN;
81 PLOT GRNDMEAN=XPRNC=GRP / HPOS=70 VPOS=40;
NOTE: THE PROCEDURE PLOT USED 0.10 SECONDS AND 828K AND PRINTED PAGE 3.

52 PROC ANOVA DATA=XPRMNT;
53 CLASS GRP XPRNC;
54 MODEL EVLTN=GRP*XPRNC;
NOTE: THE PROCEDURE ANOVA USED 0.14 SECONDS AND 844K AND PRINTED PAGES 4 TO 5.
NOTE: SAS USED 952K MEMORY.
ERROR: ERRORS ON PAGES 1.
NOTE: SAS INSTITUTE INC.
SAS CIRCLE
PO BOX 8000
CARY, N.C. 27512-8000
<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>LABEL</th>
<th>MEAN</th>
<th>STD ERROR OF MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=CNTRL EXP</td>
<td>120.375</td>
<td>19.57845124</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=CDB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=CNTRL EXP</td>
<td>218.750</td>
<td>78.67626934</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=HPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=CNTRL EXP</td>
<td>124.000</td>
<td>16.09458649</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=NVC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=TRTMT EXP</td>
<td>87.875</td>
<td>8.59908114</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=CDB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=TRTMT EXP</td>
<td>78.000</td>
<td>10.81004824</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=HPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVLTN</td>
<td>TREATMENTS=TRTMT EXP</td>
<td>82.625</td>
<td>7.38467888</td>
</tr>
<tr>
<td></td>
<td>EXPERIENCE=NVC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBS</td>
<td>GRP</td>
<td>XPRNC</td>
<td>GRNDMEAN</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>1</td>
<td>CNTRL</td>
<td>CDB</td>
<td>120.375</td>
</tr>
<tr>
<td>2</td>
<td>CNTRL</td>
<td>HPT</td>
<td>218.750</td>
</tr>
<tr>
<td>3</td>
<td>CNTRL</td>
<td>NVC</td>
<td>124.000</td>
</tr>
<tr>
<td>4</td>
<td>TRTMT</td>
<td>CDB</td>
<td>87.875</td>
</tr>
<tr>
<td>5</td>
<td>TRTMT</td>
<td>HPT</td>
<td>78.000</td>
</tr>
<tr>
<td>6</td>
<td>TRTMT</td>
<td>NVC</td>
<td>82.825</td>
</tr>
</tbody>
</table>
HEES
ANALYSIS OF VARIANCE PROCEDURE
CLASS LEVEL INFORMATION

<table>
<thead>
<tr>
<th>CLASS</th>
<th>LEVELS</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP</td>
<td>2</td>
<td>CNTRL TRTMT</td>
</tr>
<tr>
<td>XPRNC</td>
<td>3</td>
<td>CDB HPT NVC</td>
</tr>
</tbody>
</table>

NUMBER OF OBSERVATIONS IN DATA SET = 48
### Analysis of Variance Procedure

**Dependent Variable: EVLTN**

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>PR &gt; F</th>
<th>R-Square</th>
<th>C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>5</td>
<td>111591.35416667</td>
<td>22318.27083333</td>
<td>2.36</td>
<td>0.0560</td>
<td>0.219693</td>
<td>81.9058</td>
</tr>
<tr>
<td>Error</td>
<td>42</td>
<td>398350.12500000</td>
<td>9438.90773810</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>47</td>
<td>507941.47816667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>ANOVA SS</th>
<th>F Value</th>
<th>PR &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP</td>
<td>1</td>
<td>61418.52083333</td>
<td>8.51</td>
<td>0.0145</td>
</tr>
<tr>
<td>XPRNC</td>
<td>2</td>
<td>21276.54166667</td>
<td>1.13</td>
<td>0.3335</td>
</tr>
<tr>
<td>GRP*XPRNC</td>
<td>2</td>
<td>22786.29166667</td>
<td>1.33</td>
<td>0.2281</td>
</tr>
</tbody>
</table>
APPENDIX D

SAS RUN FOR TESTING HYPOTHESIS 2
**NOTE:** COPYRIGHT (C) 1984, 1988 SAS INSTITUTE INC. CARY, N.C. 27512, U.S.A.

