Order Number 1355473

Computer knowledge representation of users of command language-based interfaces and graphical user interfaces

Atlas, Robert Scott, M.A.

Rice University, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

RICE UNIVERSITY

COMPUTER KNOWLEDGE REPRESENTATION OF USERS OF COMMAND LANGUAGE-BASED INTERFACES AND GRAPHICAL USER INTERFACES

by

ROBERT S. ATLAS

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
MASTER OF ARTS

APPROVED, THESIS COMMITTEE:

Nancy J. Cooke, Chairman
Assistant Professor of Psychology

David M. Lane
Associate Professor of Psychology

Randi C. Martin
Associate Professor of Psychology

Houston, Texas
May, 1993

Abstract

Computer Knowledge Representation of Users of Command Language-Based Interfaces and Graphical User Interfaces

by

Robert S. Atlas

An exploratory study was conducted with users of a command-based system, MS- DOS, and the Graphical User Interfaces (GUI) of the Apple Macintosh. It was hypothesized that command-based interfaces and GUIs differ in the ease with which they afford attainment of particular concepts, and that in general the concepts investigated would be more readily attained by Macintosh users. The study attempted to assay subjects' knowledge of particular computer-related concepts, their ability to perform related operations, and the organization of that knowledge. Contrary to some theory (e.g., Tennyson and Cocchiarelli, 1986), a double dissociation between verbalizable conceptual knowledge and performance was observed. Results of the study suggested differences in the support provided by DOS and the Macintosh interface for development of knowledge of particular concepts and procedures, underscoring the potential value of understanding the detailed effects of particular interfaces, and classes of interface, upon user knowledge.

Acknowledgements

I would like to gratefully acknowledge all who have assisted me, directly or indirectly. Special thanks are due to my advisor, Dr. Nancy J. Cooke for her able and patient assistance , and also to Dr. David Lane, for encouraging my interest in psychology and for invaluable assistance with this and other efforts. The advice and assistance of Dr. H. Albert Napier made this project possible. Without the encouragement and support provided by my family, particularly my parents Dr. Leon T. Atlas and Maxine McCall Atlas, I would not have been in the position to pursue this project. Thank you one and all.

Table of Contents

List of Tables

Computer Knowledge Representation of Users of Command Language-Based Interfaces and Graphical User Interfaces

The practical goal of the study of Human-Computer Interaction (HCI) is the design of computer-related systems which better fulfill the traditional aims of Human Factors, such as ease of use. Yet it must be admitted that the field of HCI has had only limited success in understanding the determinants of complexity and ease of use in the interface.

While there are numerous suggested approaches to studying HCI, and numerous proposed models to account for HCI phenomena, there is nothing like a systematic body of theory with good empirical support in the field. This is unsurprising in light of the difficulty in understanding the generally simpler, more controlled phenomena brought into the psychological laboratory.

Clearly, one determinant of complexity for a user must be the user's knowledge of concepts and procedures relevant to accomplishing tasks with the system. While knowledge helps determine the subjective complexity of the system, more or less conversely, the information presented by the system, and the way it is presented, specifically by the user interface, must affect the user's learning processes and knowledge. The problem of central interest in the present line of research is, very broadly, that of characterizing the manners in which the user interface affects the user's knowledge, and of understanding the implications of such effects for usability of the system. A further problem is that of how the user interface affects the representation and organization of the user's knowledge.

In this context knowledge representation refers to the distinction between types of knowledge, such as declarative, procedural, (e.g. Anderson, 1982) or conceptual (e.g. Tennyson and Cocchiarella, 1986), rather than coding, such as imaginal or propositional. Knowledge organization refers to more or less systematic access to related information. Schemata (Bartlett, 1932), scripts (Schank and Abelson, 1977), and mental models (e.g. Gentner and Stevens, 1983), are examples from the psychological literature of constructs providing, and perhaps describing, knowledge organization.

It must be noted that learning can be affected by the existence of prior, putatively related knowledge. Fully accomplishing such goals as promoting effective interface design, user training, and transfer between systems in the future may therefore require a thorough understanding of the effects of the interface upon the way people with varying prior knowledge, or expertise, use and conceive, or misconceive, of computer systems.

Norman (1987) suggested that the study of human-machine interaction requires all of psychology and more. As Norman also pointed out, the study of applications reveals the deficiencies of, and may add to, science. Interfaces provide an arena for study of some of the traditional concerns of cognitive psychology, including the nature of the processes involved in learning and transfer of training in general, and also including: human problem solving; the nature, formation and transformation, and influence of mental models; the distinctions between conceptual, declarative and procedural knowledge; and the nature of expert-novice differences. All of these concerns are relevant to the present research program.

This research attempted to characterize selected aspects of the knowledge and knowledge organization of users with experience with one or both of two interface types, command based or graphical user interface (GUI), instantiated by DOS (command-based) and the more graphical Macintosh operating system. Before describing the particulars of the present research, it is appropriate to examine some of the methods and theoretical constructs or frameworks available in psychology and in HCI that are related to knowledge representation and organization.

Knowledge Representation

Knowledge is often categorized as either declarative (knowledge about facts and things) or procedural (knowledge how). This relation, and often enough the dissociation, between verbalizable, declarative, knowledge and skilled task performance, which may be based upon procedural knowledge, is a sizable area of study in itself. Anderson (1982) suggested that the reduction in performance time with practice is due to three main causes corresponding to stages of learning. The first stage is a declarative stage of understanding and eliminating errors, the second is compilation of procedural chunks, and the third is automatization of the procedures in each chunk. Compiled knowledge is assumed to be inaccessible to verbalization. Rumelhart and Norman (1981), on the other hand, suggest that much knowledge is procedural, and that procedural knowledge can be used to produce "knowledge that". Further, they grant that the two types of knowledge interact in learning.

Tennyson and Cocchiarella (1986) described a third form of knowledge, conceptual knowledge. According to Tennyson and Cocchiarella (1986) conceptual knowledge is more complex than declarative knowledge, comprising not just declarative knowledge but also understanding of a concept's "operational structure within itself and between associated concepts" (p.41). Otherwise their theory somewhat resembles that of Anderson (1982): learning is conceived of as a two-stage process with conceptual knowledge learned first, and procedural knowledge developing as conceptual knowledge is applied to solve problems. However, more like Rumelhart and Norman (1981), and consistent with the findings of Sanderson (1989) and of Peterson et al (1990), Tennyson and Cocchiarella acknowledge that conceptual knowledge can improve with use of procedural knowledge.

Perhaps conceptual knowledge in Tennyson's sense should be considered to be a subset of declarative knowledge in Anderson's sense, that subset of knowledge involving more abstract properties and relations. In the discussion that follows this distinction will be emphasized by dividing declarative knowledge into conceptual knowledge and knowledge that will be termed "simply descriptive". Simply descriptive knowledge will refer to knowledge of concrete, observable entities and sequences of events without reference to more abstract constructs and their properties and relations. Simply descriptive knowledge can encompass if-then (causal or correlational) relations based upon observation.

Dissociations have often been demonstrated between performance and verbalizable knowledge about a task. For example, Broadbent, Fitzgerald, and Broadbent (1986) studied the relation between verbalizable knowledge and performance in controlling two artificial systems, one concerned with transportation and one with a model of the economy. In several cases, subjects' answers to verbal questions changed without appropriate changes in performance. On this basis Broadbent et al. questioned the idea that people's behaviors are all the result of consulting and manipulating a single model of the world. Rather, they suggested, there are two extreme decision strategies, each with advantages in particular situations. In this view, one mode of performance would be based on verbalizable knowledge, while the other selects action by matching the situation to earlier experience, a "look-up table". But a system using a look-up table cannot give reasons for the actions it chooses.

Both extremes, good performance without accurate verbalizable knowledge (e.g. Broadbent, 1977, Lewicki, 1986), and knowledge but poor performance as above have often been observed. Sanderson (1989) found that, in general, verbalizable knowledge, as measured by questionnaire, improved with practice in a task involving trial-and-error solution of equations of an unknown form. With a moderate amount of practice at finding precise solutions, an association between performance and verbalizable knowledge emerged. Somewhat similarly, Peterson, Ridenour, and Somers (1990) found that students given a procedural approach to using a ruler were apparently able to use their procedural knowledge to infer conceptual knowledge of fractions. Peterson et al. concluded that their subjects were producing conceptual knowledge in the sense described by Tennyson and Cocchiarella (1986), an understanding of the nature of concepts and the relations among them. A complete theory of knowledge representation should accommodate such results.

One possible framework for describing different forms of knowledge and performance based upon them is the hierarchical model developed by Rasmussen (e.g., Rasmussen, 1983; Rasmussen, 1985; Rasmussen and Vicente, 1989). Rasmussen's hierarchy begins with *skill-based behavior*, sensory-motor performance in highly integrated behavior patterns that occur largely without conscious intervention. From a repertoire of automated, skill-based routines, new special purpose routines can be composed.

The next of Rasmussen's levels is *rule-based behavior*: Composition of subroutines is controlled by stored rules, selected from previous successful experiences, acquired by communication with others, or prepared by conscious problem-solving. According to Rasmussen, unlike the basis of skill-based performance, the rules used in rule-based performance can be reported.

The final level of the hierarchy, invoked in unfamiliar situations for which no rules are available, is goal-controlled, *knowledge-based* performance. At this level, plans are selected and tested, either by trial and error or by predictions based on conceptual understanding of the environment and the system with which the user wishes to interact. Structure of the system is represented by a "mental model".

Rasmussen's hierarchy seems, with some interpretation, able to accommodate the various forms of knowledge previously discussed. Skill-based behavior may correspond to some admixture of very fundamental perceptual-motor routines and the sort of procedural knowledge described by Anderson (1982) or, similarly, Tennyson and Cocchiarella (1986), and the sort of look-up table postulated by Broadbent et al (1986).

Presumably skills may be developed in the course of repeated execution of the rules that form the second stage of Rasmussen's hierarchy. Such rules may be formed either through conceptual knowledge-based inference or on the basis of what is being termed simply descriptive knowledge from a number of sources. According to Rasmussen rules can be reported. Rasmussen has proposed that rules select and compose subroutines, that is skills-based behaviors, an idea not in any conflict with the idea that new subroutines become compiled with repeated application of declarative or conceptual knowledge, and may become inaccessible to verbalization.

Rasmussen's final level, knowledge-based performance, would seem to correspond to use of either what is being termed conceptual knowledge or, more specifically, to "mental models", discussed later. Verbalizable knowledge in the sense used in the studies of dissociations between verbalizable knowledge and performance, such as Broadbent, et al. (1986), can, in Rasmussen's hierarchy, take the form of rules or other "higher-level" knowledge. But the skills described by Rasmussen must also be considered a form of knowledge.

Carroll and Olson (1987, p.32), in the course of a discussion of mental models in human-computer interaction, identified as important research needs "Determining how people intermix different representations in producing behavior." and "Explore how knowledge about systems is acquired." The present research program in part attempted to examine the effects of particular interface types upon the extent to which aspects of users' performance are based upon procedural, simply descriptive, or conceptual knowledge. The general thesis is that interface differences may make particular system concepts more salient or more necessary for successful use of the system, making mastery of those concepts more or less probable.

From the foregoing discussion it is hopefully apparent that differing forms of knowledge can coexist and interact. There is evidence that procedural knowledge is compiled in the course of execution of plans based upon declarative knowledge, but there is also evidence that knowledge of procedures can help induce conceptual knowledge (Peterson, Ridenour, and Somers ,1990). However, learning the sort of procedures examined in this study may in fact impart what is here described as simply descriptive knowledge, resulting in rules of the sort described by Rasmussen, and not yet the sort of compiled procedural knowledge described by Anderson (1982) or Tennyson and Cocchiarella (1986). In this context a look-up table might be described as resulting in implicit rules. The most important distinction for the present purposes is whether or not knowledge of procedures, compiled or not, is based upon or associated with conceptual knowledge. Conversely, it may be of interest to gauge the presence of conceptual knowledge that cannot be effectively used procedurally. Certainly this can be of importance in assessing ease of use for a system.

There may be other, more subtle distinctions to be drawn, particularly with respect to the organization of declarative and procedural knowledge, and with respect to the processes leading to knowledge acquisition. The succeeding sections will describe a number of methods of characterizing, modeling, or studying knowledge acquisition and organization, and some specific applications to the study of HCI.

Theoretical Approaches to User Knowledge and Knowledge Organization

GOMS and Production System-based Models.

Probably the most widely adopted modeling method in HCI is the original GOMS model of Card, Moran, and Newell (1980, 1983). GOMS is an acronym for Goals, Operators, Methods, and Selection rules. Operators are the actions available in using a system. Methods are combinations of operators, used to accomplish goals, and Selection rules specify how to choose an appropriate method to accomplish a particular goal. A GOMS analysis involves decomposing the method for accomplishing a task into a series of submethods to accomplish subgoals, which may be further decomposed as far as the mental operations (e.g., retrieving a plan from long term memory) and physical actions (e.g., using an input device to move a cursor) that the analyst chooses to treat as primitives. GOMS was developed in a pragmatic spirit to allow engineering calculations, and suffers from a number of limitations which restrict its usefulness for the purpose of modeling user cognition. To paraphrase some of the most relevant of the shortcomings from several sources (Carroll & Campbell, 1986; Karat, 1988; Wilson, Barnard, Green, & Maclean, 1988; Olson and Olson, 1990; Howes & Payne, 1990; Ziegler, Vossen, & Hoppe, 1990):

1. GOMS applies to skilled users, not beginners or intermediates, who move between problem solving and skilled behavior.
2. The model does not account for learning or recall after a period of disuse.
3. The model does not account for errors.
4. The model treats elementary perceptual and motor processes relatively explicitly, but tends to differentiate less among cognitive processes.
5. The model was developed for tasks which could be treated as serial, but component processes and tasks can occur in parallel.
6. The model does not take mental workload into account.
7. The model does not account for individual differences among users.
8. GOMS models predict that skill is associated with complete knowledge of the steps in a task. Therefore GOMS cannot account for recognition-based skills, that is users' learning to find and interpret display information to help accomplish tasks.

This last criticism, and perhaps the first, is particularly relevant to modeling performance with a Graphical User Interface (GUI). Indeed, Mayes, Draper, McGregor, and Oatley (1988) found that experienced users of the MacWrite program for the Apple Macintosh could not recall the names of the menu items that they commonly selected. Perhaps it is not only novices who commonly move between problem solving and what is more typically called skilled behavior. That is, even skilled users may rely upon the ability to make particular classes of inference in the course of routine use of a system, and the types of inference required are likely to vary with the interface. At all events, clearly it is necessary to look beyond the original form of GOMS in order to model all the complexities of human-computer interaction.

A number of researchers have tried to extend the original cognitive engineering approach of the GOMS model. Perhaps the best known of these attempts is the Cognitive Complexity Theory (CCT) of Kieras and Polson. The original CCT has now been embedded in a more comprehensive theory, CE+ (Polson and Lewis, 1990). Examination of CCT and CE+ will also serve to indicate some of the complexities of the subject of HCI in general and of the difficulty in accounting for learning and knowledge representation in particular.

The basis of these theories, CCT and CE+, is formalization of the knowledge represented in a GOMS model as a production system. This formalization was carried out, and investigated by Kieras and Polson and colleagues (Bovair, Kieras, & Polson, 1990; Kieras, 1988; Kieras & Bovair, 1986; Kieras & Polson, 1985; Polson, 1988; Polson & Kieras, 1985; Polson & Lewis, 1990; Polson, Munchner, & Englebeck, 1986).

A production system represents knowledge as a collection of rules of the form:

If (condition) Then (action)

When a match to the condition side of the rule is detected, the rule is said to fire, and the action is executed. Production system models also entail processing mechanisms to interpret rules and, typically, a working memory which may store information about current goals and actions and the state of the external environment. Adopting the production system model allows links with other research and theory on cognitive skills (e.g. Anderson, 1983). A production system analysis also allows quantification of further aspects of the knowledge and processing in a GOMS model, supporting quantitative predictions of training time and transfer of training.

Kieras and Polson's early theorizing divided a user's knowledge into *job situation*, *how-to-do-it*, and *how-it works* knowledge (see e.g., Polson, 1988). Job situation knowledge is knowledge of the tasks that can be performed with the system. How-to-do-it knowledge is knowledge of how to carry out the procedures necessary to complete tasks. How-it-works knowledge is the user's knowledge of how a system functions in terms of its internal structure and processes (Kieras and Bovair, 1984). How-it-works knowledge, also referred to as a "mental model" or, to be more specific, a "device model", can, according to Polson (1988), be based upon analogies and can be incomplete.

It is how-to-do-it knowledge which is the starting point for Kieras and Polson's GOMS extension, Cognitive Complexity Theory (CCT). CCT modeling requires first performing a GOMS analysis of how-to-do-it-knowledge and then writing production rules for the methods and control structures in the GOMS model. Kieras and Polson originally assessed the learning time for each rule by studying learning under very controlled conditions which prevent the execution of incorrect steps. Studying learning time for rules under these restricted conditions led to an estimate of 30 seconds per step, assuming a start-up time of 30 to 60 minutes. Given this estimate for steps, multiplying the time needed per production by the number of productions estimates learning time for methods.

Much of the investigation of CCT has been study of the application of the theory to transfer of training. While transfer of training is not directly relevant to the purposes of the present research, consideration of this subject is important for understanding the evidence regarding CCT. Further, the effect of prior knowledge upon acquisition of new knowledge, perhaps in interaction with interface type, is clearly very relevant to the present purposes of understanding the effects of interface upon knowledge representation and organization.

Kieras and Polson (e.g., 1983, 1985, 1988) extended their theory of learning to a theory of transfer, which is similar to Thorndike's (Thorndike & Woodward, 1901) identical elements theory of transfer. Kieras and Polson assumed that determining the number of shared productions and the number of new productions to be learned allows prediction of the amount of transfer and of time to learn a new task. It is admitted, however, that in many cases the transferred productions must be generalized before application to the new task (Polson, 1986, cited in Schmalhofer, 1987).

Kieras and Polson's studies demonstrated that this approach predicts learning performance with considerable success (Bovair et al, 1990; Kieras, 1988; Kieras & Bovair, 1986; Kieras & Polson, 1985; Polson & Kieras, 1985; Polson, et al, 1986). These studies and others (e.g. Ziegler, Vossen, and Hoppe, 1990) found that rate of learning is more or less linearly related to the number of new elements, here productions, to be learned.

Similar, quantifiable, prediction is possible from other explicit representations of the knowledge needed to perform tasks using a system, for example the Tag Action Grammar (TAG) of Payne and Green (1986, 1989). Admittedly, TAG closely resembles a production system. However, Payne and Green (1989) argued that the number of rules to be learned is less important than whether the features of the rules are already well-known to the user. Payne and Green (1989) found that a system with 28 rules described by well-known categories, with one feature mapped to one action, was learned three times as rapidly as a system with twelve rules requiring attention to a complicated set of features. In other words, according to Payne and Green it is the content and familiarity of rules rather than, or as well as, their number, that determines speed of learning.

As the work of Payne and Green (1989) would suggest, the theory's assumptions regarding the learning time required for rules is one of the major weaknesses of the original form of CCT. Times varying from Kieras and Polson's (1988) estimate of 30 seconds for learning procedural steps have been reported by, for example, Ziegler, Vossen, and Hoppe (1990) and Polson and Lewis (1990). Furthermore, as might be expected, (see e.g., Carroll and Rosson, 1987), according to Polson and Lewis (1990), studies of more natural learning conditions reveal much more variable learning times. The difficulty in accounting for learning times in the CCT framework suggests the need to consider other variables, perhaps related to content as indicated by Payne and Green (1989).

Studies of other phenomena also suggest that it may be dangerous to adopt a simplified representation of learning and transfer based upon number of productions. Negative, as well as positive transfer (e.g., Rosson and Grischkowsky, 1987; Smith, Zirkler, and Mynatt, 1985) must be explained, though Singley and Anderson (1988) described the difficulties encountered by their subjects in transferring between text editors as more likely due to activation of inappropriate productions during performance than to any interference in learning new productions.

Tetzlaff (1987) in an elegant experiment demonstrated limitations of the production element account of transfer. Groups of subjects learned different subsets of the commands of a text editor in different orders. Subjects seemed to form different concepts of the editor depending upon the order in which they learned its commands, using different commands and making different errors. At the end of the experiment, subjects performed a sorting task with the command set. Hierarchical clustering and MDS analysis indicated different groupings of commands for the different subject groups. While the common elements approach to transfer appears to allow prediction in some situations, it is apparent that it does not account for all transfer phenomena.

The implications of Tetzlaff's work extend well beyond transfer situations. By analogy, learning different subsets of an interface or learning different interfaces which require or afford learning commands and concepts in differing orders may well lead to differently organized system knowledge.

The evident incompleteness of CCT has prompted several revisions and extensions of the theory that attempt to account for some of the phenomena described above. Khalifa (1991) has attempted to extend CCT to predict ease of learning based upon number of cognitive skills to be acquired and their relative complexities, estimating complexity on the basis of the way a skill is retrieved from long term memory. This revision may enable CCT to better account for variable learning times for rules and procedures.

One of the originators of CCT, Polson, is also involved in an extension of the theory. Polson and Lewis (1990) created a theoretical framework which they called CE+, incorporating variables other than amount of knowledge required that might be expected to influence learning and performance. CE+ combines the production system representation of procedural knowledge from CCT, the analysis of outcome of actions from EXPL (Lewis, Casner, Schoenberg, and Blake, 1987) supporting causal inferences, initial decision processes from the problem solving literature, and the co-ordination of problem solving and learning from models of cognitive architecture such as SOAR (Laird, Newell, and Rosenbloom, 1987) and ACT* (Anderson, 1983).