**NOTE:** THE JOB IH31 HAS BEEN RUN UNDER RELEASE 5.18 OF SAS AT UNIVERSITY OF NORTH TEXAS (021520D1).

**NOTE:** CPUID VERSION = FF SERIAL = 010152 MODEL = 3081.

**NOTE:** SAS OPTIONS SPECIFIED ARE:

- SORT=5

**YOUR SERVICE AGREEMENT HAS EXPIRED FOR THE FOLLOWING PRODUCT(S):**

- *SAS/OR*

**PLEASE CONTACT YOUR COMPUTING INSTALLATION'S USER SERVICE PERSONNEL OR INSTALLATION SAS REPRESENTATIVE.**

(*) - DENOTES EXPIRATION ERROR. OTHERWISE, EXPIRATION WARNING.

```plaintext
1  TITLE HEES;
2  DATA XPRMNT;
3    INPUT GRP & XPRMNC & EVLTN;
4    LABEL GRP=TREATMENTS
5    XPRMNC=EXPERIENCE
6    EVLTN=EVALUATION;
7    CARDS;

WARNING 62D: OBSOLETE FORM OF STATEMENT OR TEXT82 OPTION INCORRECT.

**NOTE:** DATA SET WORK XPRMNT HAS 24 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
**NOTE:** THE DATA STATEMENT USED 0.05 SECONDS AND 552K.

32  PROC SORT DATA=XPRMNT; BY GRP XPRMNC;
**NOTE:** 5 CYLINDERS DYNAMICALLY ALLOCATED ON SYSDA FOR EACH OF 3 SORT WORK DATA SETS.
**NOTE:** DATA SET WORK XPRMNT HAS 24 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
**NOTE:** THE PROCEDURE SORT USED 0.28 SECONDS AND 952K.

33  PROC MEANS MEAN STDERR; BY GRP XPRMNC;
34  OUTPUT OUT=FACTMEAN MEAN=GRNDMEAN;
**NOTE:** THE DATA SET WORK FACTMEAN HAS 12 OBSERVATIONS AND 3 VARIABLES. 1676 OBS/TRK.
**NOTE:** THE PROCEDURE MEANS USED 0.10 SECONDS AND 659K AND PRINTED PAGE 1.

35  PROC PRINT DATA=FACTMEAN;
**NOTE:** THE PROCEDURE PRINT USED 0.08 SECONDS AND 820K AND PRINTED PAGE 2.

36  PROC PLOT DATA=FACTMEAN;
37    PLOT GRNDMEAN*XPRMNC*GRP / HPDS=70 VPOS=40.
**NOTE:** THE PROCEDURE PLOT USED 0.10 SECONDS AND 628K AND PRINTED PAGE 3.

38  PROC ANOVA DATA=XPRMNT;
39    CLASS GRP XPRMNC;
40    MODEL EVLTN=GRP*XPRMNC;
**NOTE:** THE PROCEDURE ANOVA USED 0.13 SECONDS AND 844K AND PRINTED PAGES 4 TO 5.
**NOTE:** SAS USED 932K MEMORY.

ERROR: ERRORS ON PAGES 1.

**NOTE:** SAS INSTITUTE INC.
SAS CIRCLE
PO BOX 8000
CARY, N.C. 27512-8000
```
<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>LABEL</th>
<th>MEAN</th>
<th>STD ERROR OF MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TREATMENTS<em>T10KP EXPERIENCE</em>CDB</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>104.00000000</td>
<td>21.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T10KP EXPERIENCE</em>HPT</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>75.00000000</td>
<td>8.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T10KP EXPERIENCE</em>NVC</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>100.00000000</td>
<td>25.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T10MT EXPERIENCE</em>CDB</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>73.50000000</td>
<td>6.50000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T10MT EXPERIENCE</em>HPT</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>60.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T10MT EXPERIENCE</em>NVC</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>73.50000000</td>
<td>6.50000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30KP EXPERIENCE</em>CDB</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>104.00000000</td>
<td>21.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30KP EXPERIENCE</em>HPT</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>115.00000000</td>
<td>35.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30KP EXPERIENCE</em>NVC</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>91.50000000</td>
<td>8.50000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30MT EXPERIENCE</em>CDB</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>70.00000000</td>
<td>10.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30MT EXPERIENCE</em>HPT</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>62.00000000</td>
<td>5.00000000</td>
</tr>
<tr>
<td></td>
<td>TREATMENTS<em>T30MT EXPERIENCE</em>NVC</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>EVLN</td>
<td>EVALUATION</td>
<td>85.50000000</td>
<td>5.50000000</td>
</tr>
</tbody>
</table>
HEES
ANALYSIS OF VARIANCE PROCEDURE
CLASS LEVEL INFORMATION

<table>
<thead>
<tr>
<th>CLASS</th>
<th>LEVELS</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP</td>
<td>4</td>
<td>T10KP T10MT T30KP T30MT</td>
</tr>
<tr>
<td>XPRNC</td>
<td>3</td>
<td>CDB HPT NVC</td>
</tr>
</tbody>
</table>

NUMBER OF OBSERVATIONS IN DATA SET = 24
### ANALYSIS OF VARIANCE PROCEDURE

**DEPENDENT VARIABLE: EVLTN**

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DF</th>
<th>SUM OF SQUARES</th>
<th>MEAN SQUARE</th>
<th>F VALUE</th>
<th>PR &gt; F</th>
<th>R-SQUARE</th>
<th>C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL</td>
<td>11</td>
<td>7913.33333333</td>
<td>719.39393939</td>
<td>1.39</td>
<td>0.2901</td>
<td>0.560064</td>
<td>27.4754</td>
</tr>
<tr>
<td>ERROR</td>
<td>12</td>
<td>6218.00000000</td>
<td>518.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CORRECTED TOTAL</td>
<td>23</td>
<td>14128.33333333</td>
<td>518.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SOURCE**

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DF</th>
<th>ANOVA SS</th>
<th>F VALUE</th>
<th>PR &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP</td>
<td>3</td>
<td>6085.00000000</td>
<td>3.90</td>
<td>0.0370</td>
</tr>
<tr>
<td>XPRMC</td>
<td>2</td>
<td>390.58333333</td>
<td>0.38</td>
<td>0.6838</td>
</tr>
<tr>
<td>GRP*XPRMC</td>
<td>6</td>
<td>1457.75000000</td>
<td>0.47</td>
<td>0.8186</td>
</tr>
</tbody>
</table>
APPENDIX E

SCRIPT FOR TEN MINUTES' TRAINING SESSION
In ten minutes, subjects assigned to the Treatment Group 1 and the Treatment Group 3 are trained with the script about qualifiers and choice.