The EXPL portion of CE+ was developed by Lewis, Casner, Schoenberg, and Blake (1987). The EXPL model treats the role of feedback in learning of procedures. EXPL does not address decision processes in unfamiliar settings. In order to address this failing, Polson and Lewis (1990) included a problem solving component in CE+, based upon the idea of a problem space (Newell & Simon, 1972), and search methods by which the user explores the space. The problem-solving component of CE+ employs a variant of hill climbing called label following, which selects the option that appears most closely related to the user's goal. Once the outcomes of problem-solving episodes are available as productions, CE+ must determine whether to use existing productions or to undertake further problem-solving. As in SOAR (Laird et al, 1987), problem solving is a response to inapplicability of other knowledge. In summary, then, CE+ incorporates a problem-solving component, a learning component, and an execution component. The problem-solving component selects actions, the effects of which are analyzed by the learning component. The results of the learning component's analysis are retained as CCT rules. The execution component is able to execute the CCT rules and to co-ordinate execution of those rules with the problem-solving component.

The CE+ framework retains much of the pragmatic approach of the original GOMS model, in the form of assumptions about the time required to learn procedures (from CCT), the problem solving activities engaged in by the user, and the representation and application of knowledge by production system rules. While these simplifying assumptions may ultimately limit the applicability of the theory, another limitation, possibly more important for the present purposes, and one which was noted by Polson and Lewis (1990), is failure to address the role of prior knowledge in guessing and inference. It is also rather curious that CE+ has nothing to say about the development and effects of the user's job situation knowledge and device model.

Consideration of CE+ serves to indicate some of the complexities of learning and transfer, and the attempt to account for these complexities is undoubtedly a worthwhile extension of CCT. CCT itself has found some empirical support (e. g. Bovair et al, 1990; Kieras, 1988; Kieras & Bovair, 1986; Kieras & Polson, 1985; Polson & Kieras, 1985; Polson, et al, 1986; Ziegler, Vossen, and Hoppe, 1990) for the idea that users' knowledge can be usefully described as production rules, and that learning time can be predicted by the number of rules to be mastered. And the questions raised by Polson and Lewis's CE+ framework regarding methods of causal inference and problem solving which come into play in learning and in transfer of training serve to place theories of knowledge representation and organization in a broader cognitive context. Contemplation of this context should suggest future avenues of research.

Nevertheless, it is necessary to note that every essential aspect of CE+ seems limited or open to question. Limitations of the conditions for which the predictions of CCT apply have been noted (e.g. Tetzlaff, 1987). It may be asked what other problem solving activities users might engage in, and what factors influence them. It is certainly open to question whether a production system is the best model of a user's knowledge. For present purposes it is noteworthy that this formulation makes no distinction between conceptual and other knowledge. Surely a label-following strategy in searching menus could, for example, be an indication of search for labels related to concepts or abstract constructs known to be relevant to goals, as well as to goals themselves. That is to say, label-following must clearly be mediated by other forms of knowledge. The causal inferences of the EXPL component, for example, might in CE+ as it stands be restricted to simply descriptive knowledge, understanding of causal relations among concrete, observable entities. CE+ makes no provision for development and use of conceptual knowledge, or for use of prior conceptual or simply descriptive knowledge. After all, interaction with the interface is not the sole source of knowledge available to a user. Payne and Green (1989), as noted above, found the presence of familiar categories greatly facilitated rule learning, and the work of Tetzlaff (1987) further suggests that a description of knowledge in terms of number of rules is insufficient.

Analogy is one mechanism by which the role of prior knowledge can be incorporated in user models. Essentially the same, common production-based, model of transfer described by Kieras and Polson's CCT has also been put forward by Singley and Anderson (1985, 1988), who observed positive transfer between text editors. Singley and Anderson (1985) also noted the likelihood of transfer when surface features differ but underlying conceptual structure is nearly identical, that is, transfer based upon analogy. Polson and Lewis (1990) suggested analogical generalization along lines suggested by Anderson and Thompson (1986, cited in Polson & Lewis,1990) as a way to use an EXPL causal analysis to create procedures for new but related goals. A number of authors (e.g. Carroll and Thomas, 1982; Gentner, 1983; Halasz and Moran, 1982; Holyoak, 1985; Holyoak & Koh, 1987; Holyoak & Thagard, 1989; Rumelhart & Norman, 1981; Schumacher & Gentner, 1988) consider analogy a very important method in problem-solving, or have suggested that transfer of training can be analyzed as analogical mapping. Clement (1980) observed spontaneous use of analogies in solving physics problems. Inclusion of analogical processes in the problem-solving stage may be a necessary step in making a CE+ -like framework more complete and viable.

Analogical Reasoning

Gentner and her colleagues (e.g., Falkenheimer, Forbus, & Gentner, 1986; Forbus & Gentner, 1986; Gentner, 1983; Gentner, 1989; Gentner & Toupin, 1986; Goldstone, Medin, & Gentner, 1987; Schumacher & Gentner, 1988) have made persistent efforts to analyze transfer as analogical mapping. As noted below in the discussion of mental models, however, analogy or metaphor may also remain an important component of user knowledge. Forbus and Gentner (1986) asserted their belief that people use analogies to create mental models that play a very important role in learning.

Gentner's structure-mapping theory (Gentner, 1983) holds that analogy consists in mapping knowledge from a familiar, or base, domain to another, generally less familiar, domain, the target. Mappings, according to this theory, are constrained by rules. Briefly, attributes of objects are dropped in mapping, while some relations between objects in the base are retained in the mapping. The point of the mapping is to maximize the overlap in relational structure between the two domains. Isolated objects are less likely to be mapped than objects which belong to a mappable system of relations. Therefore object correspondences are determined not by similarity of the objects but by their roles in matching relational structures. However, analogies are more easily grasped when corresponding elements are similar.

Schumacher and Gentner (1988) found that transfer was facilitated by both transparency, the degree to which corresponding elements are similar, and by systematicity. A systematic model is defined as one in which there exist higher-order constraining relations. Systematicity had a strong effect on learning the base, and both systematicity and transparency had strong effects on transfer.

Holyoak (1987) presented a model of analogy similar to Gentner's, which has been implemented as a computer program (Holyoak & Thagard, 1989). Holyoak's model makes the same prediction of facilitation from transparency and systematicity, or surface and structural similarity in Holyoak's terminology, that Gentner's model does. Holyoak held that current goals will be a strong influence upon which source analogs, or base domains, will be retrieved, and which aspects of the source will be mapped onto the target. This is a neglected area in Gentner's theory.

Studies such as Schumacher and Gentner (1988) have demonstrated positive effects of analogical mapping for transfer of training. Analogy may also be a source of negative transfer on occasion. Halasz and Moran (1982), Douglas and Moran (1983), and Allwood (1990) analyzed subjects errors on the basis of analogies from other domains. Douglas and Moran (1983), found that 60 per cent of novice's errors with the EMACS text editor seemed to be due to misplaced analogy with the typewriter. Such substantial effects deserve recognition in relatively comprehensive models such as CE+.

It should be noted that analogical reasoning does not require conceptual knowledge. But the effect of systematicity (Schumacher and

Gentner; 1988) or structural similarity in the terms of Holyoak (1987), suggests benefits for effective analogical reasoning of knowledge of the "operational structure within itself and between associated concepts", according to Tennyson and Cocchiarella (1986) the defining feature of conceptual knowledge. Knowledge of such operational structure might also be described as a mental model.

Mental Models

Knowledge organization is an important area neglected in production system models. Such organization may be transformed with the accumulation of experience and expertise. While it is possible to limit discussion of knowledge organization to which productions are invoked in a given context, it seems clear that in doing so other potentially important knowledge structures may be ignored, or may be unnecessarily cumbersome to describe. Tetzlaff's (1987) study, in which subjects who learned subsets of a command set showed effects of whether the subset was selected and presented in a coherent fashion, provides one clear demonstration that prior knowledge and its organization may affect both learning and transfer.

Knowledge organization is often discussed in terms of the construct "mental model". Discussion of knowledge organization must also touch upon the previously-discussed topics of production systems and of analogical mapping. Indeed, the topic of knowledge organization can encompass most of the subtopics of psychology. However, for the present purposes knowledge organization will be discussed under several subtopics, related to the existence and nature of mental models, the distinctions between procedural, declarative, and conceptual knowledge, and an occasional unavoidable digression into other areas.

Norman (e.g., 1983) drew attention to the distinction between the engineer's *conceptual model* upon which design of the system is based, the user's *mental model* of a system and the researcher's model, or *conceptualization of the model* in Norman's term, of the subject's model. Another of Norman's terms is *system image*, which refers to the information about the system that the system itself presents to a user. Knowledge gleaned from the system image, along with other sources such as manuals and explanations from other people, is presumably used in development of a mental model.

For a construct which is the object of a great deal of research, the "mental model" is quite vaguely defined. As an example, Gentner and Stevens (1983) provide no explicit definition of "mental model" in the introduction to their book on the subject. As is often noted (e.g., Young, 1983; Rouse & Morris, 1986) the usage of the term mental model varies between domains and researchers. Carroll and Olson (1987, p. xvi) identified a number of research needs related to mental models, among them "Produce evidence that people have and use mental models", and "Detail what a mental model would consist of and how a person would use it to decide what action to take next". Another problem mentioned by Rouse and Morris is differentiating the "mental model" from knowledge in general. Sebrechts, Marsh, and Furstenburg (1986) distinguished a mental model from a schema (Bartlett, 1932) in the following way:

> "A schema is usually taken to refer to a specific organized structure in memory and is generally associated with declarative knowledge . . . A mental model, in contrast, usually implies a representation that captures the working of some device; it is generally associated with procedural knowledge. The boundaries between these and other forms of knowledge are not well defined. In addition, it is likely that the learning of procedures includes multiple representations, since declarative knowledge constitutes a central element of early skill acquisition."

Yet other uses of the term (e.g., Gillan, Cooke, & Breedin, in press) seem to imply that mental models need not exclusively concern procedural knowledge. Certainly as a practical matter there is a need for constructs that encompass all aspects of a user's knowledge that can have significant effects upon performance.

Probably the modal view of mental models is essentially the one adopted by Rouse and Morris as sufficient to allow meaningful enquiry into the nature of mental models. A mental model is a cognitive representation that allows the user to describe, explain, and predict the behavior of a system. That users' mental models support explanation seems to suggest that users' knowledge may have more organization and more conceptual content than is best discussed as a set of production rules.

Rouse and Morris (1986) identified a number of salient issues regarding mental models. For the present purposes, the most important of these issues are: accessibility of mental models; form and content of representation of mental models; the nature of expertise, cue utilization (see also the discussion of selection of base domain and features to be attended to in analogical mapping, above), and instructional issues. These topics, after a further discussion of the general nature of mental models, will be treated to some extent in what follows. Given the number of issues with respect to mental models remaining to be investigated, and the complexity of computer systems, it is perhaps worth noting the advice of Gentner and Stevens (1983, p.2) that "Our first efforts to capture naturalistic human knowledge must necessarily center on the simplest possible domains." Yet despite the difficulty of the enterprise, a number of HCI researchers are investigating the nature and influence of mental models in computer use, surely not a particularly simple domain.

According to Norman (1983) mental models are incomplete, unstable, unscientific, and without firm boundaries. People's ability to run their models is limited, people are apt to forget the details of the system and to confuse similar devices, and people maintain superstitious behavior patterns or perform extra physical actions in order to avoid mental effort. Norman described three functional factors that he considered to be necessary properties of mental models. The model reflects a person's beliefs about the system; parameters and states of the model accessible to the person should correspond to the aspects and states of the system that the person can observe, and the model must have predictive power. Wilson, Barnard, and Maclean (1990), using three different assessment techniques to capture some of their subjects' learning of a computer system, arrived at conclusions recalling Norman's characterization of the mental model. They concluded that the representation underlying their subjects' learning was best described as a repertoire of knowledge fragments, some accurate to the system, some not. With learning, fragments may combine and there may be loss of false information but acquisition of new false information.

Several workers have offered taxonomies of mental models. Young (1983) also proposed several criteria for evaluating ideas regarding users' conceptual models (UCMS), then tentatively suggested eight different kinds of mental models. Young's criteria were aspects of user's performance, such as choice of method, timing, and locus and nature of errors; aspects of learning, such as what is retained and what forgotten, long-term memory distortion, and generalization and over-generalization; and reasoning about a system, such as explaining and predicting it's behavior and inventing methods. Young's last criterion is that the model should provide guidelines for a good design. Perhaps this requirement reveals the aptness of Norman's point regarding conceptualization versus mental model, for this criterion may embody an assumption stemming from Young's background and preferences rather than data about users. Young's proposed varieties of model are:

1. Strong Analogy: The device (D) is so similar to another device D' that the representation of D' can be used as a model of D.
2. Surrogate: A mechanistic account of how the device works
3. Mapping, or Task /Action Mapping: focus on the model as a mediator between the user's intended task and necessary actions
4. Coherence: The model is a schema (Bartlett, 1932). Individual facts may be distorted to fit the schema or to contrast with it.

5. Vocabulary (Newell, 1980; cited in Young, 1983): The model is the set of terms in which the user encodes information. This suggests that facts which can be encoded in existing vocabulary will be easier to encode, and that Strong Analogy may have its effects by providing a ready-made vocabulary.
6. Problem Space (Newell & Simon, 1972): The model is the space in which problems about the device are formulated. But it is clear that much interaction need not involve problem solving.
7. Psychological Grammar: The model serves the same role as a grammar does for a native language. Young considered this idea less concrete and tractable than that of mental model.
8. Commonality: The model is constructed by the observer who posits a common data structure to explain all the user's interactions with the device.

While, as Young admits, some of the suggestions are not necessarily mutually exclusive, his list does illustrate the diversity of approaches to mental models. It is certainly possible to suggest other types of mental models, for example the set of productions or simply associations active in a particular context. This latter view implies that a subject may have multiple models of the same device, even ignoring the effects of experience, particularly as context may include task and goal. Indeed, the idea of multiple models is not inconsistent with Young's set of types, but might arise from processes not explicitly considered by Young, for example hypothesis testing. In particular in a problem-solving situation or during initial problem-solving and learning with a system, but perhaps in (all?) other situations as well, a mental model may be computed or inferred for each instance of use rather than simply retrieved.

It may be helpful at this point to recall Norman's (1983) idea of the system image. It should be noted, however, that the term "system image" is potentially misleading in implying a unitary construct. As discussed previously, some or much of users' knowledge is acquired in the course of problem solving activities. And the work of Tetzlaff (1987) implies strong effects upon knowledge organization of the order in which aspects of the system are learned. Hence each user encounters a "system image" corresponding to her own goals, prior knowledge, and problem solving methods and abilities. Despite and to even because of these facts, it is nevertheless true that there must be system image variables that, along with or in combination with the details of system functionality, help determine the nature of users' mental models in a systematic way. If so, many such variables are likely to be related to aspects of the user interface. These same variables may also bear upon which conceptualizations of the user model by researchers are most appropriate and useful for a given system. Study of such variables affecting formation of the user's model.must clearly be extraordinarily difficult for complex systems.

At all events, the work of a number of researchers can be classified roughly according to which of Young's types of mental model they posit. As discussed above, a number of workers (e.g. Carroll and Thomas, 1982; Gentner, 1983; Halasz and Moran, 1982; Holyoak,1985; Holyoak, & Koh, 1987; Holyoak & Thagard, 1989; Rumelhart &

Norman, 1981; Schumacher & Gentner, 1988) have examined analogical mappings, which seem to qualify as Strong Analogy. Surrogate models are represented by, for example, De Kleer and Brown (1983), who discussed "runnable" mechanistic models. Greeno's (1983) emphasis upon the conceptual entities available for representing a system, what he termed the ontology of a domain, is related to the vocabulary approach to mental models. An approach combining aspects of those of De Kleer and Brown (1983) and Greeno (1983) is that of Williams, Hollan, and Stevens (1983): Mental models are composed of autonomous objects with an associated topology, and are runnable and can (sometimes) be decomposed to produce models of the initial objects. In this view a mental model is a collection of connected objects, each with a set of associated rules which modify its parameters and specify its behavior. A change of state for an autonomous object is the replacement of one set of behavior rules by another. The model can "run" in response to such a change or as a result of propagation of information through the system.

Moray (1987) described *lattice theory*, which in Young's terminology should probably be classified as a form of a Task/Action Mapping theory. Moray pointed out that it is unreasonable to expect a mental model to be an isomorphism which maps each unique state of the system onto a unique state of the model. According to this theory, a mental model is a homomorph, a faithful model based on categories, of the real system. The user tries to construct simplified models that aggregate environmental states and system outputs into useful categories, and ignores details irrelevant to the goals for which the model is constructed. When an operator learns to control a complex system, she learns the transfer functions of a "black box". Model construction can be considered the refinement of what Moray calls a "quasi-morphism" or q-morphism. In a q-morphism a "higher" layer of the model provides default predictions unless an exceptional category evokes a "lower" layer of the model. The set of layers is the "lattice" in lattice theory. The model is, according to Moray, likely to be composed of what appear to the operator to be quasi-independent subsystems, because typically in complex systems some variables can range over quite large values without affecting the system, some variables can "dominate" others, etc. Eventually learning reaches an asymptote, when they model may account for all states likely to occur. This may mean that the user accepts limitations of one sort or another, for example a probabilistic model.

What empirical predictions does lattice theory make? The tendency to treat systems as sets of quasi-independent subsystems " we have seen operators do this in our experiments" (Moray, 1987, p.625) is one prediction. This limitation in the user's model suggests to Moray the possibility of assisting the user by providing paths through the lattice which would otherwise be inaccessible. Arguably this is a major function of the user interface. It goes without saying, however, that determining these paths for particular systems and users, or classes of systems and users, may prove quite difficult.

A number of authors (e.g. Young, 1981,1983) have discussed mental models in terms of mappings from goals or tasks onto actions or into terms of the device in more explicit terms than does, for example, CCT. Payne, Squibb, and Howes (1990) advocated the yoked state space (YSS) hypothesis of device models, a version of the problem space approach to mental models which they also related to task-action mappings. Moran (1983) had argued that the user must reformulate tasks in terms of the system, that is map an external tasks she brings to the system into an internal task allowed by the system. Payne et al. (1990) elaborated this idea by utilizing Greeno's (1983) notion of the conceptual entities of the user's problem space. According to the YSS hypothesis, the user's mental model involves at least two state spaces, the goal space and the device space, and a mapping in between them. Payne et al also introduced the idea of the minimal device space, which is the least complex device space which allows particular tasks to be performed at all. The user can transform the device state using device operators. Entities in the device space are related to entities in the goal space via a semantic mapping, allowing device states to represent goal states. It should be noted that the device space may provide device states which do not map onto goal states. Surely also with increasing elaboration of a device space there is the possibility of additions to the corresponding goal space as the user infers new possible uses for the device. Perhaps also goals can in effect be pruned form the goal space as a result of difficulty or impossibility of accomplishment with a given system.

Payne et al discussed two different modes of learning which according to YSS comprise an important aspect of the relation between device models and task-action mappings. These two modes are descriptions of ways that the user may understand the sequences of more primitive interactions with the device which may comprise device operators. The user may understand necessary primitive interactions by constructing either an *operational account* or a *figurative account.* An operational account combines primitives into aggregate operators, described as sentential operators by Payne et al, which affect the device space, but a figurative account elaborates the device space with conceptual operators to which the primitive operations apply, and allows the sentential operators to be decomposed into meaningful subcomponents. For example in a text editing context the operations called by Payne, et al. (1990) *delete string* and *move string* both require marking the string and then specification of a copy or cut. The concepts of cut and copy do not apply to the minimal device space, according to Payne, et al. A figurative account could add the concept of a buffer to the device space. Figurative accounts have advantages for the user, allowing more flexible use of the components of the deconstructed operators for other purposes. This distinction between operational and figurative accounts is clearly related to the distinction between conceptual and procedural knowledge of e.g. Tennyson and Cochiarella (1986). It would seem that an operational account corresponds to the rules described by Rasmussen (1986), with Payne, et al. focusing attention upon the user's ability or lack thereof to recompose operators into new rules on the basis of what Rasmussen would term knowledge-based performance.

Payne, et al. conducted three experiments testing hypotheses consistent with YSS. The first experiment demonstrated in a sorting task that experts generally sorted in a way consistent with use of the concept of a string of adjacent characters, but novices never did. This suggests that the concept of string is developed through experience with text editors, incidentally consistent with the idea of an interaction between the various forms of knowledge.

The second experiment bears upon the question of what effects might be expected in transfer from a command line interface to a GUI. This experiment examined subjects' inferences regarding text editing operations demonstrated in two editors, IBM Personal Editor and MacWrite, and transfer between them. It was found that novices were more likely to mark strings as units when marking was done by mouse and indicated by continuous growth of the highlighted text. Evidently the interface influenced induction of the string concept. Novices also tended to re-mark text already in a buffer. Apparently the buffer concept is not easily attained.

Payne et al also inferred from this experiment that device models influence transfer of methods from one device to another. Efficient marking of strings tended to transfer between editors, or as Payne et al put it, from the device that prompted induction of the string concept to one that did not. However, efficient recopying did not transfer from the IBM editor on which the to-be-copied string was still highlighted to MacWrite, in which the string was no longer highlighted. Payne, et al. took this to mean that users of the IBM editor did not induce the concept of a buffer, but formed a rule that if it's highlighted, you can copy it. Accounting for such results in a common elements/ production system theory of transfer is not straightforward, as Payne, et al.. pointed out.

Payne, et al.'s third experiment confirmed that induction of the string concept was not easy for novices and also suggested that the names of menu items play a role in encouraging development of device models.

Using STORE as the label for the copy-to buffer phase produced more instances of efficient recopying than did using COPY.

While the experimental results of Payne, et al. are certainly consistent with YSS, they are probably not uniquely so. This is the more true because of the considerable overlap among members of the plethora of theoretical approaches to mental models. However, the Payne, et al. (1990) results seem a sufficient suggestion of the importance of "figurative accounts", or induced concepts, to require further skepticism regarding the wisdom of representing the user's system knowledge, or perhaps it is best to say all users' approaches to system knowledge, solely as a Task/Action Mapping of any stripe, including a production system. An appropriate production architecture may in fact be able to represent concepts, but may not be a necessary or very efficient way to do so. And Payne, et al. (1990) further suggests that the interface can play a crucial role in promoting both rule and concept formation.

Wilson, Barnard, and Maclean (1990) concluded that subjects' knowledge was best described as a repertoire of knowledge fragments, some accurate to the system, some not. And Moray (1987) observed that subjects tend to treat a system as a set of independent subsystems. Such results raise the question of how well enhancing subjects' mental models might improve performance.

Empirical Approaches to User Knowledge and Knowledge Organization

Mental Models and Performance

If increasing experience with a system leads to greater consistency in user's mental models with the system model and with each others' mental models, it may be expected that providing users with the appropriate mental model will benefit performance. Whether this is so has been examined in a number of studies (e.g., Halasz and Moran, 1983; Kieras and Bovair, 1984; Sebrechts, Marsh, & Furstenburg, 1987).

Halasz and Moran (1983) conducted a protocol analysis of two experimental groups of users of a calculator, model users and non-model users. The two groups were trained identically except that training materials for non-model users made no mention of the calculator's stack. The model apparently had little effect upon routine problem solving, but significantly enhanced performance in novel situations. For more complex problems, model users employed what Halasz and Moran termed *model-based problem spaced search*, in contrast to no-model users who employed methods-based problem space search. In short, the model provided an advantage for novel situations, when it provided an effective problem space, while non-model users relied upon manipulating procedure fragments. The distinction between model-based and methods based search recalls the operational vs figurative distinction of Payne, et al. This result is also reminiscent of work in the study of expertise. For example, Chi, Feltovich, & Glaser (1981) found that more expert physicists sorted problems according to relevant principles of physics, while novices sorted on the basis of more superficial features. Such results also add point to the earlier suggestion that Polson and Lewis's (1990) CE+ fails to consider the effects of prior knowledge, and in particular conceptual knowledge, for example in problem solving.

Kieras and Bovair (1984) presented three experiments also contrasting the performance of users given a model which describes the internal mechanism of a device, a *device model* in the terminology of Kieras and Bovair, with the performance of a control group. The device used in these studies was a simple control panel. In the first experiment, the two groups of subjects also received procedural training. The model group learned a procedure faster, performed better on a retention test, performed faster and more often correctly on three different tests, and used short-cut procedures more often than the no-model group.

Kieras and Bovair's (1984) second experiment required subjects to infer procedures rather than learn them. The model group was able to infer correct procedures on the first attempt in almost all cases. In contrast, the no model group tried a large number of procedures on the first attempt. The model group's verbal protocols explained their actions and device behavior almost completely in terms of the model, while the rote , no-model group tended to use a systematic trial and error strategy.

Kieras and Bovair's third experiment tested what kinds of information from the device model enabled the facilitation of inference found in the second experiment. There was a strong effect of specific information relating controls and components to possible "power flow" in the imaginary device used for the study. From this Kieras and Bovair (1984) concluded that a mental model is useful only insofar as it directly supports precise inference of procedures. They felt that teaching users general principles, metaphors, or analogies, are unlikely to be useful, since they are unlikely to support precise inference. They also concluded that the user does not need a full understanding of the system in order to be able to make inferences regarding operating procedures, hence how-to-do-it knowledge can be superficial and incomplete. Furthermore, a device model will not always be useful: the telephone is an example of a common device for which supplying a device model is apparently unnecessary. And an incorrect or poorly learned device model may even be misleading.

Kieras and Bovair (1984) carried out their study on a contrived control panel, but similar results have been found for computer tasks. Schmalhofer (1987) carried out a similar study for text editing, which also examined the consequences of two forms of instruction, mental model instruction or procedural elements instruction. The procedural elements instruction was based upon the principles used in Kieras and Bovair (1984). Subjects were then given varying amounts of practice in editing tasks, followed by testing on a different set of editing tasks. The mental model group performed significantly faster than the procedural elements group for all three training times, both during practice and on the subsequent test tasks.

There have been a number of studies in which it has been assumed that mental models can be treated as independent variables, that is to say manipulated experimentally. Halasz and Moran (1983) found providing a mental model to assist performance in novel situations, but the work of Kieras and Bovair (1984) suggested that this is so only if the model supports direct inference of procedures. Of course, the limited time and resources available to subjects in such studies, and the restricted nature of the problems upon which they are tested, may conceal some of the value of mental models involving conceptual information. But the results of Kieras and Bovair (1984) also highlight the potential of the user interface to make conceptual information more or less useful in novel situations. If conceptual information can readily be translated into the terms of the device to accomplish the user's goals, as described for example by Payne, et al.'s (1990) YSS, then performance will clearly be enhanced. It is possible to imagine a number of characteristics of the interface that might promote such translation, and this topic will be further discussed below.

But it is perhaps still more important to treat the user's mental model as dependent variable, to measure and characterize effects upon the model of experimental manipulations or of random variables.

Eliciting Mental Models

Much of the theorizing about the nature of mental models has been at most loosely based upon research. Of course, one way to approach the study of mental models is to somehow examine them more or less directly. Various methodologies have been used in the attempt to do so. Rouse and Morris (1986) noted four basic approaches to this task: inferring characteristics via empirical study, empirical modeling, analytical modeling, and verbal/written reports, each with strengths and weaknesses in different situations. Another approach, not explicitly mentioned by Rouse and Morris, and perhaps best described in a separate category, is the application of structural modeling, that is scaling and clustering techniques, to data. As Rouse and Morris noted, using multiple techniques can somewhat compensate for the weaknesses of any given method, but the possibility of totally capturing a mental model is remote, if only because it is unlikely that a mental model is a static entity having only one form.

Rouse and Morris felt that inference via empirical study can provide evidence for the effects of independent variables on mental models, but provide only indirect evidence for the form and structure of mental models. This is due to the likelihood of confounding with perception and response execution.

Empirical modeling may be possible when observations and actions are simple, and it can reasonably be assumed that processing involves only the mental model. Empirical modeling involves identifying the relations between a subject's observations and subsequent actions, and inferring the structure and parameters of the mental model from this relation. Several studies cited by Rouse and Morris (e.g. Rouse, 1977; Van Bussel; 1980; Van Heusden, 1980) have had subjects view a series of points along a time axis and asked for a prediction of the next point in time. After feedback, subjects were then asked to predict the next point, and so on. The relation of the series of predictions to the series of points was used to infer a subject's mental model. These studies found models differing systematically from the "true" system. This finding is reminiscent of the work of Sanderson (1989) discussed above with regard to the relation of verbalizable knowledge to skilled performance.

Analytical modeling, such as GOMS (Card, Moran, & Newell, 1983) or CCT (Kieras and Polson, 1985) is a common approach when empirical modeling is inappropriate due to potential confoundings. Analytical modeling involves using existing theory and data to make assumptions about the form, structure, and parameters of mental models for particular tasks. This allows calculations of human performance which can be compared with empirical data. It is common to adjust the parameters of the model to minimize the difference between real and calculated performance. If the differences which result are sufficiently small, the model can be adopted for predicting performance on the task. However, this is by no means demonstration of the true nature of a mental model.

The most straightforward attempt to identify mental models is to ask subjects for verbal or written reports about them. One common method is the verbal protocol, or "thinking aloud " (e.g. Newell & Simon, 1972; Ericsson & Simon, 1980, 1984). There has been considerable debate over the merits of the approach, with Nisbett and Wilson (1977) urging that subjects do not have access to their thought processes. Ericsson and Simon (1980, 1984) attempted to clarify the circumstances under which verbal protocols can be accepted as reliable data. As should be evident from the earlier discussion of declarative, procedural, and conceptual knowledge, verbal report is unlikely to be appropriate for eliciting all forms of knowledge. Verbal protocols provide evidence of what a subject is thinking about, but not on how the subject thinks. Verbal protocols may also lead to distortion of nonverbal images. Another verbal/written method of eliciting mental models is via interview or questionnaire. If the investigator's questions aren't task oriented, there may be no reason, apart from context effects, to collect information during rather than after performance.

Another approach to capturing mental models, less direct than protocol analysis and with some potential applicability to capturing procedural knowledge, is structural modeling. Structural modeling techniques are mathematical procedures for inferring the organization of a set of entities on the basis of proximity data for pairs from the set of entities. There are two basic categories of structural modeling techniques, scaling and clustering techniques. Scaled models assume the existence of a common set of measurable attributes underlying the set of entities under consideration, so that the scale values of the attributes uniquely define position within the domain. Attribute scales always apply to all entities in the domain. Clustering techniques, on the other hand, assume that local portions of the domain can have sets of attributes which define local structure but may not be applicable to entities in other parts of the domain. The varieties of Multidimensional Scaling (MDS), including Metric (Torgerson, 1952) and Non-Metric (Shepard, 1962; Kruskal, 1964) are examples of scaling techniques. The most commonly applied form of cluster analysis is Hierarchical Cluster Analysis (Johnson, 1967). Multidimensional scaling algorithms produce an n-dimensional spatial representation in which dimensions correspond to attributes common to the set of entities examined. Hierarchical cluster analysis produces a strict hierarchical tree structure. A promising recent development is the use of network clustering techniques such as Pathfinder (Schvaneveldt, 1990; Schvaneveldt & Durso, 1981; Schvaneveldt, Durso, & Dearholt, 1985). Network clustering results in a graph structure that is not restricted to a hierarchy. The essential idea implemented in the Pathfinder algorithm is that proximities between entities are the weights of links in the network, and a proximity should be represented as a link in the network when that link forms the minimum weight path between the entities involved in the full network (Schvaneveldt, 1990). Links in a pathfinder network may be directed, allowing asymmetric relations between nodes in a network.

The output of scaling and clustering techniques is a representation of structure, rather than content. The algorithms for the various techniques do not provide methods for obtaining or identifying the entities to be analyzed, or for meaningfully labeling network links or the dimensions. Cooke & McDonald (1989), in the context of knowledge engineering for expert systems, discussed methods for eliciting items or concepts in a particular domain. The methods reviewed include having a subject list concepts or steps, obtaining concepts through interview, and asking the subject to list chapter headings and subheadings in a hypothetical book about the domain. Cooke found that the type of knowledge elicited depended upon the elicitation technique used, with, for example, the chapter task generating mostly concepts.

Methods similar to those described by Cooke can be and have been applied in HCI. One resource sometimes available in HCI is the availability of a ready-made "system model" provided in system documentation (see e.g. Kellogg & Breen, 1990). It is often the case, however, that concepts and system models are provided rather arbitrarily by the experimenter.

Kay and Black (1984, 1985) applied MDS and hierarchical and overlapping cluster analysis to similarity data from expert, novice and naive users of an EMACS-style text editor. Subjects rated the similarity of all pairs from a set of fifteen commands chosen by Kay and Black. The three-dimensional multidimensional scaling solution seemed to reveal a high-level goal space for expert and novice users, with the dimensions representing a general goal-classification scheme for commands. No clear dimensions emerged for the naive group.

Two of the three dimensions were interpreted as the same for experts and novices. The first dimension seems to be an *editor/system* dimension, with editor commands like Put and Insert at one extreme and system commands like Exit and Cancel at the other. The second dimension appears to be a *formatting/nonformatting* dimension, with commands like Center and Justify at one extreme and commands such as Pick and Delete at the other extreme.

More interesting is the third dimension, which is different for experts and novices. For novices this appears to be a *sequence* dimension, with commands such as Argument, Search, Mark and Home, which are characteristically used at the beginning of a sequence, at one extreme, and commands such as Exit, Put, and Replace, typically used at the end of a sequence, at the other extreme. By contrast, the third dimension for experts seems interpretable as an *instrumental/achievement* dimension. At one extreme are commands such as Mark which are instrumental in achieving other goals. At the other extreme are commands such as Replace which result in the immediate attainment of an editing goal. The MDS findings in this study bear an interesting resemblance to the ideas of Payne et al. (1990) in the appearance of a goal space and in the nature of the expert-novice difference found. The novice's sequence dimension recalls Payne et al.'s operational explanations, while the instrumental/achievement dimension of experts implies a more figurative understanding of commands. The dimensions in the MDS solutions suggested to Kay and Black (1985) that general editing goals, common to all users regardless of level of expertise, are used to classify commands in order to ensure accessing the correct commands. This suggests, consistent with the theorizing of Payne et al (1990), that users of different systems with similar intended functions may have very similar goal spaces, and that MDS studies of users of such systems might lead to solutions with a similar number of dimensions of similar meaning.

Rather than a global goal space, the hierarchical clusters for the expert group seemed to represent more local relationships between commands as they relate to particular goals, with two main types of relationship; similarity of use in a sequence, and similarity of action. In the novice hierarchical clustering most clusters seem to be based upon similarity of action, with no clusters representing command sequences. Kay and Black (1985) suggested that novices possess general sequence information, as evinced by the begin sequence/end sequence dimension in the MDS solution, but that this information is not specific as to relationships between individual commands.

Finally, Kay and Black (1985) performed a confirmatory analysis examining the fit for data from the three groups of subjects to a designer model, using Overlapping Clustering. The designer model was formed using properties characterizing commands elicited from one of the designers of the editor. The designer model accounted for 33.1% of the variance for the expert group, 15.8% of the variance for the novice group, and a non-significant 9.6% of the variance for the naive group. Kay and Black took this as evidence that experts' mental representation does not completely match the design of the system. As an example, in the designer model there was a buffer-oriented cluster, while none appeared in the experts' model. Kay and Black thought that the implementation of the system made the buffer invisible to the user, and that a different implementation might make the underlying concepts more obvious.

In addition to MDS and Hierarchical Cluster Analysis, as applied by Kay and Black, Pathfinder has been employed in the attempt to "capture" mental models of computer systems (e.g. Gillan, Cooke, & Breedin, in press; Kellogg & Breen, 1990). Kellogg and Breen (1990) undertook a Pathfinder analysis of the effects of experience with a command-driven text formatting system, reminiscent of the confirmatory phase of Kay and Black (1985). Kellogg and Breen (1990) also postulated that experience in using a system should result in an increase in overlap between the user and system models. To create a system model, Kellogg and Breen (1990) applied Pathfinder to distances between concepts in a hierarchical structural diagram of the system which appeared in the system's documentation. User's models in the form of Pathfinder networks were derived from having users and nonusers sort all 51 of the systems commands into piles that seemed related. Finally, for each subject the correlation was found between that subject's sort of the command set and the system model. As expected, the correlation for the experienced group, .48 was significantly greater than that for the inexperienced group, .30. Kellogg and Breen also examined the correlations of each subject's sort with those of all other members of the same experience group. Replicating previous results (Cooke &

Schvaneveldt, 1988), the mean correlation between experienced users (M=.37), was significantly higher than the mean correlation between non-experienced users (M=.20).

It might be interesting to consider the nature of the within-groups correlations in this sort of paradigm. For example, how much of the correlation between experts is due to agreement with the system model? Is it the case that there are portions of the system model with which all or most experts' models agree and other portions which are less likely to appear in experts' models? Is some of the agreement among inexperienced subjects, or experts for that matter, due to common misapprehensions? Such information for a particular system could serve as a guide to creation of training systems and/or redesign of the system to enhance transparency.

Gillan, Cooke & Breedin (in press) used hierarchical cluster analysis and Pathfinder analysis to investigate differences among three groups of subjects in their mental models of the human-computer interface. Subjects were two types of experts: software experts and human factors experts, and a control group of computer users with no experience in software design. Not unexpectedly, both expert groups had more, and more elaborate, clusters than did novices. There were also some notable differences between the clusters produced by the two types of experts. Perhaps most notably, software developers included software concepts concerned with the user interface and applications in a larger cluster of user interface elements, while human factors experts categorized those concepts in a separate cluster unconnected to other user interface concepts.

Novices' Pathfinder networks were complex and difficult to interpret. But Gillan, Cooke, and Breedin's Pathfinder analysis suggested that the groups of experts differed in the organization of their cognitive models. Human factors experts networks consisted of distinct subnetworks, tending toward heavy internal interconnection but with only a single connection to other subnetworks. Software experts' networks seemed to have more complex interconnection schemes, with central nodes, and simple and complex subnetworks. Gillan Cooke, & Breedin speculated that human factors specialists' models may be relatively closely tied to function, while software specialists' models may represent a compromise between function and implementation. For the present purposes this kind of finding is suggestive of the possibility that experts in different areas may encounter differential benefits and problems in transferring to a new OS or user interface. At the same time, the experience of the expert groups apparently did lead to greater organization and agreement in Pathfinder networks, somewhat as found by Kellogg & Breen (1990).

What can be gleaned from the plethora of divergent theoretical frameworks developed by the various researchers in the field of mental models and HCI? Despite the differing emphases of the various models, most of them involve one or more of a set of related ideas or, not surprisingly, make similar predictions. For example, Kay and Black (1985) and Payne et al (1990) both suggest the existence of a goal space. It is interesting to compare the goal space in the YSS theory of Payne et al. with the job situation knowledge in the original framework for the CCT theory of Kieras and Polson (e.g., 1985). The apparent differences between the models held by different groups of experts found by Gillan Cooke, & Breedin may illustrate the importance of goals in the development of mental models. At the same time, the convergence of models with increasing expertise (Cooke & Schvaneveldt, 1988; Gillan Cooke, & Breedin, in press; Kellogg & Breen ,1990) holds out hope that, despite the numerous influences upon users' knowledge acquisition, such as differing goals, that may lead to widely varying models of the same system, it should be feasible to gain some understanding the effects of interface variables upon user knowledge.

Much as there is apparently convergence in the mental models of expert computer users, there is also some convergence in researchers' conceptualizations of the user model. But despite the tendency of numerous researchers to adopt at least related theoretical frameworks, there is no "modal model" adopted in HCI. And more importantly, no carefully elaborated body of empirical work supports the constructs embodied in the theorists' speculations. The focus of the present effort is exploration of users' knowledge in a practical setting. Before further description of the present research it is useful to discuss that setting.

Command-Based Versus Graphical User Interfaces

The present research program attempts to examine knowledge representation and organization of users of the DOS command-based interface and the Macintosh GUI. The graphical user interface (GUI), or, usually taken synonymously, the direct manipulation interface (DMI) (e.g., Shneiderman, 1982, 1987) is a development in the computer industry which has been widely, often glowingly, endorsed as an improvement in ease of use. Command-based interfaces, which came into wide use prior to GUIs, require users to recall or find commands and syntax, which can be a source of difficulty (Michard, 1982; Borgman, 1986).

According to Shneiderman (1987), the central ideas of direct manipulation seem to be:

> "Visibility of the objects and actions of interest, rapid reversible incremental actions, and replacement of complex command language syntax by direct manipulation of the object of interest--hence, the term *direct manipulation*." (p. 180).

The now-modal implementation of DMI is exemplified by the interface to the Apple Macintosh operating system, in which cursor movement is controlled by a mouse or other pointing device, programs and/or documents are represented by small graphics called icons, and many actions are performed by use of a pointer, such as a mouse, rather than by typing commands. Such interfaces generally employ what is referred to as a desktop metaphor, intended to facilitate learning and performance by allowing users to manipulate seemingly familiar "objects". These systems in general actually employ hybrid interfaces incorporating menus, accessible by selection with a pointing device, and in some cases commands as well.

As Shneiderman (1982, 1987) also pointed out, there can be disadvantages as well as advantages to DMIs. For some combinations of tasks and subjects, graphical representation may slow rather than facilitate performance (e.g. Shneiderman, 1982). Of course, graphical information and the methods available must be meaningful to the user. Despite the intentions of the desktop metaphor, there is learning involved for GUI users (e.g., Whiteside, Jones, Levey, and Wixon, 1985), and analogical representations can be misleading. Certainly also graphical representations require considerable display space. In some cases a list of names may be more useful, and much more compact, than a set of icons. And typing can be faster than making mouse movements.

Empirical evidence regarding the relative merits of GUIs and other interfaces, particularly command interfaces, is rather sparse and what evidence there is, somewhat mixed. Some studies have found advantages for the GUI (e.g., Davis and Bostrom, 1989; Woodgate, 1985). Other studies have found little difference or even advantages for command language interfaces (Carroll and Mazur, 1986; Dumais and Jones, 1985; Lansdale, Simpson and Stroud, 1987; Whiteside et al, 1985). There are a number of possible reasons for these equivocal findings, including the nature of the tasks examined and use of differing hardware platforms (Whiteside et al., 1985) for different OSs, and subject selection. For example, the advantage in GUI to command-based transfer as opposed to transfer from command based to GUI operating systems found by Whiteside, et al.. was probably, as the authors acknowledged, due to superior performance with a particular command system in which online help appeared in an onscreen window. And it is possible to wonder whether users of graphical systems might for some reason be better able than users of command-based systems to transfer to another type interface. But, as the authors also pointed out, GUI users must learn a fairly complex syntax of mouse movements and button clicks, and Whiteside, et al.'s subjects received no training at all. In many respects it may be more reasonable to compare ease of use given minimal training rather than given no training at all.

However great their objective merits, it has become widely accepted that GUIs have advantages over command interfaces, in particular that GUIs are easier to use and require less training. Shneiderman (1987) stated that direct manipulation interface users' reports are filled with positive feelings, including feelings of mastery of the system, ease in learning the system originally and in assimilating advanced features, enjoyment, and desire to explore more powerful aspects of the system. Currently, GUIs are being developed as replacements or supplements for command interfaces such as MS-DOS and UNIX. Examples of this trend are OS/2, Windows, and PC Geos for MS-DOS machines, and Open Look, Motif, and Next Step for UNIX systems. These interfaces, like that of the Macintosh, are in general mixtures of GUI and/or menus or command lines, but are usually informally referred to as GUIs.