**QUALIFIERS**

Qualifiers represent the evaluation criteria of hypertext engineering in the expert system, HEES. The printout of the qualifiers for the HEES provides guidelines and an input data collection sheet for the subjects involved in the experiment with KnowledgePro and MaxThink's. In the training sessions, all the subjects, except the control group who will not use the HEES, will receive both the printout and the oral explanation within the designated time period. Entry level qualifiers will be answered by the subjects, choosing one of the given values. Upper level qualifiers, whose names are preceded by one or more asterisks (*) in the printout, will be answered by the HEES using the backward chaining method.

Name: USER

< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody.

YOU ARE
1. NOT FAMILIAR WITH ANY HYPERTEXT SOFTWARE.
2. FAMILIAR WITH AT LEAST ONE HYPERTEXT SOFTWARE.

Name: TASK

< Another major preliminary consideration in evaluation software is task which you would like to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.

YOUR INTENDED PRIMARY TASK IS
1. BROWSING using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition, etc.) or accessing large, complex information sources in various ways depending on various viewpoints.
2. AUTHORING  using hypertext software as a tool for building full text database or an idea organizer.

Name: BACKWARD

< Backward traversing is the navigating to go back along the path they took to reach the current screen in the hypertext.>

BACKWARD TRAVERSING IS
1. NOT SUPPORTED.
2. SUPPORTED.

Name: BROWSING MAP

< Browsing map is the graphical presentation of current location within the system and the relationship to the rest of the system by links.>

BROWSING MAP IS
1. NOT AUTOMATICALLY GENERATED BY THE SYSTEM.
2. AUTOMATICALLY GENERATED BY THE SYSTEM.

Name: WINDOW

< ‘Often a hypermedia user will want to see the contents of a new node before leaving the node they are currently viewing. The most common approach to this issue is to present each node in a window whose size is something less than that of the entire screen. [Bego90]’ Multiple projection (or overlapping) windowing allows a number of windows which contain a node respectively to be open on the screen at the same time.>

MULTIPLE PROJECTION WINDOWING IS
1. NOT PROVIDED.
2. PROVIDED.

* Name: TRAVERSING
* TRAVERSING IS
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.

Name: TYPED NODES
**Typed Nodes** (defining a set of data structure and valid operation) is
1. NOT AVAILABLE.
2. AVAILABLE.

Name: **Annotation**

**Annotation on Nodes is:**
1. NOT AVAILABLE.
2. AVAILABLE.

Name: **Multiple User**

**Multiple Users' Access** (for concurrent authoring) is:
1. NOT AVAILABLE.
2. AVAILABLE.

Name: **Versioning**

**Versioning** (maintaining a history of changes of a node or set of nodes) is
1. NOT AVAILABLE.
2. AVAILABLE.

Name: **Create Graphic**

**Capability of Creating Graphics** is
1. NOT PROVIDED.
2. PROVIDED.

Name: **Sound**

**Capability of Sound** is
1. NOT PROVIDED.
2. PROVIDED.

Name: **Animation**

**Capability ofAnimation** is
1. NOT PROVIDED.
2. PROVIDED.

* Name: **Optional Functions**

* **Optional Functionalities for Editing Nodes are**
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.

Name: EDITING NODES

CAPABILITY OF EDITING NODES (e.g., moving, copying, deleting, inserting, etc.) IS
1. NOT ENOUGH.
2. ENOUGH.

Name: MODIFYING

CAPABILITY OF MODIFYING (e.g., dividing, merging, sorting, etc.) IS
1. NOT ENOUGH.
2. ENOUGH.

Name: DISPLAY GRAPHIC

CAPABILITY OF DISPLAYING GRAPHICS IS
1. NOT PROVIDED.
2. PROVIDED.

** Name: NODES

** USER'S MANIPULATION OF NODES IS
** 1. POORLY SUPPORTED.
** 2. FAIRLY WELL SUPPORTED.
** 3. WELL SUPPORTED.

Name: EDITOR LINKING

TEXT EDITOR LINKING (linking topics by typing text using text editor) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: MATRIX OUTLINING

< In a hierarchical organization, the user may display all possible parent (input) or child (output) topics to a specific topic on windows by touching function keys. Then the user may mark on the necessary topics to be linked to the current topic as a parent or child node. This, matrix outlining, is a one of the powerful way of linking.>
MATRIX OUTLINING IS
1. NOT FACILITATED.
2. FACILITATED.

Name: KEY WORD LINKING

< The user may list all the unique words in a file by touching a function key and mark on the necessary topics. Then, the system automatically links the key words to all the topics which contain the key words. This, key word linking, is also one of the powerful ways of linking.>

KEY WORD LINKING IS
1. NOT FACILITATED.
2. FACILITATED.

Name: FILE LINKING

FILE LINKING (linking a topic to a file or linking between files) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: HIERARCHY LINKS

HIERARCHICAL LINKING (implementing hierarchical tree linkages within the hypertext having parent, child, or sibling nodes) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: REFERENTIAL

REFERENTIAL LINKING (linking points or regions of text without hierarchical relationship [Horn89]) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: CLUSTER LINKING

CLUSTER LINKING (linking a group of shorter pieces of information, called path or web) IS
1. NOT FACILITATED.
2. FACILITATED.