The wholesale migration to GUIs strongly suggests several questions, both theoretical and practical. How do GUIs and command-based interfaces differ with respect to cognitive complexity and, in terms of the present research, effects upon users' knowledge representation and organization? And perhaps most obviously, what needs to be done, and what can be done, to facilitate learning to use software which runs in a graphical environment for new users or for the many users accustomed to a command-based user interface? An understanding of the effects of the interface upon user knowledge may help to answer the transfer question by suggesting ways in which the knowledge developed by users of one system type may mismatch the requirements of another system type. Ideally training programs should, for both expert and less expert users of command interfaces, capitalize upon opportunities for positive transfer , while minimizing negative transfer.

More generally, Foss and DeRidder (1987) proposed that the selection criterion for choosing between different interface designs should be the maximization of transfer of learning for the successor design. Certainly there might be problems with this approach. One example is possible conflict between maximizing transfer and enabling new capabilities. Maximal transfer, of course, should occur with no change at all. And maximizing transfer might quite conceivably have the effect of restricting ease of use or ease of learning for users not experienced, or not much experienced, with the previous system. It is evidently premature to hope for maximal transfer before understanding the cognitive and other characteristics of interfaces, and indeed, tasks, that dictate ease of learning and use. The emergence of the GUI seems to be a somewhat neglected reason and opportunity to explore this subject further.

The migration to the GUI is not a sudden phenomenon or a new opportunity. The Macintosh, for example, has been available since 1984, and many Macintosh users had earlier experience with command interfaces. Versions of Windows have been available for about as long, and Digital Research's GEM environment for the PC has also long been available. There are other examples. Yet, despite the wide recognition in HCI circles of the importance of GUI and direct manipulation interfaces, the topic of transfer between command language and GUI is little studied, hence of particular topical interest. Nonetheless, Whiteside et al (1985) may be the only study (until very recently) of transfer between command-based and graphical OSs, though there are other studies of comparative performance and /or of training requirements. For example, Davis and Bostrom (1989) trained students recruited from a computer class to use either a DMI (the Apple Macintosh) or a command-based interface (MS-DOS). DMI users performed better and perceived the system to be easier to use than did users of the command-based interface. Dyck (1990) attempted to study transfer of knowledge from DOS to the Macintosh Finder, but prior knowledge seemed to have no effect, perhaps, as the author suggested, because subjects did not have enough prior knowledge of DOS.

A Research Program

It has been seen that there are a number of theoretical frameworks that attempt to characterize computer users' system knowledge, including GOMS (Card Moran & Newell, 1983), CCT (Kieras and Polson, 1985), CE+ (Polson and Lewis, 1990), YSS (Payne, Squibb, and Howes, 1990), and a number of others. In most cases the theorizing appears to have occurred in advance of the evidence, despite the advice of Sherlock Holmes. While there are some, largely isolated results supporting particular theoretical frameworks, none can be said to be based upon careful validation of the constructs involved. A good example of these problems is CCT. While there is evidence in support of CCT's predictions with regard to learning and transfer, (e.g., Bovair et al, 1990; Kieras, 1988; Kieras & Bovair, 1986; Kieras & Polson, 1985; Polson & Kieras, 1985; Polson, et al, 1986; Ziegler, Vossen, and Hoppe, 1990), researchers with other interests have shown that these predictions apply in only rather restricted conditions (e.g. Payne and Green , 1989; Tetzlaff, 1987). And several of the alternative theoretical frameworks are so general as to be able to accommodate any data, that is they are difficult to falsify. For these reasons it is important to attain an empirical understanding of users' actual knowledge, both under laboratory conditions and in the real world. An assay of what phenomena are prominently in need of explanation will help separate the theoretical wheat from chaff.

The general focus of the present research effort is how computer users' knowledge representation, and particularly concept formation is affected by the user interface. This might be seen, in part, as an analogy with the Whorfian hypothesis (Whorf, 1956). Just as characteristics of a given language may affect the facility with which people may form and use particular concepts, different interfaces may differentially promote the formation and use of particular computer-related concepts or other forms of system knowledge.

The question of how users represent system knowledge and of how knowledge representation and organization is affected by the interface has several prominent corollary questions. How does increasing expertise interact with the effects of the interface upon knowledge representation? Is it feasible to intervene in order to compensate for any interfaces which afford poor knowledge representation and concept formation? And, of course, what is the effect upon transfer to a new system of knowledge about the old system?

The proposed research program is an empirical, rather pragmatic and exploratory, effort to characterize knowledge representation and organization by, in particular, users of DOS and/or the Macintosh OS. The initial study examined the effects of two variables, expertise and interface type, upon computer users' knowledge of their systems.

The study was conducted using volunteer subjects of varying prior experience. Due to the presence of subject variables, and hence the non-experimental nature of this research, and to the quite variable experience and skill of the subjects, it is perhaps more natural to consider the findings in terms of the case study approach current in neuropsychology (see e.g. Shallice, 1979; Ellis and Young, 1988), rather than only in terms of treatment groups. The varied experience of the subjects means that averaging might well hide interesting individual differences, and also that subjects cannot very readily be ascribed to distinct groups. Accordingly, the aim of the study was to characterize the knowledge, and the organization of that knowledge, of individual subjects as well as to examine group differences . An attempt was made to characterize users' knowledge regarding a set of important computer concepts as descriptive, procedural, or conceptual, and users' organization of these concepts was assayed using a relatedness rating task and Pathfinder network clustering. Target concepts for the study, which apply to both operating systems, include:

1. Directory/folder
2. File Format
3. Data versus program
4. Storage buffer/Clipboard

It was predicted that the command-based interface better supports the discrimination of the related concepts of data and program. The

GUI was predicted to better support the concepts of folder and clipboard, while both interface types should support the concept of file format. It was hypothesized that users of both interface types are likely to have procedural knowledge allowing use of constructs and functions that are at best imperfectly understood, and conversely declarative and conceptual knowledge that cannot be readily used.

In general there are three factors that seem likely, a priori, to differentially affect users' system knowledge and it's usefulness. The DMI of the Macintosh may better support user's acquisition of concepts by enhanced salience through visual representation and ready availability of related commands in system menus. Acquiring conceptual knowledge may have lower cost in terms of cognitive effort for Macintosh users. At the same time, a second, related factor comes into play: that same knowledge may have lower value due to the greater general ease of use of the Macintosh DMI, which may lead to satisficing, and those concepts or rules that are learned may be less thoroughly learned in consequence. A third factor, however, may tend to promote the value of conceptual knowledge for Macintosh users: the consistency of the methods embodied in the Macintosh interface and Macintosh applications should make it easier to translate conceptual knowledge into rules for accomplishing tasks. The Macintosh interface may tend to satisfy the Kieras and Bovair (1984) criterion for value in mental models by more or less directly supporting inference of procedures based upon conceptual knowledge. The real-world effects of satisficing may be difficult to assess accurately in a controlled experiment. This is a further rationale for, at least initially, a quasi-experimental, somewhat exploratory approach to user knowledge.

Experiment 1

Method

Subjects. Subjects for this study were thirteen recruits from among the entering graduate students at the Jones Graduate School of Business at Rice University. The study required users of two different operating systems, one, DOS, with a command interface and one, that of the Macintosh, with a graphical user interface. Subjects proved to be a more homogeneous group than could be considered ideal , with many DOS users having some GUI experience, and many Macintosh users having DOS experience.

With the aid of a preliminary interview and of a questionnaire assessing computer - related experience, subjects were assigned to two groups on the basis of stated computer preference and home computer type, resulting in six Macintosh users, six DOS users, and one unclassifiable subject who uses a PC at home but who expressed a preference for the Macintosh. As noted, however, under these circumstances the significance of group membership is questionable for all subjects.

Subjects will be referred to by alphanumeric pseudonyms, M1 through M6 for Macintosh users, PC1 through PC6 for IBM PC users, and B1 for the unclassifiable subject.

Materials. A printed questionnaire was prepared to elicit information regarding subjects' experience with computer systems and applications. Another printed questionnaire took the form of a check-list of common computer tasks, with rating scales below each task for how often the task is done and how difficult it is. The experimenters also prepared printed lists of computer concepts, including a five point familiarity rating scale and space for subjects to provide a brief explanation of the concepts. The lists include concepts relevant to both of the operating systems/user interfaces along with a few concepts relevant to only one of the operating systems/user interfaces (see Appendix A). In addition a Hypercard program for the Apple Macintosh computer was prepared. This program is designed to present concepts from the lists previously mentioned and record subjects ratings of similarity. All rating scales used in this study range from 1 to 5, with one corresponding to the low end of every scale, such as low familiarity, ease, relatedness, or frequency. Additionally, the similarity rating scale for the last-mentioned task included a value of zero for unrelated.

Subjects were also required to verbally solve problems and to perform a set of practical tasks using a computer. The problem sets and task descriptions and instructions were printed. Two problem sets were prepared. Although the tasks involved on the two sets were conceptually very similar, each set was described in terms appropriate for a particular operating system, DOS or Macintosh. The practical tasks were performed using Lotus 1-2-3 and WordPerfect 5.1 on IBM-compatible PCs and Microsoft Word, Microsoft Excel, and Superpaint on Macintosh computers.

Copies of all materials used in this study, including replicas of the computer screen seen by subjects in the similarity rating task, are to be found in Appendix A.

Design and Procedure. The study involved nine types of task, administered in three sessions, as described below.

Session I:

1. Subjects began by completing a computer experience questionnaire, using pen and paper.

2. Following completion of the questionnaire, subjects were interviewed regarding their computer experience and knowledge. The interview was directed in part by questions suggested by subjects' questionnaire responses, in part by attempts to elicit demonstrations of understanding of particular computer concepts. The interviews were recorded on audio tape.

Session II.

3. Subjects were then called upon to solve a number of problems, giving verbal responses. These problems are intended to test understanding of basic OS and application concepts. Subjects were provided a printed copy of the verbal problems, but responded aloud. Subjects were given problems related to the OSs under study with which they were familiar . Responses were tape recorded and/or noted by the experimenter.

4. Subjects filled out a checklist of relatively common computer tasks, indicating those which they perform regularly and rating the frequency with which they perform those tasks and how difficult they find them.

5. Subjects were required to rate on paper the familiarity of fourteen computer-related concepts. Subjects also briefly explained or defined those of the concepts with which they rated familiarity greater than 2 on a scale from 1 to 5, with 5 representing very familiar..

6. Subjects were required to indicate, by selecting points on a scale displayed by a Hypercard program on an Apple Macintosh computer, the relatedness of all possible pairs of each set of fourteen concepts from the previous task. The scale ranges from unrelated to highly related, with four points indicated between. A depiction of this scale can be found in Appendix A. The Hypercard program presented concept pairs in random order and recorded subjects' relatedness ratings.

Session III

8. Subjects were tested for possession of some basic OS and application skills, for which understanding of the concepts used in the tasks described above, and particularly the target concepts, are relevant. Think aloud protocols (Ericsson and Simon, 1983) were collected during performance of these tasks. For both Macintosh and DOS users the skills tested included:

creating a folder or directory
copying a file between drives
opening a file of unidentified creator/file format
importing a file into a spreadsheet
creating a graph in a spreadsheet using an existing worksheet
opening a text file with a word processor

adding the previously created graph to a word processor document

moving the graph within the document

Subjects were tested on the computer type(s) with which they were familiar. Macintosh users were tested on an additional task:

copying a graph from the clipboard into Superpaint and predicting quality when printed

Results and Discussion

Description of Variables and Analyses.

A number of exploratory statistical analyses were conducted upon variables derived from all tasks: the Experience Questionnaire, the Task Checklist, the Concept Definition task, the Relatedness Ratings task, and the Performance Tasks. In a few cases the results of particular analyses are at least of suggestive interest with respect to the research problem, and will be discussed briefly. The grouping variable of interest is a random variable, computer system used by the subject, Macintosh or IBM PC compatible. The subjects in this study all have more or less experience with systems other than their current system of choice, including in a number of cases both Macintosh and PC experience. There are, accordingly, several relatively plausible ways to group the subjects, and it might be argued that selecting one may capitalize on chance. Statistical inference is questionable under such circumstances, and the results must be regarded as hints for future study.

For the most part results from this study will be discussed in terms of observations regarding individual subjects. In keeping with this case study approach, a summary was prepared for each subject describing that subject's computer-related experience, practices, preferences, grasp of concepts and operations, and performance. Portions of these summaries will be repeated in the discussion that follows. The complete summaries can be found in Appendix B.

Subjects' similarity ratings for the three classes of concept were submitted to the Pathfinder algorithm in order to derive network representations related to subjects' mental models of the operating systems and applications. Comparisons will be drawn between the models of DOS and GUI users and between those of subjects evincing varying levels and mixes of conceptual and operational knowledge. Average Pathfinder networks for Macintosh users and DOS users may be found in Appendix C, along with a Pathfinder network for each of the individual subjects.

A more quantitative summary of each subject's experience and knowledge as measured in this study, and of those of each group (the grouping scheme is described below) is presented in the form of Tables 1 through 3.

The analyses utilized a grouping variable, fifteen variables directly derived from subjects' responses, and two composites assessing knowledge of the target concepts and related procedures, or seventeen variables in all. Seven experience -related variables derived directly from subjects' responses on the preliminary questionnaire are presented in Table 1. Nine variables related to subjects knowledge of and self-rated ease of learning of concepts and procedures are presented in Table 2. Table 3 presents subjects' scores on the components of two composite measures from table 2, to be discussed below.

Two crude composite measures of conceptual knowledge and ability to perform related procedures were constructed. One measure, to be referred to as TC, attempted to assess subjects' grasp of the target concepts for the study. These target concepts are:

1. Program vs Document
2. File Format
3. Storage Buffer/Clipboard
4. Directory/Folder.

The other measure, to be referred to as O, attempted to measure subjects ability to perform operations related to these concepts. Scores for each concept ranged from zero to two for each measure, with a zero indicating no understanding or ability to use the concept or procedure, one indicating partial understanding or ability, and two indicating good understanding or ability. Subjects' conceptual knowledge of particular concepts was assessed on the basis of responses to verbal questions requiring such knowledge, and of performance on the definition task. Subjects' operational knowledge related to particular concepts was assessed on the basis of responses to verbal questions requiring such knowledge, of subjects' rated frequency and difficulty of performing related tasks from the Task Checklist, and on the basis of ability to accomplish the performance tasks.

The two composites, TC for conceptual knowledge and O for operational knowledge, simply summed the subjects' scores for individual concepts. See Table 2 for subjects' scores on these measures. Subjects' scores on the components of TC and O are to be found in Table 3.

The high ratio of variables to subjects is an indication of the exploratory nature of the study. The seventeen variables, along with short names for them, are:

1. Years of computer experience. YRS
2. IBM PC familiarity. PC
3. Macintosh familiarity. MAC
4. UNIX familiarity. UNIX
5. Mean familiarity with other computers. OTH
6. Number of applications used. APP
7. Number of computer-related courses. CRSE
8. Rated ease of learning new programs. LRN
9. Mean rated frequency of performing routine tasks. FREQ
10. Mean rated ease of performing routine tasks. EAZ
12. Number of concepts rated unfamiliar. UNF
13. Mean familiarity rating from Concept Definition. FR
14. Mean familiarity rating for familiar concepts. KF
15. Number of well-defined concepts. WELL
16. Understanding of target concepts. TC
17. Knowledge of operations related to target concepts. O

Table 1 presents the variables most directly related to experience: YRS, PC, MAC, UNIX, OTH, APP, and CRSE. Table 2 shows rated ease of learning along with variables that may measure knowledge of the concepts and procedures: LRN FREQ, EAZ, UNF, FR, WELL, KF, TC, and O. Table 3 shows subjects' scores for knowledge of target concepts (columns labeled C) and of procedures related to those concepts (columns labeled P).

Table 1

Subjects' Self-Reported Experience

Group	YRS	PC	MAC	UNIX	OTH	APP	CRSE
Mac							
M1	4	3	5	4	2	6	4
M2	3.500	3	2	1	3.500	4	2
M3	8	3	5	2	4	5	3
M4	8	1	5	1	3	5	0
M5	7	5	3	2	2	9	5
M6	9	3	4	1	3	5	2
Mean	6.583	3	4	1.833	2.917	5.667	2.667

table continues

Group	YRS	PC	MAC	UNIX	OTH	APP	CRSE
PC							
PC1	6	5	2	1	2	7	1
PC2	9	5	1	1	5	8	5
PC3	8	4	5	1	1	11	0
PC4	6	4	1	1	2	5	1
PC5	3	3	2	1	2	3	1
PC6	10	4	2	1	1	6	2
Mean	7	4.167	2.167	1	2.167	6.667	1.667
B1	8	3	3	2	1	8	2
Overall Mean	6.885	3.538	3.077	1.462	2.423	6.308	2.154

Note. YRS = Years of computer experience; PC = IBM PC familiarity; MAC = Macintosh familiarity; UNIX = UNIX familiarity; OTH = familiarity with other systems; APP = number of applications used; CRSE = number of computer-related courses taken.

Table 2

Subjects' Scores on Variables Related to Concepts and Operations.

Group	LRN	FREQ	EAZ	UNF	FR	WELL	KF	TC	O
Mac									
M1	4	2.56	3.89	5	3.286	6	4.556	6	5
M2	5	4.11	4.67	1	4.143	10	4.308	6	8
M3	4	4.39	4.89	0	4.429	13	4.429	6	6
M4	4	4.17	4.25	9	2.357	1	4.2	3	3
M5	4	3.44	4.83	3	3.357	8	3.909	7	7
M6	4	3.56	5.00	0	4.714	11	4.714	6	7
Mean	4.167	3.704	4.59	2.67	3.762	8.17	4.486	5.667	6

table continues

Group	LRN	FREQ	EAZ	UNF	FR	WELL	KF	TC	O
PC									
PC1	4	4.56	4.33	1	4.571	8	4.846	6	6
PC2	4	4.33	4.17	1	4.071	9	4.308	7	7
PC3	4	3.78	4.56	1	4.429	9	4.692	7.500	7
PC4	4	2.11	2.72	4	2.286	5	3.333	4	5
PC5	3	3.94	4.17	4	2.929	6	3.500	6	6
PC6	3	3.67	3.96	5	2.786	6	3.556	4	5
Mean	3.667	3.59	3.93	2.667	3.512	7.167	4.039	5.750	6.000
B1	3	3.11	3.78	3	3.071	8	3.900	5	5
Overall Mean	3.846	3.641	4.233	2.692	3.593	7	4.235	5.654	5.923

Note. LRN = ease of learning new computer programs; FREQ = frequency of routine tasks; EAZ = ease of routine tasks; UNF = number of unfamiliar concepts; FR = mean familiarity of concepts; WELL = well-defined concepts; KF = mean familiarity rating for familiar concepts; TC = understanding of target concepts; O = knowledge of operations related to target concepts.

Table 3

Knowledge of Target Concepts and Related Procedures.

	Directory/ Folder		File Format		Program vs Document		Clipboard		Mean
Group	C	P	C	P	C	P	C	P	
Mac									
M1	2	1	0	1	2	2	2	1	1.375
M2	2	2	1	2	2	2	1	2	1.750
M3	2	2	1	1	1	2	2	1	1.500
M4	1	1	1	0	1	1	0	1	0.750
M5	1	2	2	1	2	2	2	2	1.750
M6	1	2	2	1	1	2	2	2	1.625
Mean	1.500	1.667	1.167	1.000	1.500	1.833	1.500	1.500	1.458

table continues

	Directory/ Folder		File Format		Program vs Document		Clipboard		Mean
Group	C	P	C	P	C	P	C	P	
PC									
PC1	2	2	2	2	2	2	0	1	1.500
PC2	2	2	2	2	2	2	1	1	1.750
PC3	2	2	2	2	2	2	1.500	1	1.813
PC4	1	1	1	1	1	2	1	1	1.125
PC5	2	2	1	1	2	2	1	1	1.500
PC6	2	2	0	0	2	2	0	1	1.125
Mean	1.833	1.833	1.333	1.333	1.833	2	0.750	1.00	1.469
B1	1	1	1	1	2	2	1	1	1.250
Overall Mean	1.615	1.692	1.231	1.154	1.692	1.923	1.116	1.154	1.447

Note. C = knowledge of concept; P = knowledge of procedures related to concept.

An overall understanding of individual subjects is of particular importance because of the real-world, exploratory nature of this study. As has often been repeated, the results, certainly including any statistical results, of this study must be considered suggestive at best. Therefore in what follows the apparent characteristics of individual subjects will often be advanced as suggesting the presence of phenomena of interest to HCI and psychology in general.

This effort has yielded at least the appearance of several phenomena that seem likely to occur frequently in use of computers or complex systems in general:

1. An apparent double dissociation between conceptual and operational knowledge.

   A. Subjects display knowledge of concepts of which they are unable to make use in practice.

   B. Subjects display the ability to accomplish practical tasks without understanding, or fully understanding, the relevant concepts.

2. Context effects. Subjects may be able to recall relevant information, or apply relevant concepts, in some contexts but not others.

3. Effects of experience with particular computer systems. Effects upon which concepts and procedures subjects are likely to learn. There is some evidence consistent with the hypothesis that the Macintosh GUI affords better understanding of the Storage Buffer/Clipboard concept and related operations than does DOS.

4. Operation of heuristics for generating and testing hypotheses regarding procedures, and perhaps concepts.

5. Misunderstandings and/or superstitions, some of which may be commonplace.

Several of the above phenomena are plausible a priori, that is they appear likely based upon common sense. But several may not be entirely obvious, or seem not to be well explained by some of the better-known theories regarding human-computer interaction. Consideration of individual subjects will help to understand the origins of these phenomena in the interplay of subjects' prior knowledge and cognitive processes with the particulars of their system experience and the characteristics of interfaces.

Dissociation Between Conceptual and Operational Knowledge

Dissociations between what are usually termed declarative and procedural knowledge have often been observed in other contexts, e.g. Sanderson (1989). One difficulty for this area is potential for confusion regarding terminology. For example, Anderson (1982) has referred to procedural knowledge as non-verbalizable, compiled, declarative knowledge. According to Anderson's theory, subjects first memorize relevant facts, while performing on the basis of problem-solving procedures. There follows an associative phase, in which errors in understanding are corrected and associations required for performance are strengthened. This results in formation of a procedure, which becomes more and more automated. In this final phase of Anderson's model, the ability to verbalize knowledge relevant to the skill may be lost. This form of non-verbalizable skill is often referred to as procedural knowledge. This model may underestimate the role of conceptual knowledge in problem solving and performance. Subjects can undoubtedly also accomplish tasks on the basis of verbalizable rules or by making inferences or testing hypotheses based upon conceptual knowledge, perhaps in the form of a mental model (see e.g. Rasmussen, 1989). In an attempt to minimize confusion between procedural knowledge in Anderson's sense and knowledge of procedures in other forms, that is other forms of knowledge that support performance, all knowledge supporting performance of tasks related to the target concepts in this study will be referred to as operational knowledge.

In this study there was a high correlation between a measure of subjects' grasp of the target concepts (TC) and a measure of their ability to perform operations related to those concepts (O), $r = .824$, $p < .0005$. Nevertheless there is an appearance of dissociation between some subjects' conceptual and operational knowledge. Several subjects were unable to accomplish tasks despite having shown understanding of the relevant concepts. Others were able to describe how to perform or to actually perform tasks despite appearing not to understand the relevant concepts, or crucial aspects of the relevant concepts that might be describable as declarative knowledge in Anderson's sense.

PC3 is one example of a subject who seems to have conceptual knowledge that he is unable to apply in all situations, while M1, M4, PC1 and PC6 were able to accomplish some of the performance tasks without appearing to understand the relevant concepts. B1 seems to show both sides of the dissociation with respect to different concepts. PC3 and M1 seem to illustrate opposite sides of the dissociation for the same concept, File Format. Evidence from these subjects will be presented below. More detailed information regarding these subjects' experience and apparent knowledge may provide a useful context for the discussion, here and in later sections, and may be found in Appendix B.

Conceptual Knowledge Without Operational Knowledge

Subject PC3. Subject PC3 showed good understanding of the concept of File Format, but appears to have difficulties carrying out tasks related to this concept.

He seems to have a good general understanding of the concept of File Format, which he defined as "Type of file - which application and how it is configured or stored." In PC3's Pathfinder network File Format is linked to seven concepts, Application, ASCII, Creator of a File, Data File, File, Filename Extension, and Finder. He was one the three subjects to realize that a change of filename extension or icon might mean a change of file format (question 13). And he mentioned the possibility of saving to another format to facilitate using a document from another a different word processor (question 14).

Despite his apparent grasp of the concepts, PC3 evinced very limited ability to carry out related operations. His ratings on the Task Checklist are suggestive in this regard. He seldom saves a document to another format, and he finds it difficult to insert into a document something that originated in another program, but performs this task fairly often. When called upon to use his knowledge related to File Format in the Performance Tasks he was often hesitant, and was unable to complete the tasks of importing a graphic and moving it within a word processor document. Trying to open a file of unknown format, he first looked for utilities, then attempted to retrieve the file into Lotus 1-2-3. When that failed he did not attempt to import, but decided to use Word for

Windows "Because I can copy anything into that." Deciding somehow that it looked like a word processor document, he finally elected to convert from text, and succeeded in opening the file.

PC3 had no difficulty creating a graph with Lotus 1-2-3, but was unable to import it into Word Perfect, despite recalling the existence of a "neat embedding feature" from work experience. He attempted to use Excel and Word for Windows, and created a graph, but, unfortunately, the experimental session was marred by the necessity that it be conducted at a public computing facility with networked applications. Inexplicably, at least on this occasion, the system runs one Windows application at a time, and to use another it is necessary to restart Windows. Therefore it was not feasible to use the Clipboard for this task. PC3 saved his graph to a file. Then he tried to import his previously newly created graph into a Microsoft Word for Windows document and was unable to do so. He stated in explanation that he was accustomed to using Word 2.0 for Windows, which has a better Insert feature, and that he doesn't like Excel graphs.

In summary, despite his fine performance on questions related to File Format, he appeared to be unaware in general of the existence of specialized procedures for importing files in various file formats when using DOS applications. He was, however, aware of the use of the Insert Picture function in Word for Windows, and of "a neat embedding feature" in WordPerfect, though he was unable to apply either successfully on the occasion of the test. The evidence of the Task Checklist suggests that these examples do not reflect any lack of necessity for PC3 to meet problems related to file format in practice.

Subject B1

B1 showed, in separate instances, limitations in her operational knowledge related to concepts with which she is familiar, knowledge of particular procedures without apparently understanding the relevant aspect of a related concept, and ability to accomplish tasks in ways that she had failed to describe in answer to previous questions. That is, she seems to display both sides of the dissociation between conceptual/declarative and operational knowledge.

Her grasp of Directory/Folder is somewhat problematic. She defined the concept as "Different areas..." and seems able to use pre-existing directories for organization, for example adding that she would put all data files for Systat in the Systat directory. She indicated on the task Checklist that she seldom performs routine operations directly involving directories, and as a matter of fact was unable to create a new directory or folder for the performance tasks. Further, she did not suggest creating a directory in order to avoid confusion with files related to other programs for question 9, despite having said that she would put all data files for Systat in the Systat directory. Nor did she reply that names without extensions in DOS might indicate directories in DOS. She thought that folders might be like data files on the Macintosh, but said "I would have to sit in front of the screen to be sure." Perhaps this was correct, since she was able to open an Excel folder in order to find and open Excel for the Macintosh performance tasks. Despite the limitations of her ability to manipulate directories, Directory/Folder is a central concept in her pathfinder network, with links to Data File, File, File Format, Finder, Filename Extension, and Program.

Operational Knowledge Without Conceptual Knowledge

Subject B1. B1 seems to have some understanding of the concept of the Storage Buffer/Clipboard, but may think of it as a feature of particular applications, and may think of it as limited to text. On DOS question 11 regarding copying or moving text within an application she mentioned blocking text and moving or copying to multiple locations, then as though discussing a completely separate concept, mentioned a "Scratchpad type thing in Word on the Apple...Mac or Windows sticks it off on a Clipboard that you can access." When asked about bringing a graphic into a word processor document (question 12) she mentioned only importing and dealing with possible difficulties and a necessity to "decode" if the graphic was created in another package. Even on the Macintosh questions she did not indicate that it might be possible to accomplish this using the Clipboard. However, She did just this in the DOS and Macintosh performance tasks.

B1 used Windows to run DOS programs for the DOS performance tasks. Unable to complete the tasks using DOS programs, she did so using Windows and the Clipboard, cutting and pasting a chart created in Excel into a Word document without hesitation. B1 was able to complete the Macintosh tasks once given a hint to open a file of unknown format using an application, but hesitated before trying to cut and paste, saying "The only way I know how to do that .." and pausing to think for some time. This may however have been due to her supposing that both applications must be open to paste between them. She closed Excel prior to opening MS Word, remarking that she doesn't know how to open multiple applications at one time.

Subject M1. M1's grasp of File Format appears opposite that shown by PC3. M1 rated himself unfamiliar with File Format, but is able to accomplish some related tasks.

M1 rated himself unfamiliar with the concept of File Format, and his performance on other tasks showed no evidence of understanding of this concept. His Pathfinder network shows links between File Format and only two concepts, File and Data File, but suggests at least some pragmatic understanding of the consequences of file format in a link between Data File and ASCII. He did not think of saving documents in other formats to facilitate use with another program (question 14). Furthermore, it did not occur to him that a change of suffix (DOS) or of icon (Macintosh) might indicate that a file had been changed and might no longer be able to be opened by a particular program (question 13).

Nonetheless, M1 apparently is able to perform some tasks related to the concept of File Format. According to his ratings on the Task Checklist he very seldom saved a document in a different format, but he fairly often inserted into a document something that originated in another program, which he found very easy. He was aware that many programs have translators that enable them to open files created by other programs (question 14). When called upon to open a file of unidentified origin, M1 elected to run Microsoft Word "just for fun", as he said, and successfully used this program to open the file.

Subject M4. Another Macintosh user, M4, seemed to possess the least conceptual knowledge among the subjects in this study, but was able to carry out most of the Macintosh performance tasks. M4 rated herself unfamiliar with Creator of a File, Filename Extension, Data File, Storage Buffer/Clipboard, Operating System, Finder, Command Interpreter, ASCII, and Memory (disk and RAM).

Despite her long experience with the Macintosh, M4 is "Just beginning to learn about the Clipboard and Scrapbook" as she stated in the initial interview. In her Pathfinder network, there are no links between Storage Buffer/Clipboard and any other concept, hence she evidently rated it as unrelated to all other concepts. Her use of the Scrapbook and Clipboard provides a notable example of performance without conceptual knowledge. she knows how to highlight, which she calls darken, and copy and paste within applications, without apparently realizing that this involves the Clipboard (question 11). She uses the Scrapbook to copy between applications (question 12) and is also able to copy and paste from the Scrapbook without realizing that the Clipboard is involved. She did just this in the Macintosh performance tasks. M4 stated that she always uses a sequence of copy, paste, then delete instead of cut and paste, "In case of mistake."

M4's lack of conceptual knowledge may be attributable to satisficing by a Macintosh user, encouraged by the system's ease of use. However, some of the PC users in this study also exhibited the ability to perform tasks related to concepts that they did not understand.

<u>Subject PC1.</u> PC1 has about 6 years of computer experience. His first experience was with an IBM PC, and he appears to be one of the most DOS-oriented subjects from this study.

Although he seems to have an excellent understanding of the other target concepts, PC1 does not appear to possess the concept of Storage Buffer/Clipboard. His response on the Concept Definition task was confused with the DOS Buffers command, which was further confused with the Files command. None of his responses from the DOS questions indicated any abstract appreciation of this concept. In his Pathfinder network, Storage Buffer/Clipboard is linked to Program/Application and to Memory (Disk and RAM), which are fairly reasonable associations for the DOS Buffers command, but would also be plausible links for a DOS user with some understanding of the concept. In fact the link to Program?Application is the only link to Storage Buffer/Clipboard in the average network for DOS users.

He was able to describe some procedures related to a storage buffer, for example using Word Perfect's Move and Copy functions, in response to DOS question 11. But when asked why the Move and Copy functions work, he could only give a procedure-oriented response, "It works because you're blocking it in." He seemed unaware that the last item remains available in the Storage Buffer, and may think of this function strictly with regard to text, as he indicated that he would move a graphic within Word Perfect by deleting it and then re-importing it.

Subject PC6. Another DOS user with little or no GUI experience, PC6 also appears to have no understanding of the concept of Storage Buffer/Clipboard, in fact in his Pathfinder network there are no links to this concept, but he has at least a limited ability to accomplish related tasks.

PC6 rated himself unfamiliar with Storage Buffer/Clipboard on the Concept Definition task was . As was the case for PC1, none of his responses from other tasks indicated understanding of this concept. However, according to his responses on the Task Checklist, he occasionally inserts into a document something that originated in another program, finding it somewhat difficult/somewhat easy. He is also aware of procedures in Word Perfect for blocking, moving, and copying (question 11). In fact he recalled the appropriate function keys as he did also for importing a graphic (question 12). But he specifically stated that he does not know how the Move and Copy functions work.

Summary: Declarative/Conceptual vs. Operational Knowledge

Several theories of knowledge representation, e.g., those of Anderson (1982) and Tennyson and Cocchiarella (1986), argue that declarative or conceptual knowledge precedes development of procedural knowledge. Similarly, Polson and Lewis's CE+ theory (1990) argues that specific forms of causal inference precede procedural knowledge.

In the course of the present study several subjects, M1, M4, PC1, PC6, and B1, were able to carry out performance tasks without showing signs of having had any understanding of the relevant concepts. While it must be conceded that the nature of the study does not allow characterization of subjects' performance as procedural in Anderson's or Tennyson's sense, or as skill-based, rule-based, or knowledge-based in Rasmussen's senses, the performance of these subjects does suggest that conceptual knowledge need not precede knowledge of procedures. At the same time the opposite side of a dissociation between conceptual or declarative and operational knowledge was apparent in the performance of PC3 and B1 who evidently have knowledge that they are unable to apply.

Theoretically interesting as the dissociation may be, it is important to note that there may be other perspectives upon and explanations of some of the same data. For example, it has been convenient to consider particular concepts and related operations in isolation from other concepts and operations, but obviously such isolation is exceptional in realistic situations. Certainly some concepts have others as pre-requisite, and the same is even more clearly true of operations. An example from the present study may be B1's hesitation in attempting to use the Clipboard to transfer a chart between applications because she was of the opinion that this operation requires having applications open simultaneously, and she did not know how to do so on the Macintosh. This constitutes, at least in part, a serial relation between concepts and/or operations, but in many cases a parallel or reciprocally interactive development is likely, with improvement in one concept or operation occurring more or less simultaneously with that of others. A speculative example using target concepts from the present study could involve simultaneous development of the concepts of Program vs Document and File Format.

Another perspective upon data from the present study might be drawn from the memory literature. Perhaps, to use Tulving and Pearlstone's (1966) terms, some of the knowledge that seemed to be unavailable in this study was, rather, inaccessible, and might have been found by different tests, as suggested by the idea of transfer appropriate processing (Morris, Bransford, and Franks, 1977), or in a different context or "cognitive environment" (Tulving and Thompson, 1971).

<u>Context Effects</u>

It is, of course not to be supposed that any one test demonstrates the extent of a subject's knowledge. The likelihood that some forms of knowledge are not verbalizable is one reason for this fact, but there are others. Clearly some knowledge, including knowledge of procedures, may be more easily retrievable in some circumstances than others.

A possible example of context dependence in access to concepts is provided by M1's responses on the DOS and Macintosh versions of question 9. As it happened, M1 was administered the DOS version of these questions first, and was originally unable to answer question 9. He was later able to propose in response to the Macintosh version of this question that a new program should be put into its own folder in order to avoid confusing files related to that program with others, and to make the analogous suggestion for DOS in retrospect, at that point. To be precise, he stated "I'm sure the same goes for DOS." Perhaps M1 has readier access to the relevant concepts in the context of his preferred system, the Macintosh. Perhaps he was only able to produce this idea when given two opportunities. Another class of explanation for this event is discussed below under the topic of heuristics. The suggestion of difficulty in accessing a relatively familiar concept even in "transfer" to a relatively familiar, though less often used, system, is interesting. Perhaps envisioning attempting to accomplish the same goal with a more familiar system could sometimes be an effective aid to transfer.

Subject B1 on several occasions showed what might be context effects in recall of particular concepts or operations. In response to Macintosh question one she said that "You click on the disk and get little icons and folders. I don't know, I'd have to sit and play." Shortly later, in response to question 5, she thought that "folders are like data files, I think. I don't know...I'd have to sit in front of the screen." This suggestion of reliance on cues from the screen is reminiscent of the display-based competence described by Mayes et al (1988) in MacWrite users. On both the DOS and Macintosh versions of question 12, regarding inserting a graphic into a word processor document, B1, as noted previously, described importing procedures that are available in particular packages for reading files of particular formats, and did not refer to use of the Clipboard, which she had recommended on the previous question for moving and copying text within an application. Yet she used the Clipboard to put a Microsoft Excel chart into a Microsoft Word document in order to accomplish one of the performance tasks within the Windows environment. Is this an example of a dissociation between verbalizable conceptual knowledge and non-verbalizable procedural knowledge, an example of knowledge that is more accessible in the presence of particular cues, or an example of problem-solving behavior by the subject?

An example of failure to apply knowledge verbalized in other situations is the failure of most subjects to consider that a change of filename extension or icon might be associated with a change of file format in answer to question 13. This was true for several subjects who had evinced a fairly good understanding of file format and its relation to filename extension or icon. For example, M3 did not mention this possibility on question 13, but defined Filename Extension in this way: " A three-character addition to a DOS file's name. Often describes how the file is used, or what application created the file." and File Format in this way: "Describes whether the file is executable, text, data, etc. Characterized by the extension on DOS machines or the icon on a Mac." In M3's Pathfinder network, Filename Extension is linked to Creator of a File, Data File, Directory, File, and File Format, further indicating an understanding the relation between Filename Extension and File Format. Perhaps M3, and others, can have such an appreciation of the relations among these concepts without realizing that it implies internal differences in files, or perhaps it is simply somehow difficult to utilize the relevant knowledge in this case. As it happens, only PC users, subjects PC2, PC3, PC4, and B1, who uses both DOS and Macintosh, brought up this possibility in response to question 13.

Effects of Computer Preference upon User Knowledge

Failure to apply in a given context knowledge that has been demonstrated in another may mean that knowledge has been incorrectly or incompletely acquired. In the previous section M1's initial failure to suggest creating a new directory in response to DOS question 9 was taken as a possible example of context effects upon access to relevant concepts. It is at least possible, however, that his retrospective suggestion to do so after making the analogous suggestion for the Macintosh represented a new inference regarding DOS directories for M1, perhaps because the question-answering process led him to contemplate the relationship between directories and folders more completely than he had in the past. It is worth noting that M3 stated that "Folders on the Macintosh let me understand directories on the PC."

Effects upon Knowledge of Concepts and Operations

For the present purposes, probably the most dramatic possible effect of interface type would be systematic differences in the knowledge acquired by users of the different interfaces. There is some evidence from this study consistent with just such effects, though the varied interface experience of the two groups of subjects renders it impossible to draw firm conclusions upon this point.

Prior to the study, it was hypothesized that two countervailing tendencies might affect Mac users' acquisition of conceptual knowledge. The general ease of use of the Macintosh, and the ready availability of information in the GUI, along with the consistency of the interfaces of different applications, should make conceptual knowledge easier to acquire than would a command-based interface, and more valuable once acquired since related procedures could be readily inferred in a variety of situations. But the same factors make it easy to accomplish basic tasks with minimal knowledge, and this fact may lead to satisficing by some Macintosh users. In the present study, there is some indication that both tendencies may have been in operation, at least with respect to the concept of Storage Buffer/Clipboard.

Kay and Black (1985) found in hierarchical cluster analysis that even expert users of a text editor lacked a buffer-oriented cluster, and speculated that a different implementation might make the buffer more obvious to users. Consistent with this idea, Payne et al (1990), in comparing users of MacWrite and an IBM text editor, concluded that the concept of buffer is not readily induced, and that users of the IBM editor did not arrive at this concept. A priori, the Clipboard, which is an important adjunct of the Macintosh operating system, might also be more salient to Macintosh users than more or less equivalent functions of particular DOS applications. Accordingly, in the present study, the Storage Buffer/Clipboard concept was considered a particularly promising candidate to produce differences in the knowledge of Macintosh and DOS users. There is anecdotal evidence and statistical indications consistent with this idea.

Composite measures of subjects' understanding of the target concepts and related operations were constructed on the basis of evidence from all the tasks performed in this study. Individual and group mean scores on these measures can be seen in Table 3. For each concept, scores for understanding of the concept are in a column labeled "C", while the columns labeled "O' contain scores for grasp of operations related to the target concept.

It may be seen from the table that the greatest differences between groups occur on scores related to Storage Buffer/Clipboard. On the measure of conceptual knowledge, the mean for Macintosh users ($\underline{M}$ = 1.50) was greater than that for DOS users ($\underline{M}$=.75), but this difference only somewhat approached statistical significance, $\underline{F}(1,10)=3.14$, $\underline{p} < .1068$. As it happened, the smaller difference between means for the Macintosh group (M=1.50) and the DOS group (M=.83) on a composite measure of operational knowledge related to use of the Storage Buffer/Clipboard did appear statistically significant, $F(1,10) = 5.00$, $p < .049$.

These differences may be better understood by considering some of the details of the understanding of these concepts and operations evinced by individual users. One Macintosh user, M4, and two DOS users, PC1 and PC6, appear not to have any understanding of the Storage Buffer/Clipboard concept, though all three were able to describe procedures that use a storage buffer. A third DOS user, PC4, was also able to describe operations involving a storage buffer, and showed only very slight evidence of understanding the concept. PC1, PC4, and PC6 are perhaps the most exclusively DOS users among the subjects in this study, in that none has had much exposure to Microsoft Windows and, as might be expected, their descriptions of cut, copy, move, and paste operations are restricted to procedures within applications.

M4 might be described as the most strictly operationally oriented of the Macintosh users, with very little conceptual knowledge related to the target concepts. Giving M4 a score of zero for knowledge of the Storage Buffer/Clipboard is possibly over-conservative, since she is aware of the existence of something called the Clipboard, and that it is related to the Scrapbook. She stated in the preliminary interview that she was just learning the Clipboard and Scrapbook, but later rated herself unfamiliar with Storage Buffer/Clipboard and did not attempt to define the term. She was able to use the Clipboard without knowing that she did so, to cut, copy and paste within an application or to the Scrapbook, which she used to copy and paste between applications.

In a similarly conservative vein, PC4 was given a score of 1 for knowledge of the Storage Buffer/Clipboard concept on the basis of his definition, "Clipboard concept is where information is stored in a different location than the program.", despite the fact that he referred to the concept in none of his answers to the DOS questions and rated himself only very slightly familiar with the concept. In PC4's Pathfinder network, Storage Buffer/Clipboard is linked only to directory, suggesting a misunderstanding of the kind of storage involved in the concept, conceivably confusion with the directory used for temporary files, or even that his definition was based upon surface features, i.e. the presence of the word "storage".

While M4 is aware of the existence of something called a Clipboard, PC1, who otherwise seems to have a very good grasp of the target concepts, appears not to have encountered such a term. He rated himself very familiar with Storage Buffer/Clipboard, but appeared to have confused it with the Buffers command in the DOS config.sys file. In response to DOS question 11, regarding moving text within a word processor document, PC1 described a process of selecting, cutting, and pasting, but was unable to explain why the process worked, beyond "It works because you're blocking it in." He also appeared to be unaware that the last item copied or cut would remain in the storage buffer, allowing the same text to be pasted into another location without recopying.