* Name: LINK TYPES
* NUMBER AND TYPE OF LINKS SUPPORTED ARE
  1. POORELY SUPPORTED.
  2. FAIRLY WELL SUPPORTED.
  3. WELL SUPPORTED.

Name: INTUITIVENESS

THE INTUITIVENESS OF LINKING PROCEDURE IS
1. NOT INTUITIVE.
2. INTUITIVE.
3. VERY INTUITIVE.

** Name: DYNAMICS

** DYNAMICS OF LINKING IS
** 1. POORELY SUPPORTED.
** 2. FAIRLY WELL SUPPORTED.
** 3. WELL SUPPORTED.

Name: TRANSFORMABILITY

TRANSFORMATION OF FILE (, etc.) FROM OTHER APPLICATIONS (ASCII text, graphics, Lotus 1-2-3, DBase IV) INTO HYPERTEXT IS
1. NOT FACILITATED.
2. FACILITATED.

*** Name: HYPERTEXTABLE

*** HYPERTEXTABILITY OF THE SOFTWARE IS
*** 1. POORELY SUPPORTED.
*** 2. FAIRLY WELL SUPPORTED.
*** 3. WELL SUPPORTED.

Name: SPEED

< The speed of operation in hypertext can be measured in terms of link activation response time which is the length of time between user's pointing on button (link source) and the show-up of linked node (link destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION RESPONSE IS
1. SLOWER THAN 2 SECOND AND FASTER THAN 0.25 SECOND (can be categorized into poor
response time).
2. FASTER THAN 2 SECOND AND SLOWER THAN 1 SECOND (can be categorized into fair response time).
3. FASTER THAN 1 SECOND OR SLOWER THAN 0.25 SECOND (can be categorized into good response time).

Name: PRINTING

PRINTING HYPERTEXT (the concatenated nodes) IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: HELP FEATURES

ONLINE HELP FEATURES ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: RESPONSE

VISIBLE OR AUDIBLE RESPONSE TO USER’S ALL ACTION WITH THE SYSTEM IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: ERROR MESSAGE

HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER’S ERRORS ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: MENU SELECTION

MENU SELECTION PROCEDURE IS
1. NOT PROVIDED.
2. PROVIDED WITHOUT DEFAULT OPTION.
3. PROVIDED WITH DEFAULT OPTION.

Name: INPUT METHOD

INPUT METHOD FOR DIRECT MANIPULATION (e.g., A single-point-and-click by mouse or arrow keys) IS
1. NOT PROVIDED.
2. PROVIDED.

Name: COLOR DISPLAY

COLOR DISPLAY IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: TUTORIAL

TUTORIAL INTRODUCTION OR DEMONSTRATION IS
1. NOT FACILITATED.
2. FACILITATED.

* Name: USER-FRIENDLY
*   
* USER-FRIENDLY FEATURES ARE
*   1. POORLY SUPPORTED.
*   2. FAIRLY WELL SUPPORTED.
*   3. WELL SUPPORTED.

Name: PROGRESSIVE

< The apparent complexity of the system can be reduced by progressive disclosure which is presenting the user with only the information relevant to him at any point in time. Menu interfaces and icons (where data is represented in a compact format, and details are only presented upon explicit request) are methods of progressive disclosure.>

PROGRESSIVE DISCLOSURE IS
1. POORLY FACILITATED.
2. FAIRLY WELL FACILITATED.
3. WELL FACILITATED.

Name: METAPHORS

< A metaphor provides a simple explanation of the system which can be used to predict the system's behavior. It suggests that the computer is like something with which the user is already familiar. Icons displayed on the screen, as the metaphor, may represent a number of documents, file folders, and software programs.>
ANY READILY APPARENT METAPHORS ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: "LOOK and FEEL"

"Look" refers to the appearance of the screen displays (i.e. the visual layout of words of pictorial features of the screen). "Feel" refers to how the program interacts with the user when performing its functions (i.e. the sequence of functions that occur when the user selects an option made available on the screen).

"LOOK AND FEEL" OF THE COMMAND STRUCTURE IS
1. CLUMSY.
2. FAIR.
3. SMOOTHLY FUNCTIONING.

* Name: COMPLEXITY

* THE COMPLEXITY OF USE
* 1. VERY COMPLEX.
* 2. COMPLEX.
* 3. NOT COMPLEX.

** Name: EASE

** LEARNING AND USE ARE
** 1. NOT EASY.
** 2. EASY.
** 3. VERY EASY.

Name: CONSULTANCY

SUPPORTING SERVICE THROUGH HOT LINE (e.g., toll free phone number, 24 hours phone service including weekend, online bulletin board system) IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: POLICY

POLICIES ALERTING USERS TO CHANGES AND ENHANCEMENTS ARE
1. NOT AVAILABLE.
2. AVAILABLE.
**Name: WARRANTY**

**WARRANTY PERIOD IS**
1. LESS THAN 1 YEAR.
2. EQUAL TO OR GREATER THAN 1 YEAR.

* **Name: SUPPORT**
* **VENDOR SUPPORT IS**
  * 1. NOT ENOUGH.
  * 2. ENOUGH.