PC6 indicated on his familiarity ratings that he is unfamiliar with Storage Buffer/Clipboard, and this is also evident from the disconnection of this concept on his Pathfinder network. However, he had earlier described procedures in Word Perfect for blocking, moving, and copying (question 11). In fact he recalled the appropriate function keys, but admitted that he did not know how it works.

The fact that M4, PC1, and PC6 are all relatively exclusive users of their particular systems might seem to suggest that Storage

Buffer/Clipboard is a concept better developed after exposure to multiple system types, but in fact it may simply be that M4 is the exception among the Macintosh users. Of the other Mac users, all but M2 scored a 2 on knowledge of the Storage Buffer /Clipboard concept, but no PC user, with the exception of PC3, who was originally a Macintosh user, scored above a one. The clear difference between M4 and the other Mac users may indicate the operation of the two counterposed interface effects upon concept acquisition noted above. This one Macintosh user, M4, has apparently not troubled to acquire much conceptual knowledge, though she is able to accomplish what she needs to do, with some limitations, seemingly a good illustration of satisficing. That is to suggest, she has not acquired the concept because she does not need to do so. On the other hand, speculatively, at least, the PC users have not acquired this concept because it is difficult to do so for a DOS user.

From the above discussion, it should be evident that the same countervailing tendencies could affect mastery of operations on the Macintosh, with general ease of use and relatively easy generalization to new situations favoring learning, while also allowing satisficing. In this study overall differences between groups appeared only for mastery of operations related to Storage Buffer/Clipboard, with DOS users, even PC3 with his Macintosh background, consistently scoring a one on the composite measure of this form of knowledge, while three of the Mac users scored a two on this measure. Examination of statistics and of individual subjects has shown something about the content of users' knowledge of particular concepts and related operations, but what can be determined about the organization of users' knowledge?

Group Mean Pathfinder Networks

One available method for attempting to find effects upon knowledge, and specifically upon knowledge organization, is examination of group average Pathfinder networks, here for PC users as opposed to Macintosh users (see Figures 1 and 2). Pathfinder networks for all subjects are to found in Appendix A. The C statistic for the similarity of the group average networks is rather low, $\underline{C}$ = .275.

Several fairly natural differences are noticeable in the group average networks. All PC users, and B1 also, rated themselves unfamiliar with Finder, a Macintosh concept, on the Concept Definition task, and in the PC users' network Finder is linked only to Data File, probably because PC users imagined that it was a utility for finding files. Macintosh users, on the other hand, generally confused Finder with Operating System, and in the Mac users' network Finder is linked only to Operating System. In two Mac users' networks Finder was linked with Storage Buffer/Clipboard, and had Finder not been confused with Operating system there might have been a link between Storage Buffer/ Clipboard in the Mac users' group average network. Another Macintosh concept, Creator of a File, is linked in the PC users' network to File, probably as a result of a natural language interpretation of the term, while in the Mac users' network Creator is linked to Program/Application: the Macintosh term Creator of a File refers to the program that created the file, and apparently Macintosh users know it. Storage Buffer/Clipboard is a more salient concept on the Macintosh than in DOS, though not Windows, since it is a service of the Operating System, not of particular applications. It is directly linked in the Mac users' network only to memory, but as noted this may be due to Mac users' confusion of Finder and Operating System. In the PC users' network Storage Buffer/ Clipboard is linked, as might be expected, to Program. Here is a clear case where different operating systems seem to have supported differing understandings of a concept and of the relation of that concept to others.

There are also several differences between the networks with regard to concepts related to File and File Format. Interestingly, File Format is a fairly central concept for Mac users, linked to File, Data File, and Program. PC users, three of whom rated themselves unfamiliar with File Format, link this concept only to File and to ASCII. Somewhat curiously, PC users' network links Filename Extension only to Data File, while in the Mac users' network it is linked to the more general concept, File. Mac users, on the other hand, link ASCII only to Data File, while PC users link it to Operating System. It appears that for PC users ASCII is somehow linked to the operation of the computer, while for Mac users it is rather peripheral, just one form of File Format, likely for a data file. Conceivably PC users' link between ASCII and Operating System is the result of the importance of two text files, the Config.sys and Autoexec.bat files, for the operation of the PC. At all events, the group mean network differences suggest that Macintosh users have a clearer conception of File Format and of its relations to other concepts

It is also somewhat curious that Memory (Disk and RAM) is linked only to File and Data File in the PC users' network, as though they are considering only the disk aspect of the concept. It might have been expected that RAM would be particularly salient for DOS users, and that memory would perhaps be linked to Operating system and Program/Application, because of the 640 kilobyte and 1 megabyte limits imposed by DOS and the resulting workarounds to make more memory available for programs, ROM shadowing, and device drivers. Of course, Macintosh users sometimes also face memory limitations, and this could be one reason that Memory (Disk and Ram) is linked to Program/Application in the Macintosh users' network, along with Storage Buffer/Clipboard and Initialize/Format a Disk. All this suggests that for these subjects, Macintosh users as a group are also more completely aware of the meaning and relations of Memory (Disk and RAM) than are PC users.

Summary: Effects of Interface upon User Knowledge

This study has provided a number of indications consistent with the hypothesis that interfaces differ in the extent to which they support learning of concepts and procedures, with consequent systematic differences in the knowledge and knowledge organization of users of the differing interfaces.

Based upon the composite measures of conceptual and operational knowledge, Macintosh users as a group appear to have a better grasp of procedures related to the Storage Buffer/Clipboard than do PC users, and there is some indication that they also understand the concept better than do PC users. Examination of the responses of individual users from the different groups is consistent with this pattern of results. While three of the Macintosh users have a very good grasp of the concept, none of the PC users do, and three PC users appear to lack the concept of Storage Buffer/Clipboard completely, though they are acquainted with related procedures. And PC users show signs of considering the Clipboard to be related to particular applications, even in Windows.

This latter idea is consistent with the group average Pathfinder network for PC users, in which Storage Buffer/Clipboard is linked only to Program/Application. The group average networks suggest other differences between groups that were not revealed by examination of composite scores for knowledge of concepts and related operations. On the basis of these networks, Macintosh users seem to have more complete understandings of the concepts of File Format and of Memory (Disk and RAM) than do PC users. In short, while no definite conclusions can be drawn, in this study there is the appearance that Macintosh users have a better understanding of the concepts examined than do PC users.

Whether or not this is so, how do users of the different interfaces learn concepts and procedures? The present study sheds little direct light on this subject, apart from question 15, which asked subjects how they proceed when faced with computer problems. All Macintosh users mentioned exploring, while PC users were more likely to look in the manual or ask, or as one subject mentioned, pray. But while this study does not truly examine learning, it allows a certain amount of speculation regarding the heuristics or rules of thumb employed by subjects in unfamiliar situations, which may result in learning.

Subjects' Use of Heuristics

Subjects in this study often exhibited behaviors that might be the result of use of heuristics in more or less unfamiliar situations. Polson and Lewis (1990) restricted their discussion of use of heuristics in problem-solving to one heuristic, label-following, selecting options that seem to resemble the desired goal. Previously it was argued here that users are very likely also employ other methods in problem solving, and that label-following must in any case be dependent upon other forms of knowledge not necessarily addressed in CE+, Polson and Lewis's theoretical framework.

It must be admitted that in a number of instances in the present study when subjects were at a loss for a continuation in the Performance Tasks they resorted to searching through menu structures, and apparently selected items bearing some perceived relation to the intended goal. For example, M4, when called upon to create a new folder in the Macintosh performance tasks, did not now how to proceed, but quickly found the New Folder option under the File Menu of the Macintosh Finder. But subjects' performance in some cases seemed to illustrate the dependence of label-following upon prior knowledge, in particular conceptual knowledge, a factor that was not taken into account in CE+. Consider what occurred when subjects who were relatively unfamiliar with a particular application attempted to open a text file using that program. To pick a straightforward example, subjects using Lotus 1-2-3 invariably displayed the first level of the menu structure and selected File. Some subjects, who apparently had a better grasp of the concept of File Format, then selected Import, while others chose the Retrieve option. These latter subjects were frustrated in their attempt to open the file, but often repeated the attempt several times, perhaps in the supposition that they might have made some slight error in the procedure. The main point is that label following behavior is very clearly dependent upon prior knowledge in this example.

It is possible at least to speculate regarding other heuristics employed by subjects in this study. Heuristics may be regarded as methods of generating hypotheses, which may then be tested. Clearly this is effectively true of label following. In some cases, the solution generated by use of a heuristic may seem so clearly correct that the user will not consider testing necessary. It might be argued that analogical mapping as a method of problem solving is itself a heuristic. The intention here is to argue that heuristics, such as analogy, may be applied to inference regarding concepts as well as to finding procedures. A speculative example might be M1's retrospective suggestion of creating a new DOS directory in response to Macintosh question 9. M1 suggested putting a new application in its own folder or directory, and then said he was "...sure it goes for IBM too". It was previously suggested that this event could show context effects upon accessibility of knowledge, and it is difficult to deny the likelihood of context effects, but there are other possible effects at work in this example. One possibility is that M1 had a better understanding of Macintosh folders than of DOS directories, and in the course of answering the question suddenly realized, or realized more clearly, the relation between directories and folders or applied some of his knowledge regarding folders to DOS directories for the first time. This would certainly be consistent with the remark of another subject, M3, that "Folders, etc. helped me to understand directories and subdirectories." Unfortunately, Concepts common to DOS and the Macintosh for which there exist platform-specific names, for example Directory/Folder, were presented in the Concept Definition task of this study in a fashion that might have encouraged the realization of the relation between the terms. It is likely that some users with experience of both platforms do not fully understand the relation of these terms. While some of the tasks in the present study, notably the DOS and Macintosh questions, which were administered prior to the Concept Definition Task, were intended to discover such differences, it might be desirable to present the terms separately in another study. In the present study this procedure might have made it possible to determine, for example, whether M1's inability to think of creating a subdirectory in the context of DOS was a failure of retrieval or an indication of incomplete understanding.

If this last example shows a heuristic in action, how should such a heuristic be characterized? I would argue, without wishing to insist, that this might be an example of hypothesis generation by extension of a known concept or procedure, and that such an extension can be attempted on a number of bases. In this example an extension could have been based upon analogy with something known to be true of a very closely related concept. Note that M1 may have suddenly appreciated the extent of that relation, or suddenly achieved an analogical mapping based perhaps upon a mapping of relations between the "domains" of directory and folder. If this is so, it may well have been that the overall context of the DOS and Macintosh questions, in which in most cases identical or very directly analogous questions were asked for each system, assisted M1 in understanding the relation between directory and folder.

Analogy is also likely to play a role in attempts to apply familiar ideas and procedures to new, but related tasks. B1's use of the Clipboard to move a graphic from Excel into a Microsoft Word document could be an example of this process. B1's responses on the DOS and Macintosh questions indicated clearly that she thought of the Clipboard solely with respect to moving, cutting, or copying text. In response to Macintosh question 12, about putting a graphic into a word processor document, she stated that she thought it was the same situation as for DOS. When called upon to accomplish this task in the performance tasks she said "The only way I know to do that.." rather unhappily, and fell silent for a short time, then used the Clipboard to copy the graphic. It seems very reasonable to interpret this sequence in terms of an extension heuristic, but it must be admitted that it is possible that B1 had had previous intimations that the Clipboard could be used for this task.

Misconceptions, Meta-knowledge, and/or Superstitions

While users may often succeed in discovering correct ideas and methods by inference from or application of heuristics to whatever information is available, it must be the case that subjects also often arrive at incorrect ideas or methods, and may or may not be aware of the limitations of their knowledge. A systematic understanding of what errors are likely in use of a particular interface could prove very valuable. The present study affords glimpses of some of the subjects' misconceptions or superstitions without allowing more than speculation regarding their origin.

Some misconceptions were manifested by only one subject, but it is possible to speculate regarding their origins. The case of PC1's confusion between Storage Buffer/Clipboard and the Buffers command from the DOS config.sys file seems mediated by natural language, and it is likely that misunderstandings often arise in this way. Most subjects, for example, interpreted the Macintosh "Application busy or missing" message very literally, seemingly coming to an interpretation when asked to do so, without showing knowledge of less superficially plausible possibilities.

M2, in answer to DOS question 5 regarding the meaning of filename extensions, remarked that ".bat is like a driver." Perhaps she knew, probably from experience, that .bat files, which are batch files, can be related to making things happen without themselves being programs.

M2 and M4 each had an interpretation of the Macintosh "Application busy or missing" message that is not clearly related to a superficial, natural language interpretation. M2 supposed that the application was "already opened and you're trying to open it again," seemingly a natural language interpretation, but added, "I've gotten uninterpretable messages before. I've been told it's better to open the application first, then the file. I think I've gotten it when I was trying to open the file, not the application." M2's reaction is close to that appropriate when the error message results from attempting to open a file for which the creator is not present. In combination with her response to Macintosh question 6, "I have double-clicked on a sheet of paper with a bent edge, and opened a Word file. I didn't have to open Word to pull up the document, it automatically opened Word.", this suggests that she has not realized that this error message can occur when the creator of a document is not available, or that Word is not in general the creator, or in some sense the appropriate application, for text files. Rather, she seems to have accepted a concocted explanation for, or rule to apply in case of, such events, imagining that running the application first is somehow more reliable than double-clicking a document in order to open it. This explanation is, practically speaking, true, and it may be unlikely that M2 will learn to discriminate those occasions when it is necessary to open the application first in order to open a file. Given the ease of double-clicking a file to open it, perhaps she will simply continue to do so, and if she receives an error message, then try opening the application first. This expectation is consistent with her behavior during the Macintosh Performance Tasks. She attempted to open an unknown file by double clicking. This resulted in the "Application Busy or Missing" error message, and she immediately and successfully tried to open the file using Microsoft Word. When later specifically requested to open a prepared text file to create a Word document, she again first tried double-clicking it, then opened it from within Word.

M2 referred to the error message in passing as "uninterpretable." M4's reaction to Macintosh question 2 suggests that she, too, finds this message uninterpretable, or that she considers it a possible result of a more general inability to run the appropriate program. Her suggestion was that too much else was open, "At times you have to close everything and try again." M4 may have rules or heuristics for dealing with problem situations, and she may apply them in an overgeneral way. This might be considered to be an instance of application of an extension heuristic, as previously discussed. But it may equally well simply be that M4 is unable to discriminate between the conditions in which a program fails to load. Such attempts can, of course, sometimes solve new problems and/or result in learning, and may be part of an important learning mechanism.

The ideas of M2 and M4 mentioned above were apparently unique to them. Several misconceptions seemed widespread among the subjects in this study, and some were seemingly unrelated to subjects' computer preference. A good example is the tendency for users with experience of a GUI, either Windows or the Macintosh, to expect unnecessary limitations in the use of the Clipboard to transfer information between applications. Several subjects, including PC2, PC5, M1, M3, and M4, had what seem to be related uncertainties regarding this process. It might be imagined that this uncertainty could be the result of experience with DOS , where storage buffers are features of particular applications. Even those DOS users familiar with Windows were uncertain on this point, with the exception of PC3 whose original experience was with the Macintosh. PC2 thought it might be necessary to have both applications running at the same time in order to accomplish this task, while PC5 was sure that this is necessary.

M1 and M4, on the other hand, use the Scrapbook to copy data between applications, though M1 described the use of the Clipboard for this purpose in response to question 12. M3 advocated using the Clipboard for this purpose with MultiFinder available and both programs running, but thought that if MultiFinder is not available it would be "advisable" to use the Scrapbook for this purpose. All of these subjects seem to operate on the supposition that the Clipboard will operate more effectively, or only operate, if all programs involved are running. It could be that PC2 and PC5 make this supposition as the result of some sort of extension of an original understanding of storage buffers as implemented by particular programs, as is the case for DOS. The Mac users, M1, M3, and M4, may have an idea that the Clipboard and Scrapbook are intended for complementary purposes, the Clipboard for use within a program and the Scrapbook for use between programs. But this would not explain the fact that M1 and M3 apparently know that the Clipboard can be used for transfer between programs.

One possible origin for some of the ideas and practices of these subjects is caution. The Scrapbook is a more permanent form of storage than the Clipboard, hence information stored there is less susceptible to accidental loss. Subjects demonstrate such caution in other ways. One example is the tendency of DOS users to use the DIR command to view the consequences of their commands. This may be due to the lack of, or incompleteness of, automatic feedback in DOS and the ease of mistyping commands. In another case where there is incomplete feedback, some Macintosh users who understand the use of cut and paste commands tend to copy, paste, and then cut. M4 always does this "in case of mistake", as she said. M2 originally used this sequence because, she said, she "used to be scared, I wasn't sure where it went, then I found out it went to the Clipboard and it was OK." M2's experience suggest that the accumulation of conceptual knowledge can assist in eliminating unnecessary behaviors. This rather simple example illustrates an interaction between conceptual and operational knowledge with, in this case, operational knowledge preceding conceptual but later being modified by application of the relevant concept. While it is not at all clear in this example, it may also be that M2's acquisition of the Storage Buffer/Clipboard concept was motivated by or to some extent the consequence of her knowledge of related operations. If users sometimes develop conceptual knowledge following operational knowledge, as found by Peterson et al (1990), this is likely at least some of the time to be a consequence of deliberate investigation by the user. This is one way in which general ease of use, in terms of ease of acquiring procedures, could facilitate acquisition of concepts. It also must be the case that the ability to learn and use procedures readily increases the likelihood of correct feedback in hypothesis testing, whether related to other procedures or to constructs.

In this study subjects' expressions of confidence regarding the validity of their ideas and the extent of their knowledge often did not seem related in any very simple way to the experimenter's assessments of the same knowledge. Subjects ratings of familiarity with various computer systems may have been, at least in some cases, based upon relative familiarity rather than any attempt to assess the extent of mastery of a system relative to what is possible, or of course, simply mistaken. For example, M4 rated herself very familiar (five out of five) with the Macintosh, though she was unfamiliar with nine out of fourteen of the concepts on the Concept Definition task. M1 also rated himself very familiar with the Macintosh, but later in the study acknowledged that his needs had been simple, and guessed that he wasn't a very sophisticated user. Doubtless M4's computer needs have also been simple. Both subjects have probably not previously been exposed to circumstances that highlight their areas of ignorance. The general ease of use of the Macintosh may well be a factor tending to produce overestimation of familiarity along with satisficing. These tendencies may help account for the failure to find very interesting statistical relationships between subjects' ratings of systems familiarity and other variables.

Subjects were also sometimes quite confident of their misconceptions. For example, PC1 rated himself very familiar (five out of five) with Storage Buffer/Clipboard, though, as discussed above, he apparently had the Buffers command from the DOS config.sys file in mind. But it is also true that subjects often expressed uncertainties, as for example PC2 did with regard to the necessity to have both programs running in order to cut and paste between them using the clipboard. Perhaps in this case PC2 actually had multiple sources of related information or inferences, and his response was a compromise between them.

The greater a user's general knowledge, the more likely it may be that the user will realize the extent to which it is limited in specific instances. This supposition is consistent with some of the responses of PC2 on the Concept Definition task and DOS questions. For example, PC2 initially rated himself very familiar (five out of five) with Initialize/Format a Disk, then changed his rating to 3. He admitted that he knows how to do it, but not what occurs. Speculatively, at least, it seems likely that the technically oriented PC2 had in mind a deeper form of ignorance than would have occurred to many of the other subjects. As an example, M2 rated herself very familiar with this concept, but defined it only as "to prepare it for storage." Unsurprisingly, then, there may be a positive correlation between general knowledge and meta-knowledge.

Users are prone to misconceptions, and this study has produced examples of both idiosyncratic, individual errors and misconceptions that appear to be common among users. The cases examined seem to be plausibly explicable, but it would be rather difficult to arrive at a few simple principles with which to predict them.

Relation of Results to Existing Theory

Prediction, of observed misconceptions and of other phenomena, is one criterion, a very important one, by which to judge theory. A theory may relate to specific phenomena in several ways. It may deal with the phenomena in a very general way, it may ignore the phenomenon but prove either compatible or incompatible with it, or the theory may allow specific predictions that prove either consistent or inconsistent with observation.

The results of this study are difficult to discuss in terms of existing theories. Some theories, such as GOMS (Card, Moran, & Newell, 1980, 1983), and to a slightly lesser extent CCT (e.g., Kieras and Polson, 1985) are very restricted in their areas of applicability, while others, such as Moray's lattice theory (Moray, 1987) or YSS (Payne, Squibb, & Howes, 1990), and the skill-rule-knowledge framework of Rasmussen (e.g., 1983), are so general as to be compatible with almost any evidence, but do not specifically address, for example, contexts effect or the origins of subjects' misconceptions. CE+ (Polson and Lewis, 1990) attempts to be broadly applicable, at least as a model of a lowest-common-denominator user, and could produce predictions of at any rate a range of possibilities for subjects' behaviors and knowledge, but much of the behavior of subjects in this study bears no apparent relation to the theory. In general, HCI theories ignore such phenomena as context effects and retrieval processes, and the limitations in understanding that might mimic such effects. Knowledge representation and organization is treated in a general, if somewhat limited way by several theories or frameworks, such as Rasmussen's (e.g., 1983) hierarchical framework and Moray's (e.g., 1987) lattice theory, but specific predictions are difficult to achieve. The same is true of the content of users' knowledge, particularly conceptual content, hence existing HCI theory also cannot readily describe origins for or predict the content of users' misconceptions.

A primary aim of this study was to examine the hypothesis that differences in interfaces would be reflected in differences in the knowledge of users of those interfaces. The likelihood of such differences is not treated very explicitly in most of the theories reviewed here, but this hypothesis is not incompatible with the majority of them, though Moray's lattice theory seems to ignore the role of the interface in allowing users to infer system states. In principle, analysis in the spirit of Payne et al's (1990) YSS can be used to derive hypotheses regarding the knowledge, operational or figurative, that users are likely develop while using a particular interface. This would require reasoning from the specific actions required to accomplish particular goals with a given system. However, Payne et al provided no principles or guidelines for doing so and acknowledged the great complexity of YSS analysis, which was presented for discussion rather than as a basis of prediction.

GOMS descriptions of skilled methods will necessarily vary with the interface, but GOMS is silent with respect to development of knowledge. Polson and Lewis' (1990) CE+ also seems compatible with the idea that the interface affects user knowledge, and Polson and Lewis provided a list of eight interface design principles to promote successful guessing. In principle the heuristics of the problem solving and learning components of CE+ can be applied to derive specific predictions regarding the knowledge developed by a (lowest-common-denominator) user of a particular system and interface. Clearly any single user is likely to attain only a subset of the possible knowledge of the system allowed by these heuristics, determined by the user's goals, etc. But there are other serious difficulties in using CE+ to describe or predict user knowledge. CE+ is unable to readily describe conceptual knowledge or its origin, hence is incompatible with any effects of interface upon conceptual knowledge and the organization of conceptual knowledge, and does not treat the relation of conceptual knowledge to operational knowledge and performance.

Treatment of this relation between conceptual and operational knowledge is another area in which the theories previously reviewed here differ. Some theories, such as GOMS, make no pretense of treating the details of cognition, including knowledge representation. But there are other theories to which knowledge representation is central. The evidence suggesting a dissociation between operational and conceptual knowledge is not consistent with any theory, such as that of Tennyson and Cocchiarella (1986), that requires acquisition of concepts and operations in a particular order. Because declarative knowledge, described by Anderson (1982) as preceding the development of procedural knowledge, can be what has here been called simply descriptive knowledge, perhaps in the form of received rules initially supporting operations, Anderson's theory is not inconsistent with the observed dissociation. But the failure to treat conceptual knowledge explicitly is a weakness of theoretical frameworks based upon production systems, including Anderson's.

As noted, the work of Kieras and Polson and associates, including the CE+ framework of Polson and Lewis (1990), has also largely ignored conceptual knowledge, and the present study among many others suggests that this is a very significant weakness. Some mental-model oriented theories, such as the yoked state space (YSS) of Payne, Squibb, and Howes (1990) and Moray's (e.g., 1987) lattice theory do treat conceptual knowledge, though somewhat implicitly in the case of lattice theory, but the precise nature of the relationship between conceptual and operational knowledge is tangential to the concerns of many theories of mental models. While YSS tends to imply that operational accounts are characteristic of novices and precede figurative accounts, which are more characteristic of experts, this is not a requirement of the theory, which is therefore compatible with the conceptual-operational dissociation. Moray's theory deals with knowledge of transition functions in a state space, and apparently cannot explain the development of conceptual knowledge or the conceptual-operational dissociation.

Context effects and other memory phenomena are also largely orthogonal to the concerns of most of the theories generated for or applied to HCI. As previously noted, GOMS cannot account for difficulty in recall. Nor does the GOMS extension CE+ explicitly consider the details of memory phenomena. As Polson and Lewis (1990) admitted, CE+ has no difficulty retrieving the productions it stores. Certainly CE+ could not readily describe, for example, M4's better access to knowledge related to Directory/Folder in the context of the Macintosh as opposed to DOS.

Theories of mental models also, in general, fail to treat context or other effects upon memory accessibility. All of these theories could be modified to admit such effects. But the relation, if any, of memory phenomena to the constructs of these theories and to performance remains to be explored. Does conceptual understanding, for example a "figurative" explanation in Payne et al's (1990) YSS terms, promote better recall, as would be suggested by levels of processing (Craik and Lockhart, 1972)? Another possibility is that inability to accomplish an operation might hinder recall, for example by reducing frequency of retrieval. This possibility might help explain M4's inability to think of creating a subdirectory in the context of DOS, though he suggested doing so during the Macintosh questions. At any rate, treating mental models, production rule systems, or other theoretical constructs as unvarying unless modified by the theoretically-allowed learning processes is a simplification of doubtful merit. And just this is implied by neglect of memory effects.

Another area neglected in many theories is the acquisition of knowledge through problem solving. Subjects in this study seemed to evince several heuristics for dealing with unfamiliar situations. Of the theories reviewed here, CE+ and theories of analogical mapping attempt explicit analyses of users' behavior in problem solving situations. Mental model theories such as YSS and lattice theory do not address problem solving directly. Moray's (1987) suggestion that users refer to successive layers of his lattice construct when confronted with failure of the default predictions in a current layer describes, in very general terms, a form of problem solving activity unrelated to initial knowledge acquisition.

One well-known problem solving method is analogical reasoning. The present study did not provide clear cut examples of analogical reasoning, though M1's realization during the Macintosh questions that creating a subdirectory would be an appropriate organizing strategy in DOS may be an example of analogy. On other occasions subjects seemed ready to utilize new predicates in known methods. In earlier discussion this was termed an extension heuristic, but it could also be described as a simple form of analogical reasoning. An example was B1's transporting a graphic from Excel to Word on the Macintosh, apparently for the first time.

The problem solving component of Polson and Lewis' (1990) CE+, on the other hand, is limited in ascribing to the user a single heuristic, label following, and cannot address the role of prior knowledge in guessing. Examples of guessing as problem solving from the present study might be M1's use of Microsoft Word "just for fun" to open an unfamiliar file, and M4's suggestion of closing everything and starting over in response to the "Application Busy or Missing" error message. Quite evidently neither attempt is related to label following, the sole problem solving mechanism in CE+. Nor does CE+ appear to address the range of subjects' causal inferences. A relevant example might be subjects' common supposition that having both applications open is a necessity for using the clipboard to transfer data between them. Accounting for this idea in terms of CE+ heuristics is somewhat awkward. But perhaps people are prone to assigning a prerequisite link, in CE+ terms, in cases of conditions that are sufficient but not necessary, and the users expressing this idea had all originally attempted this operation with both applications running. While the example is not entirely clear, it does suggest a role for conceptual knowledge in guessing about causality.

The detailed, highly specific, nature of the effects, such as the misconception described above, that can arise in human-computer interaction are an imposing difficulty for the study of HCI and for interface design. This is illustrated by the varying misconceptions evinced by subjects in the present study. At present even describing users' errors, much less the cognitive or other difficulties giving rise to them, is a problem for HCI.. Lang, Graesser, and Hemphill (1991) concluded that HCI does not have a theory-based categorization method adequate to describe the set of errors that they studied, and even an ad hoc scheme induced from those errors was only moderately reliable across judges. Lang, et al also concluded that a mental model explanation could predict the pattern of perseverance errors in their study, but that the production architecture SOAR (Laird, et al., 1987), which has a "No Operator Retry Rule", could not. In the present study, several subjects committed perseverance errors, hence any theoretical architecture such as SOAR that precludes such errors cannot explain their behavior. But it is certainly possible to postulate error-handling productions that persevere temporarily to provide against, for example, mistyping of commands.

Theory suggests other possible origins for user errors. Misleading applications of analogy have been described by e.g., Halasz and Moran (1982), but it would be difficult to ascribe the errors observed in this study to such a source. Many of the theories developed for or applied to HCI, GOMS for example, are concerned with correct knowledge, or in the case of CCT and CE+ also the acquisition of such knowledge, rather than errors and/or misconceptions. Yet understanding of the mechanisms responsible for errors is surely a necessity for HCI. Admittedly some misconceptions may illustrate Moray's (e.g., 1987) assertion that user knowledge is a homomorphism of the system, but lattice theory allows no predictions regarding the limitations of user knowledge in specific cases, and seems to ignore the interface. Consider M2's idea that running the appropriate Macintosh application first is a more reliable method of opening a document than double-clicking the document. M2 had an adequate but not always efficient procedure coupled with confused or incomplete declarative knowledge. The existence of this confusion is closely related to the nature of the

Macintosh GUI, since it could not arise without the availability of two methods of running an application and opening a document.

In summary, the relation of the findings of the present study to existing theory is at best obscure. No single existing theory is applicable to all the phenomena seen in this study. None of the theories reviewed addresses context effects or user misconceptions. The general theories and frameworks, such as Moray's lattice theory, offer little aid in understanding or predicting specific phenomena. Other theories, such as GOMS or CE+, are restricted in scope and cannot readily describe possible origins for many of the results of this study.

Conclusions

The present state of theory and research regarding users' knowledge of computer systems and the effects upon that knowledge of user interfaces seems insufficient for the purpose of practical prediction. There is little relevant empirical research. And the competing (or complementary?) theoretical frameworks such as CE+ (Polson and Lewis, 1990), Structure- Mapping Theory (e.g., Gentner, 1983) and the wide variety of theories of mental models (see e.g., Gentner and Stevens, 1983; Rouse and Morris, 1986) are rather general in nature and provide little account of specific effects of past experience or interface elements.

The present research is directed at tentatively characterizing users' knowledge, and the representations of that knowledge, of systems and applications and, ultimately, the effects of specific types of interface experience upon that knowledge and it's representation. An understanding of how interfaces afford varieties of knowledge is potentially very useful for design purposes.

Examination of data from Macintosh users and PC users as groups and as individuals, along the lines of the case study approach that many researchers find advisable in the study of cognitive neuropsychology, has provided suggestive evidence of several potentially important phenomena. Subjects evinced understanding of concepts that they were unable to apply, or conversely, of operations for which they apparently lacked relevant conceptual knowledge. Subjects' knowledge did not appear to be equally accessible in all circumstances, that is computer use is evidently subject to context effects. The evidence regarding operations related to Storage Buffer/Clipboard, and to some extent that of individual users' knowledge of that concept, was consistent with the idea that user interfaces differ in their support for users' acquisition of conceptual and operational knowledge. Differences between the group average Pathfinder networks for Macintosh and PC users suggest that the system knowledge of these groups differs in content and organization. The data from this study seem to illustrate the operation of miscellaneous heuristics for dealing with unfamiliar, or somewhat unfamiliar, situations. These include the label-following heuristic (Lewis, 1987; Lewis et al, 1986; Polson and Lewis,1990), but illustrate the importance of prior knowledge in determining label-following. And, finally, this study has naturally revealed misconceptions or superstitions on the part of some subjects. Future research may be directed at a more detailed examination of users' learning and problem solving processes, and of the concomitant transformations in knowledge organization.

Experimentally controlled research into human acquisition and use of any real degree of expert skill with complex systems such as personal computers is extraordinarily difficult, given the time scales involved and the large number of potentially interacting system and subject variables, ranging from, but not limited to, interface elements and system functionality to individual aptitudes and experience that can critically affect problem solving and learning. Longitudinal studies are desirable, but studies such as the present one, employing existing user populations, allow more rapid completion and have the merit of contact with the knowledge and methods of real-world users. Further such efforts are necessary. At the same time, it is necessary to continue to bring HCI phenomena into the laboratory for controlled study, despite the difficulty of imputing practical importance to experimental results.

The general method of the present study could be applied with users of other systems and interfaces, or prototypes of interface elements of interest, and with focal questions other than the effect of interface upon knowledge. Several of the observations from the present study merit closer examination, either by more pointed application of the present method or in controlled laboratory study. A more heterogeneously experienced selection of subjects might reveal more substantial and more detailed effects of interface upon knowledge. Users of different systems could be presented with a wider range of performance tasks, including novel but characteristic tasks, and their errors, misconceptions, and problem solving methods examined. There is certainly room for more careful probing, at the time of commission, for the causes of errors in question answering and performance tasks than was undertaken in this study.

Some questions are suitable for examination in laboratory studies using contrived systems and interfaces simple enough to be rapidly learned. In such a setting it should, for example, be possible to investigate context effects, and the ability of subjects to discriminate different contexts, by training the same subjects to varying levels of familiarity with different systems or situations and exploring effects upon recourse to known concepts or operations.

Of course, these projected studies offer at most incremental improvement in the understanding of HCI, rather than a Grand Unified Theory. The present study has, perhaps, highlighted some of the limitations of existing theories. It is strongly suggested that learning of concepts and procedures is related to prior knowledge, and that the heuristics employed in learning may often utilize and/or operate upon conceptual knowledge. Understanding the determinants of usability, and of the way in which interfaces differ, must require understanding of all of the forms of knowledge afforded by the interface that can affect performance. Further, it is necessary to understand how users organize and make use of (and under what circumstances they are able to make use of) their knowledge.

The evidence of individual users in this study suggests a lively interplay between conceptual and operational knowledge and problem solving strategies, in novel situations and also in routine operation of a system. Any theories of the interface that fail to thoroughly consider these factors and more must be at best incomplete.

References

Allwood, C. M. (1990). Learning and using text-editors and other application programs. In P. Falzon (Ed.) Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction (pp. 85-106). London: Academic Press.

Anderson, J. R. (1982). Acquisition of Cognitive skills. Psychological Review, 89, 369-406.

Anderson, (1983). The Architecture of Cognition, Cambridge, MA: Harvard University Press.

Anderson and Thompson (1986). Use of analogy in a production system architecture. Paper presented at the Illinois Workshop in Similarity and Analogy, Champaign-Urbana.

Bartlett, F. C. (1932). Remembering: An experimental and social study. Cambridge, England: Cambridge University Press.

Borgman, C. L. (1986). The user's mental model of an information retrieval system: An experiment on a prototype online catalog. International Journal of Man-Machine Studies, 24(1), 47-64.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. Human-Computer Interaction, 5(1), 1-48.

Broadbent, D. E. (1977). levels, hierarchies, and the locus of control. Quarterly Journal of Experimental Psychology, 29, 181-201.

Broadbent, D. E., Fitzgerald, P., & Broadbent, M. H. P. (1986). Implicit and explicit knowledge in the control of complex systems. British Journal of Psychology, 77, 33-50.

Card, Moran, & Newell, (1980). Computer text-editing: An information processing analysis of a routine skill. Cognitive Psychology, 12(1), 32-74.

Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Carroll, J. M., & Campbell, (1986). Softening up hard science: Reply to Newell and Card. Human-Computer Interaction, 2(3), 227-249.

Carroll, J. M., & Mack, R. L. (1985). Metaphor, computing systems, and active learning. International journal of Man-Machine Studies, 22, 39-57.

Carroll, J. M., Mack, R. L., & Kellogg, W. A. (1988). Interface metaphors and user interface design. In M. Helander (Ed.) Handbook of Human-Computer Interaction, (pp. 67-85). North-Holland: Elsevier Science Publishers.

Carroll, J. M., & Mazur, (1986). LisaLearning. IEEE Computer, 19(11), 35-49.

Carroll, J. M., & Olson, J. R. (Eds.) (1987). Mental Models in Human-Computer Interaction.: Research Issues About What the User of Software Knows. Washington, D.C.: National Academy Press.

Carroll, J. M., & Olson, J. R. (1988). Mental models in human-computer interaction. In M. Helander (Ed.) Handbook of Human-Computer Interaction, (pp. 45-65.) North-Holland: Elsevier Science Publishers B. V.

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. In J. Carroll (Ed.) Interfacing Thought. (pp. 184-235). Cambridge, Mass: The MIT Press.

Carroll, J. M., & Thomas (1982). Metaphor and cognitive representation of computing systems. IEEE transactions on Systems, Man, and Cybernetics, 12(2), 107-116.

Chi, M. T. H. Feltovich, & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5(2), 122-152.

Clement, J. (1982). Spontaneous analogies in problem solving: The progressive construction of mental models. Paper presented at AERA, New York.

Cooke, N. M., & McDonald, J. E. (1986). A formal methodology for acquiring and representing expert knowledge. Proceedings of IEEE, 74, 1422-1430.

Cooke, N. J. & Schvaneveldt, R. W. (1988). Effects of programming experience on network representations of abstract programming concepts. International Journal of Man-Machine Studies, 29(4), 407-427.

Davis, S. & Bostrom, R. P. (1989). Training end-users to compute: An experimental investigation of the roles of the computer interface and training methods. Unpublished manuscript.

De Kleer, J. & Brown, J. S. (1983). Assumptions and ambiguities in mechanistic mental models. In D. Gentner and A. L. Stevens (Eds.), Mental Models (pp. 155-190). Hillsdale, NJ: Erlbaum.

Douglas, S. A., & Moran, T. P. (1983). Learning text-editor semantics by analogy. CHI'83 Human Factors in Computing Systems Proceedings, New York: ACM SIGCHI.

Dumais, S. T. & Jones, W. P. (1985). A comparison of symbolic and spatial filing. Proceedings of CHI 1985, San Francisco.

Dyck, J. E. (1990). Transfer of learning from DOS to Finder for novice users. Report on Research Opportunity Award (NSF Grant IRI-8722792). Fresno: California State University.

Ellis, A. W., & Young, A. W. (1988). Human Cognitive Neuropsychology. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ericsson, K. A. & Simon, H. A. (1980). Verbal Reports as Data. Psychological Review, 87(3), 215-251.

Ericsson, K. A. & Simon, H. A. (1984). Protocol Analysis. Cambridge MA: MIT Press.

Falkenheimer, Forbus, & Gentner, (1986). The structure-mapping engine. in Proceedings of the International Association for Artificial Intelligence., Philadelphia, PA.

Forbus, K. D., & Gentner, D. (1986). Learning physical domains: Toward a theoretical framework. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.) Machine Learning: An Artificial Intelligence Approach, Vol. II (pp. 311-348). Los Altos, CA: Morgan Kaufmann.

Foss, D. F., & DeRidder, M. (1987). Technology Transfer: On learning a new computer-based system. In J. Carroll (Ed.) Interfacing Thought. (pp. 184-235). Cambridge, Mass: The MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7(2), 155-170.

Gentner, D., & Stevens, A. L. (Eds.). (1983). Mental Models, Hillsdale, NJ: Erlbaum.

Gentner, D. & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. Cognitive Science, 10(3), 277-300.

Gillan, D. J., Breedin, S. D., & Cooke, N. J., (1992). Network and multidimensional representations of the declarative knowledge of human-computer interaction design experts. International Journal of Man-Machine Studies, 36, 587-616.

Greeno, J. G. (1983). Conceptual entities. In D. Gentner and A. L. Stevens (Eds.), Mental Models (pp. 227-252). Hillsdale, NJ: Erlbaum.

Halasz, F. G., & Moran, T. P. (1982). Analogy considered as harmful. In Proceedings of a Conference on Human Factors In Computing Systems. New York: Association for Computing Machinery.

Halasz, F. G. & Moran, T. P. (1983). Mental models and problem solving in using a calculator. In Proceedings of CHI '83 Human Factors In Computing Systems. New York: Association for Computing Machinery.

Holyoak, K. J. (1985). The pragmatics of analogical transfer. In. G. H. Bower (Ed.), The Psychology of learning and Motivation (Vol 19). New York: Academic.

Holyoak, K. J. & Koh, K. (1987). Surface and structural similarity in analogical transfer. Memory & Cognition, 15, 332-340.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. Cognitive Science, 13, 295-355.

Howes, A., & Payne, S. J. (1990). Display-based competence: towards user models for menu-driven interfaces. International Journal of Man-Machine Studies, 33, 637-655.

Johnson, S. C. (1967). Hierarchical Clustering Schemes. Psychometrika, 32, 241-254.

Karat, J. (1988). Software evaluation methodologies. In M. Helander (Ed.), Handbook of Human-Computer Interaction (pp. 891-903). North-Holland: Elsevier.

Karat, J., McDonald, J. E., & Anderson, M. (1986). A comparison of menu-selection techniques: Touch panel, mouse, and keyboard. International Journal of Man-Machine Studies, 25(1), 73-88.

Kay, D. S., & Black, J. B. (1985). Changes in memory representation with increasing expertise in using computers. Bulletin of the Psychonomic Society, 33(4), 289.

Kay, D. S., & Black, J. B. (1990). Knowledge transformations during the acquisition of expertise. In S. P. Robertson, W. W. Zachary, & J. B. Black (Eds.) Cognition, Computing, and Cooperation, (pp. 268-303.) Norwood, NJ: Ablex Publishing Corporation.

Kellogg, W. A. & Breen, T. J. (1990). Mental models and user performance. In R. W. Schvaneveldt (Ed.), Pathfinder Associative Networks, (pp. 179-195). Norwood, NJ: Ablex Publishing Corporation.

Kieras, J. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), The Handbook of Human-Computer Interaction, (pp. 135-158). Amsterdam: North-Holland.

Kieras, D. E. & Bovair (1984). The role of a mental model in learning to operate a device. Cognitive Science, 8, 255-273.

Kieras, D. E., & Bovair, S. (1986). The acquisition of procedures from text: A production system analysis of transfer of training. Journal of Memory and Language, 25, 507-524.

Kieras, D. E. & Polson, P. G. (1983). A generalized transition network representation of interactive systems. Proceedings of the CHI 1983 Conference on Human Factors in Computing, Boston, MA: Association for Computing Machinery.

Kieras, D. E. & Polson, P. G. (1985). An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies, 22, 365-394.

Khalifa, M. (1991). TAMPEL: A task analytic methodology for predicting ease of learning of interactive computer systems. Poster presented at CHI '91 Conference on Human Factors in Computing, New Orleans, LA.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrica, 29, 1-27.

Laird, J., Newell, A., & Rosenbloom, P. (1987). SOAR: An architecture for general intelligence: Artificial Intelligence, 33, 1-64.

Lang, K., Graesser, A. C. & Hemphill, D. D. (1991). Understanding errors in human computer interaction. Poster presented at CHI '91 Conference on Human Factors in Computing, New Orleans, LA.

Lansdale, M. W., Simpson, M., & Stroud, T. R. M. (1987). Comparing words and icons as cue enrichers in an information retrieval task. In H. J. Bullinger & B. Shackel (Eds.) Human-Computer Interaction-INTERACT '87, (pp. 911-915). North-Holland: Elsevier Science Publishers B. V.