**Name: AGE**

**AGE OF THE SOFTWARE ON THE MARKET IS**
1. LESS THAN 1 YEAR.
2. EQUAL TO OR GREATER THAN 1 YEAR.
3. NOT KNOWN.

**Name: USER GROUPS**

**USER GROUPS ARE**
1. NOT ACTIVE.
2. ACTIVE.
3. NOT KNOWN.

**Name: LITERATURE**

**LITERATURE (other than advertising or product announcing) ABOUT THE SOFTWARE ARE**
1. NOT PUBLISHED.
2. PUBLISHED.
3. NOT KNOWN.

* **Name: MATURITY**
* **SOFTWARE MATURITY**
  * 1. NOT MATURED.
  * 2. MATURED.

**Name: SEARCH**

**SEARCH OF CHARACTER STRING IS**
1. NOT AVAILABLE.
2. AVAILABLE.
Name: DOCUMENTATION

DOCUMENTATION IS
1. POORLY ORGANIZED AND INCONVENIENT TO USE.
2. FAIRLY WELL ORGANIZED AND CONVENIENT TO USE.
3. WELL ORGANIZED AND VERY CONVENIENT TO USE.

*** Name: GENERIC

*** < Characteristics generic to most of software include ease of learning and use, search of character string, "look and feel" of the command structure, manual and documentation, vendor support, and software maturity.>

*** GENERIC SOFTWARE CHARACTERISTICS ARE
*** 1. POORLY SUPPORTED.
*** 2. FAIRLY WELL SUPPORTED.
*** 3. WELL SUPPORTED.

Name: EXTRA FUNCTION

EXTRA FUNCTIONALITY (expert system, database management system, spreadsheet system) IS
1. NOT ENOUGH.
2. ENOUGH.
CHOICES

1. This hypertext software is extremely poor for you to do your primary hypertext task (browsing or authoring).

2. This hypertext software is quite poor for you to do your primary hypertext task (browsing or authoring).

3. This hypertext software is poor for you to do your primary hypertext task (browsing or authoring).

4. This hypertext software is fair for you to do your primary hypertext task (browsing or authoring).

5. This hypertext software is good for you to do your primary hypertext task (browsing or authoring).

6. This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).

7. This hypertext software is extremely excellent for you to do your primary hypertext task (authoring or browsing).

------------------

Uses all applicable rules in data derivations.

Probability System: 0 - 10

Display Threshold: 0
APPENDIX F

SCRIPT FOR THIRTY MINUTES’ TRAINING SESSION
In thirty minutes, subjects assigned to the Treatment Group 2 and the Treatment Group 4 are trained with the script about evaluation criteria of hypertext software, preliminary considerations, qualifiers, and choices.

**EVALUATION CRITERIA OF HYPERTEXT SOFTWARE**

1. Hypertextability
   1.1. Dynamics of Linking
      1.1.1. Number and Type of Links Supported
         1.1.1.1. Text Editor Linking
         1.1.1.2. Linking by Matrix Outlining
         1.1.1.3. Key Word Linking
         1.1.1.4. File Linking
         1.1.1.5. Hierarchical Links
         1.1.1.6. Referential Links
         1.1.1.7. Cluster Linking
      1.1.2. The Intuitiveness of Linking Procedure
   1.2. User's Manipulations of Nodes.
      1.2.1. Capability of Editing Nodes (e.g., moving, copying, deleting, inserting)
      1.2.2. Capability of Modifying and Arranging both Nodes and Links Together (e.g., dividing, merging, sorting, etc.)
   1.2.3. Displaying Graphics
   1.2.4. Optional Functionalities for Node Manipulation
      1.2.4.1. Typed Nodes Defining a Set of Data Structure and Valid Operation
      1.2.4.2. Annotation on Nodes
      1.2.4.3. Multiple Users' Access for Concurrent Authoring
      1.2.4.4. Versioning
      1.2.4.5. Creating Graphics
      1.2.4.6. Sound
      1.2.4.7. Animation
   1.3. Traversing
      1.3.1. Backward Traversing
      1.3.2. Browsing Map
      1.3.3. Multiple Projection Windowing
   1.4. File Transformation from Other Applications (e.g., ASCII text, Word Perfect 5.1., Lotus 1-2-3, dBASE IV, graphics) into Hypertext
2. The Speed of Operation in Link Activation
3. Printing Hypertext (the Concatenated Nodes)
4. Characteristics Generic to Most Software
   4.1. Ease of Learning and Use
      4.1.1. User-Friendly Features
4.1.1.1. Help Menu
4.1.1.2. Visible or Audible Response to User's All Action with the System
4.1.1.3. Error Messages to a User's Error
4.1.1.4. Menu Selection Procedure
4.1.1.5. Direct Manipulation
4.1.1.6. Color Display
4.1.1.7. Tutorial Introduction or Demonstration

4.1.2. The Complexity of Use
4.1.2.1. Progressive Disclosure
4.1.2.2. The Availability of Any Readily Apparent Metaphors
4.1.2.3. The "Look and Feel" of the Command Structure.

4.2. Search of Character String
4.3. Documentation
4.4. Vendor Supports
4.4.1. Support Services through Hot Line (e.g., Toll Free Phone Number, 24 Hours and 7 Days Phone Service, Online Electronic Bulletin Board System)
4.4.2. Policies Alerting Users to Changes and Enhancements
4.4.3. Warranty

4.5. Software Maturity
4.5.1. Age of Software on the Market
4.5.2. Existence of Active User Group
4.5.3. Literature (Other Than Advertising or Product Announcing) about the Software

5. Extra Functionality

Differences in Tasks and Users: Preliminary Considerations in Evaluating Software

1. Tasks

A major preliminary consideration in evaluating software is the task which you plan to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.

Two general functions are usually associated with hypertext environments: (1) the ability to
retrieve information in a large information space, and (2) the ability to create non-sequential text (Knuth 1990).

Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind (Halasz 1988).

People having different tasks will use hypertext systems in different ways, so different hypertext mechanisms are needed to support different tasks (Nielsen 1989).

Task.1: Authoring

Authoring centres on the information creating and linking aspects. That is, hypertext systems can provide a means to support the construction of both individual and community information bases.

Systems designed primarily for authoring are generally characterized by the well-developed tools for "creation and modification of the networks (Halasz 1988).

Intermedia was described as an environment "to link information together, create paths through a body of related material, annotate existing texts, and create notes that direct readers to either bibliographic data or the body of the reference text (Yankelovich 1988)."

Task.2: Browsing

Browsing through traversing or graphical browsers is one of prevalent information retrieval in hypertext.

The ability to explore large bodies of information easily and naturally is the important issue to these experts. ... several mechanisms (e.g. graphical browsers, sophisticated search facilities, etc.) which are intended to support these exploratory tasks (Knuth 1990).

Systems designed primarily for browsing are generally characterized by relatively well-developed tools for "information
presentation* and "exploratory browsing" (Halasz 1988).

2. Users

Individual difference is one of the major preliminary considerations in evaluating software. It is hard to expect that one software satisfies everybody.

Thus, it is necessary to identify yourself. What is best for one group of users may not be best for another, since different people will perform very differently (Nielsen 1990).

User.1: Not Familiar with Any Hypertext Software.

User.2: Familiar with at Least One Hypertext Software.

3. Different Evaluation Results

Depending on types of users and tasks, there should be different results in evaluating software.

Because of these two observations (differences in both users and tasks), there is little hope for a single, universal hypertext user interface design which will be optimal to everybody (Nielsen 1990).
QUALIFIERS

Qualifiers represent the evaluation criteria of hypertext engineering in the expert system, HEES. The printout of the qualifiers for the HEES provides guidelines and an input data collection sheet for the subjects involved in the experiment with KnowledgePro and MaxThink's. In the training sessions, all the subjects, except the control group who will not use the HEES, will receive both the printout and the oral explanation within the designated time period. Entry level qualifiers will be answered by the subjects, choosing one of the given values. Upper level qualifiers, whose names are preceded by one or more asterisks (*) in the printout, will be answered by the HEES using the backward chaining method.

Name: USER

< Individual difference is one of the major preliminary considerations in evaluating software. What is best for one group of users may not be best for another since different people perform very differently. There is little hope for a single, universal hypertext software which will be optimal to everybody.

YOU ARE
1. NOT FAMILIAR WITH ANY HYPERTEXT SOFTWARE.
2. FAMILIAR WITH AT LEAST ONE HYPERTEXT SOFTWARE.

Name: TASK

< Another major preliminary consideration in evaluation software is task which you would like to do with the software. Many hypermedia systems have been designed to support a specific task. Even generic systems are usually designed with a target task domain in mind.

YOUR INTENDED PRIMARY TASK IS
1. BROWSING using hypertext software as a tool for information presentation (e.g., tour guide, museum exhibition, etc.) or accessing large, complex information sources in various ways depending on various viewpoints.
2. AUTHORING using hypertext software as a tool for building full text database or an idea organizer.
Name: BACKWARD

< Backward traversing is the navigating to go back along the path they took to reach the current screen in the hypertext.>

BACKWARD TRAVERSING IS
1. NOT SUPPORTED.
2. SUPPORTED.

Name: BROWSING MAP

< Browsing map is the graphical presentation of current location within the system and the relationship to the rest of the system by links.>

BROWSING MAP IS
1. NOT AUTOMATICALLY GENERATED BY THE SYSTEM.
2. AUTOMATICALLY GENERATED BY THE SYSTEM.

Name: WINDOW

< 'Often a hypermedia user will want to see the contents of a new node before leaving the node they are currently viewing. The most common approach to this issue is to present each node in a window whose size is something less than that of the entire screen. [Bego90]' Multiple projection (or overlapping) windowing allows a number of windows which contain a node respectively to be open on the screen at the same time.>

MULTIPLE PROJECTION WINDOWING IS
1. NOT PROVIDED.
2. PROVIDED.

* Name: TRAVERSING
* 
* TRAVERSING IS
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.

Name: TYPED NODES

TYPED NODES (defining a set of data structure and valid operation) IS
1. NOT AVAILABLE.
2. AVAILABLE.
Name: ANNOTATION

ANNOTATION ON NODES IS:
1. NOT AVAILABLE.
2. AVAILABLE.

Name: MULTIPLE USER

MULTIPLE USERS’ ACCESS (for concurrent authoring) IS:
1. NOT AVAILABLE.
2. AVAILABLE.

Name: VERSIONING

VERSIONING (maintaining a history of changes of a node or set of nodes) IS:
1. NOT AVAILABLE.
2. AVAILABLE.

Name: CREATE_GRAPHIC

CAPABILITY OF CREATING GRAPHICS IS
1. NOT PROVIDED.
2. PROVIDED.

Name: SOUND

CAPABILITY OF SOUND IS
1. NOT PROVIDED.
2. PROVIDED.

Name: ANIMATION

CAPABILITY OF ANIMATION IS
1. NOT PROVIDED.
2. PROVIDED.

* Name: OPTIONAL FUNCTIONS

* OPTIONAL FUNCTIONALITIES FOR EDITING NODES ARE
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.
Name: EDITING NODES

CAPABILITY OF EDITING NODES (e.g., moving, copying, deleting, inserting, etc.) IS
1. NOT ENOUGH.
2. ENOUGH.

Name: MODIFYING

CAPABILITY OF MODIFYING (e.g., dividing, merging, sorting, etc.) IS
1. NOT ENOUGH.
2. ENOUGH.

Name: DISPLAY GRAPHIC

CAPABILITY OF DISPLAYING GRAPHICS IS
1. NOT PROVIDED.
2. PROVIDED.

** Name: NODES

** USER'S MANIPULATION OF NODES IS
** 1. POORLY SUPPORTED.
** 2. FAIRLY WELL SUPPORTED.
** 3. WELL SUPPORTED.

Name: EDITOR LINKING

TEXT EDITOR LINKING (linking topics by typing text using text editor) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: MATRIX OUTLINING

< In a hierarchical organization, the user may display all possible parent (input) or child (output) topics to a specific topic on windows by touching function keys. Then the user may mark on the necessary topics to be linked to the current topic as a parent or child node. This, matrix outlining, is a one of the powerful way of linking.>

MATRIX OUTLINING IS
1. NOT FACILITATED.
2. FACILITATED.
Name: KEY WORD LINKING

< The user may list all the unique words in a file by touching a function key and mark on the necessary topics. Then, the system automatically links the key words to all the topics which contain the key words. This, key word linking, is also one of the powerful way of linking.>

KEY WORD LINKING IS
1. NOT FACILITATED.
2. FACILITATED.

Name: FILE LINKING

FILE LINKING (linking a topic to a file or linking between files) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: HIERARCHY LINKS

HIERARCHICAL LINKING (implementing hierarchical tree linkages within the hypertext having parent, child, or sibling nodes) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: REFERENTIAL

REFERENTIAL LINKING (linking points or regions of text without hierarchical relationship [Horn89]) IS
1. NOT FACILITATED.
2. FACILITATED.

Name: CLUSTER LINKING

CLUSTER LINKING (linking a group of shorter pieces of information, called path or web) IS
1. NOT FACILITATED.
2. FACILITATED.

* Name: LINK TYPES
* 
* NUMBER AND TYPE OF LINKS SUPPORTED ARE
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.
Name: INTUITIVENESS

THE INTUITIVENESS OF LINKING PROCEDURE IS
1. NOT INTUITIVE.
2. INTUITIVE.
3. VERY INTUITIVE.

** Name: DYNAMICS

** DYNAMICS OF LINKING IS
** 1. POORLY SUPPORTED.
** 2. FAIRLY WELL SUPPORTED.
** 3. WELL SUPPORTED.

Name: TRANSFORMABILITY

TRANSFORMATION OF FILE (, etc.) FROM OTHER
APPLICATIONS (ASCII text, graphics, Lotus 1-2-3, 
DBase IV) INTO HYPERTEXT IS
1. NOT FACILITATED.
2. FACILITATED.

*** Name: HYPERTEXTABLE

*** HYPERTEXTABILITY OF THE SOFTWARE IS
*** 1. POORLY SUPPORTED.
*** 2. FAIRLY WELL SUPPORTED.
*** 3. WELL SUPPORTED.

Name: SPEED

< The speed of operation in hypertext can be 
measured in terms of link activation response 
time which is the length of time between 
user’s pointing on button (link source) and 
the show-up of linked node (link 
destination).>

THE SPEED OF OPERATION IN LINK ACTIVATION 
RESPONSE IS
1. SLOWER THAN 2 SECOND AND FASTER THAN 0.25 
SECOND (can be categorized into poor 
response time).
2. FASTER THAN 2 SECOND AND SLOWER THAN 1 
SECOND (can be categorized into fair 
response time).
3. FASTER THAN 1 SECOND OR SLOWER THAN 0.25
SECOND (can be categorized into good response time).

Name: PRINTING

PRINTING HYPERTEXT (the concatenated nodes) IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: HELP FEATURES

ONLINE HELP FEATURES ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: RESPONSE

VISIBLE OR AUDIBLE RESPONSE TO USER'S ALL ACTION WITH THE SYSTEM IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: ERROR MESSAGE

HELPFUL AND EASY-TO-UNDERSTAND MESSAGES TO USER'S ERRORS ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: MENU SELECTION

MENU SELECTION PROCEDURE IS
1. NOT PROVIDED.
2. PROVIDED WITHOUT DEFAULT OPTION.
3. PROVIDED WITH DEFAULT OPTION.

Name: INPUT METHOD

INPUT METHOD FOR DIRECT MANIPULATION (e.g., A single-point-and-click by mouse or arrow keys) IS
1. NOT PROVIDED.
2. PROVIDED.
Name: COLOR DISPLAY

COLOR DISPLAY IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: TUTORIAL

TUTORIAL INTRODUCTION OR DEMONSTRATION IS
1. NOT FACILITATED.
2. FACILITATED.

* Name: USER-FRIENDLY
* USER-FRIENDLY FEATURES ARE
* 1. POORLY SUPPORTED.
* 2. FAIRLY WELL SUPPORTED.
* 3. WELL SUPPORTED.

Name: PROGRESSIVE

< The apparent complexity of the system can be reduced by progressive disclosure which is presenting the user with only the information relevant to him at any point in time. Menu interfaces and icons (where data is represented in a compact format, and details are only presented upon explicit request) are methods of progressive disclosure.>

PROGRESSIVE DISCLOSURE IS
1. POORLY FACILITATED.
2. FAIRLY WELL FACILITATED.
3. WELL FACILITATED.

Name: METAPHORS

< A metaphor provides a simple explanation of the system which can be used to predict the system’s behavior. It suggests that the computer is like something with which the user is already familiar. Icons displayed on the screen, as the metaphor, may represent a number of documents, file folders, and software programs.>

ANY READILY APPARENT METAPHORS ARE
1. NOT AVAILABLE.
2. AVAILABLE.
Name: "LOOK and FEEL"

"Look" refers to the appearance of the screen displays (i.e. the visual layout of words of pictorial features of the screen). "Feel" refers to how the program interacts with the user when performing its functions (i.e. the sequence of functions that occur when the user selects an option made available on the screen).>

"LOOK AND FEEL" OF THE COMMAND STRUCTURE IS
1. CLUMSY.
2. FAIR.
3. SMOOTHLY FUNCTIONING.

* Name: COMPLEXITY

* THE COMPLEXITY OF USE
* 1. VERY COMPLEX.
* 2. COMPLEX.
* 3. NOT COMPLEX.

** Name: EASE
**
** LEARNING AND USE ARE
** 1. NOT EASY.
** 2. EASY.
** 3. VERY EASY.

Name: CONSULTANCY

SUPPORTING SERVICE THROUGH HOT LINE (e.g., toll free phone number, 24 hours phone service including weekend, online bulletin board system) IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: POLICY

POLICIES ALERTING USERS TO CHANGES AND ENHANCEMENTS ARE
1. NOT AVAILABLE.
2. AVAILABLE.

Name: WARRANTY

WARRANTY PERIOD IS
1. LESS THAN 1 YEAR.
2. EQUAL TO OR GREATER THAN 1 YEAR.

* Name: SUPPORT

* VENDOR SUPPORT IS
  * 1. NOT ENOUGH.
  * 2. ENOUGH.

Name: AGE

AGE OF THE SOFTWARE ON THE MARKET IS
1. LESS THAN 1 YEAR.
2. EQUAL TO OR GREATER THAN 1 YEAR.
3. NOT KNOWN.

Name: USER GROUPS

USER GROUPS ARE
1. NOT ACTIVE.
2. ACTIVE.
3. NOT KNOWN.

Name: LITERATURE

LITERATURE (other than advertising or product announcing) ABOUT THE SOFTWARE ARE
1. NOT PUBLISHED.
2. PUBLISHED.
3. NOT KNOWN.

* Name: MATURITY

* SOFTWARE MATURITY
  * 1. NOT MATURED.
  * 2. MATURED.

Name: SEARCH

SEARCH OF CHARACTER STRING IS
1. NOT AVAILABLE.
2. AVAILABLE.

Name: DOCUMENTATION
DOCUMENTATION IS
1. POORLY ORGANIZED AND INCONVENIENT TO USE.
2. FAIRLY WELL ORGANIZED AND CONVENIENT TO USE.
3. WELL ORGANIZED AND VERY CONVENIENT TO USE.

*** Name: GENERIC

*** Characteristics generic to most of software include ease of learning and use, search of character string, "look and feel" of the command structure, manual and documentation, vendor support, and software maturity.

*** GENERIC SOFTWARE CHARACTERISTICS ARE
*** 1. POORLY SUPPORTED.
*** 2. FAIRLY WELL SUPPORTED.
*** 3. WELL SUPPORTED.

Name: EXTRA FUNCTION

EXTRA FUNCTIONALITY (expert system, database management system, spreadsheet system) IS
1. NOT ENOUGH.
2. ENOUGH.
CHOICES

1. This hypertext software is extremely poor for you to do your primary hypertext task (browsing or authoring).

2. This hypertext software is quite poor for you to do your primary hypertext task (browsing or authoring).

3. This hypertext software is poor for you to do your primary hypertext task (browsing or authoring).

4. This hypertext software is fair for you to do your primary hypertext task (browsing or authoring).

5. This hypertext software is good for you to do your primary hypertext task (browsing or authoring).

6. This hypertext software is excellent for you to do your primary hypertext task (browsing or authoring).

7. This hypertext software is extremely excellent for you to do your primary hypertext task (authoring or browsing).

---------

Uses all applicable rules in data derivations.

Probability System: 0 - 10

Display Threshold: 0
BIBLIOGRAPHY


---

Less than Full-Text Indexing Using a Non-Boolean Searching Model.

DeRose, Steven J. "Expanding the Notion of Links," Hypertext '89 Proceedings, November 1989. 249-257.


---


Myers, Judy. Presentation to Library and Information Technology Association (LITA) Hypertext Interest Group (IG) - ALA Annual 1990.