Lewis C. H. (1988). How and why to learn why: Analysis-based generalization of procedures. Cognitive Science, 12, 211-256.

Lewis, C., Casner, S., Schoenberg, V., & Blake, M. (1987). Analysis-based learning in human-computer interaction. In H. J. Bullinger & B. Shackel (Eds.) Human-Computer Interaction-INTERACT '87, (pp.275-280.) North-Holland: Elsevier Science Publishing B. V.

Mayes, J. T., Draper, S. W., Mcgregor, M. A., & Oatley , K. (1988). Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens. In D M. Jones & R. Winder (Eds.) People and Computers IV. Cambridge, UK: Cambridge University Press.

Michard, A. (1982). Task allocation between man and computer for an electronic appointment book., Travail-Humain, 45(2), 241-249.

Moran, T. P. (1983). Getting into a system: External-internal task mapping analysis. Proceedings of the CHI 1983 Conference on Human Factors in Computing, ACM SIGCHI.

Moray, N. (1987). Intelligent aids, mental models, and the theory of machines. International Journal of Man-Machine Studies, 27, 619-629.

Morris, C. D., Bransford, J. D., & Franks, J. J. (1977). Levels of processing versus transfer appropriate processing. Journal of Verbal Learning and Verbal Behavior, 16, 519-534.

Muncher, E. (1989). The acquisition of spreadsheet skills. Unpublished master's thesis. University of Colorado, Department of Psychology. cited in Polson and Lewis (1990).

Neisser, U. (1967). Cognitive Psychology. New York: Appleton.

Newell, A. (1980). Reasoning, problem solving, and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.). Attention and Performance VIII (pp. 693-718). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Newell, A. & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

Nisbett, R. E., Fong, G. T., Lehman, D. R., & Cheng, P. W. (1987). Teaching reasoning. Science, 238(4827), 625-631.

Nisbett, R. E. & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. Psychological Review, 84(3), 231-259.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner and A. L. Stevens (Eds.), Mental Models, (pp. 7-14). Hillsdale, NJ: Erlbaum.

Moray, N. (1987). Intelligent aids, mental models, and the theory of machines. International Journal of Man-Machine Studies, 27, 619-629.

Morris, C. D., Bransford, J. D., & Franks, J. J. (1977). Levels of processing versus transfer appropriate processing. Journal of Verbal Learning and Verbal Behavior, 16, 519-534.

Muncher, E. (1989). The acquisition of spreadsheet skills. Unpublished master's thesis. University of Colorado, Department of Psychology. cited in Polson and Lewis (1990).

Neisser, U. (1967). Cognitive Psychology. New York: Appleton.

Newell, A. (1980). Reasoning, problem solving, and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.). Attention and Performance VIII (pp. 693-718). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Newell, A. & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

Nisbett, R. E., Fong, G. T., Lehman, D. R., & Cheng, P. W. (1987). Teaching reasoning. Science, 238(4827), 625-631.

Nisbett, R. E. & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. Psychological Review, 84(3), 231-259.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner and A. L. Stevens (Eds.), Mental Models, (pp. 7-14). Hillsdale, NJ: Erlbaum.

Norman D. A. (1987). Cognitive engineering-cognitive science. In J. Carroll (Ed.) Interfacing Thought. (pp. 184-235). Cambridge, Mass: The MIT Press.

Olson, J. R & Olson G. M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. Human-Computer Interaction, 5, 221-265.

Payne, S. J. (1987). Complex problem spaces: Modeling the knowledge needed to use interactive devices. In H. J. Bullinger and B. Shackel (Eds.) Human-Computer Interaction-INTERACT '87, (pp. 203-208). North-Holland: Elsevier Science Publishers.

Payne, S. J. & Green, T. R. (1986). Task-action grammars: A model of the mental representation of task languages. Human-Computer Interaction, 2(2), 93-133.

Payne, S. J. (1987). Complex Problem spaces: Modeling the knowledge users need to use interactive devices. In H. J. Bullinger & B. Shackel (Eds.) Human-Computer Interaction-INTERACT '87, (pp. 203-208). North-Holland: Elsevier Science Publishers B. V.

Payne, S. J. & Green, T. R. (1989). The structure of command languages: An experiment on task-action grammar. International Journal of Man-Machine Studies, 30(2), 213-234.

Payne, S. J., Squibb, H. R., & Howes, A. (1990). The nature of device models: The yoked state space hypothesis and some experiments with text editors. Human-Computer Interaction, 5(4), 415-444.

Peterson, S. E., Ridenour, M. E., and Somers, S., L. (1990). Declarative, conceptual, and procedural knowledge in the understanding of fractions and acquisition of ruler measurement skills. Journal of Educational Psychology, 58(3), 185-193.

Polson, P. G. (1987). A quantitative theory of human-computer interaction. In J. Carroll (Ed.) Interfacing Thought. (pp. 184-235). Cambridge, Mass: The MIT Press.

Polson, P. G. (1988). The consequences of consistent and inconsistent interfaces. In R. Guindon (Ed.) Cognitive science and its application for human-computer interaction (pp. 59-162). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Polson, P. G., & Kieras D. E. (1985). A quantitative model of the learning and performance of text editing knowledge. Proceedings of the CHI 1985 Conference on Human Factors in Computing, (pp. 78-83). San Francisco, Association for Computing Machinery.

Polson, P. G., & Lewis, C. H. (1990). Theory-based design for easily learned interfaces. Human-Computer Interaction, 5, 191-220.

Polson, P. G., Munchner, E., & Englebeck, G. (1986). A test of a common elements theory of transfer. In M. Mantei & P. Oberton (Eds.), Proceedings CHI'86 Human Factors in Computing Systems (pp. 78-83). New York: Association for Computing Machinery.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(2), 257-266.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man, and Cybernetics, SMC-15(2), 235-243.

Rasmussen, J., & Vicente, K. (1989). Coping with human errors through interface design: Implications for ecological interface design. International Journal of Man-machine Studies, 31(5), 517-534.

Rosson, M. B., & Grischkowski, N. L. (1987). Transfer of learning in the real world. In H. J. Bullinger & B. Shackel (Eds.) Human-Computer Interaction-Interact '87. (pp. 269-273). North-Holland: Elsevier Science publishers B. V.

Rouse, W. B. (1977). A theory of human decision-making in stochastic estimation tasks. IEEE Transactions on Systems, Man, and Cybernetics, SMC-12, 605-610.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. Psychological Bulletin, 100(3), 349-363.

Rumelhart, D. E. & Norman, D. A. (1981). Analogical processes in learning. In Anderson, J. R. (Ed.) Cognitive Skills and Their Acquisition (pp. 335-359). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sanderson, P. (1989). Verbalizable knowledge and skilled task performance: Association, dissociation, and mental models. Journal of Experimental Psychology: Learning, Memory, and Cognition, 15(4), 729-747.

Schank, R. C., & Abelson, R. (1977). Scripts, Plans, Goals, and Understanding. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schmalhofer, F. (1987). Mental model and procedural elements approaches as guidelines for designing word processing instructions. In H. J. Bullinger & B. Shackel (Eds.) Human-Computer Interaction-Interact '87. (pp. 269-273). North-Holland: Elsevier Science Publishers B. V.

Schumacher, R. M & Gentner, D. (1988). Transfer of training as analogical mapping. IEEE Transactions on Systems, Man, & Cybernetics, 18(4), 592-599.

Schvaneveldt, R. W. (1990). Proximities, networks, and schemata. In R. W. Schvaneveldt (Ed.). Pathfinder Associative Networks, (pp. 135-148). Norwood, NJ: Ablex Publishing Corporation.

Schvaneveldt, R. W., & Durso, F. T. (1981). Generalized semantic networks. Bulletin of the Psychonomic Society, 18(2), 76-77.

Schvaneveldt, R. W. Dearholdt, D. W., & Durso F. T. (1985). Measuring the structure of expertise. International Journal of Man-machine Studies, 23(6), 699-728.

Sebrechts, M. M., Marsh, R. L., & Furstenburg, C. T. (1986,1987). Integrative Modeling: Changes in mental models during learning. In S. P. Robertson, W. W. Zachary, & J. B. Black (Eds.) Cognition, Computing, and Cooperation, (pp. 338-398). Norwood, NJ: Ablex Publishing Corporation.

Shallice, T. (1979). Case-study approach in neuropsychological research. Journal of Clinical Neuropsychology, 1, 183-211.

Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. Part. I. Psychometrika, 27, 125-140.

Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. Behaviour and Information Technology, 1(3), 237-256.

Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, Mass.: Addison-Wesley.

Singley, M. K., & Anderson J. R. (1985). The transfer of text editing skill. International Journal of Man-machine Studies, 22, 403-423.

Singley, M. K., & Anderson J. R. (1988). Learning and transfer in text editing. Human-Computer Interaction, 3, 275-307.

Smith, K. H., Zirkler, D., & Mynatt, B. (1985). Transfer of training from introductory computer courses is highly specific . . . and negative! Behavior Research methods, Instruments, & Computers, 17(2), 259-264.

Tennyson, R.D., & Cocchiarella, M. J. (1986). An empirically based instructional design theory for teaching concepts. Review of Educational Research, 56, 40-71.

Tetzlaff, L. (1987). Transfer of learning: beyond common elements. In CHI + GI 1987. ACM.

Thorndike, E. L. & Woodworth, R. R.(1901). The influence of improvement in one mental function upon the efficiency of other functions. Psychological Review, 8, 247-261.

Torgerson, W. S. (1952). Multidimensional Scaling: I. Theory and method. Psychometrika, 17, 401-419.

Tulving, E., & Pearlstone, Z. (1966). Availability versus accessibility of information in memory for words. Journal of Verbal Learning and Verbal Behavior, 5, 381-391.

Tulving, E. & Thompson, D. M. (1971). Retrieval Processes in recognition memory - effects of associative context. Journal of Experimental Psychology, 87, 116-124.

Van Bussel, F. J. (1980). Human prediction of time series. IEEE Transactions on Systems, Man, and Cybernetics, 10(7), 410-414.

Van Heusden, A. R. (1980). Human prediction of third-order autoregressive time series. IEEE Transactions on Systems, Man, and Cybernetics, 10(1), 38-43.

Whorf, B. L. (1956). Language, Thought, and Reality. Cambridge, MA: MIT Press.

Whiteside, J., Jones, S., Levey, P.S., & Wixon, D. (1985). User performance with command, menu, and iconic interfaces. Proceedings of the CHI 1985 Conference on Human Factors in Computing, San Francisco, CA: ACM SIGCHI.

Williams, M. D., Hollan, J. D. & Stevens, A. L. (1983). Human reasoning about a simple physical system. In D. Gentner and A. L. Stevens (Eds.) Mental Models, (pp.131-153). Hillsdale, NJ: Erlbaum.

Appendix A: Materials

Subject Identifier ________

Computer Experience Questionnaire

1. Years of computer experience: ________
2. What was the first kind of computer that you used? ______________
3. What other kinds of computers (i.e. Macintosh, IBM PC or compatible, UNIX workstation) and operating systems have you used? Please circle a number to indicate your relative familiarity with the system, with one indicating no familiarity and five indicating great familiarity.

IBM PC	1	2	3	4	5
	Not familiar				Very familiar
Macintosh	1	2	3	4	5
	Not familiar				Very familiar
UNIX Workstation	1	2	3	4	5
	Not familiar				Very familiar

Other Computers (Please name)

______________ 1 2 3 4 5

Not familiar Very familiar

______________ 1 2 3 4 5

Not familiar Very familiar

______________ 1 2 3 4 5

Not familiar Very familiar

4. Do you use a computer at home? yes no

   If yes what kind?____________________________

5. What kinds of computer applications do you use? Circle types used and indicate type or types of computer (e.g. Macintosh, IBM PC or compatible, UNIX workstation) on which you use each kind of application.

Application Type(s) of computer

communication____________________________

database____________________________

desktop publishing____________________________

graphics____________________________

personal information manager (e.g., calendars)______________

presentation (e.g., presentation aids)______________

programming (e.g., Basic)____________________________

project management____________________________

spreadsheet____________________________

statistics and math____________________________ word processing________________________________

other (please list)________________________________

____

6. What computer related courses have you taken? Please give a brief description (e.g. a course title) of the class, and indicate the computer(s) used in the spaces provided.

<u>Course</u> <u>Type(s) of computer</u>

____

____

____

____

____

7. In general, how easy is it for you to learn to use new computer programs?

(Circle one)

1	2	3	4	5
very difficult	moderately difficult	somewhat easy/ somewhat difficult	moderately easy	very easy

Please answer the following Sections of questions according to your computer experience as indicated below. Additional paper is available for your answers if needed:

<u>Computer experience</u>	<u>Sections to answer</u>
DOS only	Section I
Macintosh only	Section II

DOS and Macintosh learned DOS first	Section I, II, and III
Macintosh and DOS learned Macintosh first	Section I, II, and IV

Section I: Users of DOS only or both DOS and Macintosh answer this section.

1. Are there any capabilities that you find lacking in DOS? Please explain.

____

____

____

2. Are there any areas that you initially found difficult in using DOS? Please explain. Please indicate if you still find this area difficult and, if not, try to explain why.

____

____

____

3. Are there any areas, other than those discussed above, that you still find difficult in using DOS? Please explain.

____

____

____

4. Are there any areas that you particularly like about using DOS? Please explain.

____

____

____

<u>Section II</u>: Users of Macintosh only or both DOS and Macintosh answer this section.

1. Are there any capabilities that you find lacking in the Macintosh? Please explain.

____

____

____

2. Are there any areas that you initially found difficult in using the Macintosh? Please explain. Please indicate if you still find this area difficult and, if not, try to explain why.

____

____

____

3. Are there any areas, other than those discussed above, that you find difficult in using the Macintosh? Please explain.

____

____

____

4. Are there any areas that you particularly like about using the Macintosh? Please explain.

____

____

____

<u>Section III</u>: Users of both DOS and Macintosh who learned DOS first answer this section.

1. Are there any capabilities in DOS that you find lacking in the Macintosh? Please explain.

____

____

____

2. Are there any capabilities in the Macintosh that you find lacking in DOS? Please explain.

____

____

____

3. Are there any areas that you think you found difficult in the Macintosh because of what you remembered from DOS? Please explain.

____

____

____

4. Are there any areas that you think were easier in using the Macintosh because of what you remembered from DOS? Please explain.

____

____

____

5. Has using the Macintosh changed your thinking about DOS and its usefulness? Has learning about the Macintosh helped you in understanding DOS? If so, how?

____

____

____

Section IV: Users of both Macintosh and DOS who learned Macintosh first answer this section.

1. Are there any capabilities in the Macintosh that you find lacking in the DOS? Please explain.

____

____

____

2. Are there any capabilities in DOS that you find lacking in the Macintosh? Please explain.

____

____

____

3. Are there any areas in DOS that you think you found difficult because of what you remembered from the Macintosh? Please explain.

____

____

____

4. Are there any areas in DOS that you think were easier because of what you remembered from the Macintosh? Please explain.

____

____

____

5. Has using DOS changed your thinking about the Macintosh and its usefulness? Has learning DOS helped you in understanding the Macintosh? If so, how?

____

____

____

Subject Identifier_____

DOS Questions

These questions will be answered aloud.

1. Your screen shows only C:\> and a blinking _. You want to run a program. You think the program is on a floppy disk in the A drive. You are not entirely sure of the name of the program. What should you do? Why would this help?

2. You have typed something and gotten the message "Bad command or file name". Describe several possible reasons for this error message.

3. You think the program is on drive C but you are not sure where. What should you do?

4. You have tried to display the contents of drive C. Some of the names you see are a single word, but some are a word followed after a space by three letters. What is the difference?

5. Which names might indicate programs you can run? What do some of the other names indicate?

6. You don't see the program you were looking for. What more can you do to try to find it on this drive?

7. You have found your program. What must you do to run it?

8. You have several disks containing a brand-new program which you wish to begin using. You have misplaced the documentation. What can you do?

9. You want to be sure that nothing belonging with your new program gets mixed up with anything else on your disk. What can you do? How would you do it and why would it help?

10. In order to make room you decide to remove some files. Which files would it be particularly unwise to remove? Why?

11. You are using a word processor and wish to move some text to a different location within the document. How could you do so? . Right afterwards you also want to put the same text in another location. How can you do so? Why does this work?

12. You want to have a graphic within your document. What can you do? Why?

13. A co-worker is working with the same document and has changed the last three letters of the name of your document. What , if anything, must you do to continue working on it? Why?

14. Another co-worker using a different word processor has produced a document for you to work on. How might you be able to use it? What might your co-worker have done to make it easier for you to use his document ? Why is this necessary or useful?

15. When you have a computer related problem, how do you normally try to solve it? That is, do you for example ask someone, look in a manual, guess, experiment, etc.?

Subject Identifier_____

Macintosh Questions

These questions will be answered aloud.

1. Your screen shows only the menu bar and two disk icons. You want to run a program. You think the program is on a floppy disk. You are not entirely sure of the name of the program. What should you do? Why?

2. You have clicked something and gotten the message "Application busy or missing". Describe several possible reasons for this message.

3. You think the program is on the hard drive but you are not sure where. What should you do? Why?

4. You have tried to display the contents of the hard drive. You see little pictures like a sheet of paper with a bent edge and others that look like a file folder. What is the difference?

5. Which icons and names might indicate programs you can run?

6. You clicked on something and found yourself reading a document on the screen. What happened, and why?

7. You have found your program. What must you do to run it?

8. You have several disks containing a brand-new program which you wish to begin using. You have misplaced the documentation. What can you do?

9. You want to be sure that nothing belonging with your new program gets mixed up with anything else on your disk. What can you do? How?

10. In order to make room you decide to remove some files. Which files would it be particularly unwise to remove? Why?

11. You are using a word processor and wish to move some text to a different location within the document. How could you do so? . Right afterwards you also want to put the same text in another location. How can you do so? Why would this work?

12. You want to have a graphic within your document. What can you do? Why?

13. A co-worker is working with the same document and has somehow changed the icon of your document. What, if anything, must you do to continue working on it? Why?

14. Another co-worker using a different word processor has produced a document for you to work on. How might you be able to use it? What might your co-worker have done to make it easier for you to use his document ? Why is this necessary or useful?

15. When you have a computer related problem, how do you normally try to solve it? That is, do you for example ask someone, look in a manual, guess, experiment, etc.?

Subject Identifier____

Task Checklist

Please place a check mark beside any of the tasks that you do on a regular basis. Then for each checked task please circle the appropriate points on the provided scales to indicate how often you do the task and how difficult you find it to be in normal circumstances.

1. Back up your data. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

2. Create a new document ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

3. Open a previously created document. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

4. **Copy a document** ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

5. **Insert into a document, without retyping, something that originated in another document created using the same program.** ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

6. **Insert into a document something that originated in another program.** ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

7. **Save a document.** ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

8. **Save a document in a different format.** ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

9. Use a brand-new disk. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

10. Install a new program. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

11. Switch between programs. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

12. Move a file. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

13. Create a new directory/folder. ________

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

14. **Delete a directory/folder.** ______

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

15. **Delete a file.** ______

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

16. **Rearrange directories/folders.** ______

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

17. **Rename a directory/folder.** ______

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

18. **Rename a file.** ______

1	2	3	4	5
Very seldom	Seldom	Occasionally	Fairly often	Very often
1	2	3	4	5
Very difficult	Difficult	Somewhat difficult Somewhat easy	Fairly easy	Very easy

Subject Identifier____

Computer Concepts

Please rate your familiarity with each of the following concepts on a scale from 1 to 5 (circle one), with 1 being completely not familiar and 5 being very familiar. Then briefly define or describe each concept for which your rating was greater than 2.

File: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Creator of a File: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Filename Extension : 1 2 3 4 5

Not familiar Very familiar

____

____

____

Program/Application: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Data File: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Storage Buffer/Clipboard: 1 2 3 4 5

Not familiar Very familiar

____

____

____

File Format: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Operating System:	1	2	3	4	5
	Not familiar				Very familiar

____

____

____

Finder:	1	2	3	4	5
	Not familiar				Very familiar

____

____

____

Command Interpreter :	1	2	3	4	5
	Not familiar				Very familiar

____

____

____

Directory/Folder: 1 2 3 4 5

Not familiar Very familiar

____

____

____

ASCII: 1 2 3 4 5

Not familiar Very familiar

____

____

____

Memory (disk and RAM): 1 2 3 4 5

Not familiar Very familiar

____

____

____

Initializing/formatting a disk: 1 2 3 4 5

Not familiar Very familiar

____

____

____

DOS Task Instructions

1. Create a directory on drive C.

2. Find the file FILE.USE on the diskette provided and copy it to the new directory.

3. Open the file.

4. Now import the file into Lotus 123

5. Create a Graph using an existing .wk1 file.

6. Put the graph into a Word Processor document. You may use the file SAMPLE.TXT from the floppy.

7. Move the graph within the document.

Macintosh Task Instructions

1. Create a folder on the hard drive.

2. Find the file USE ME on the diskette provided and copy it to the new folder.

3. Open the file.

4. Now import the file into Excel.

5. Create a graph using an existing .Excel.

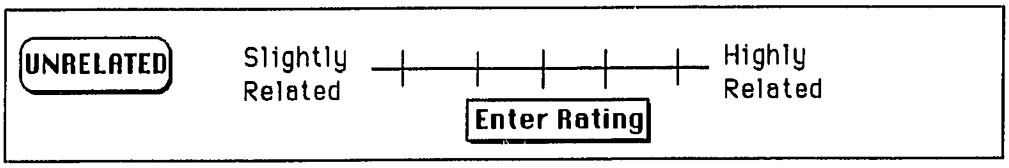
6. Put the graph into a Word Processor document. You may use the file SAMPLE TEXT from the floppy.

7. Move the graph within the document.

8. Open the program SuperPaint. Bring your graph into a SuperPaint document.

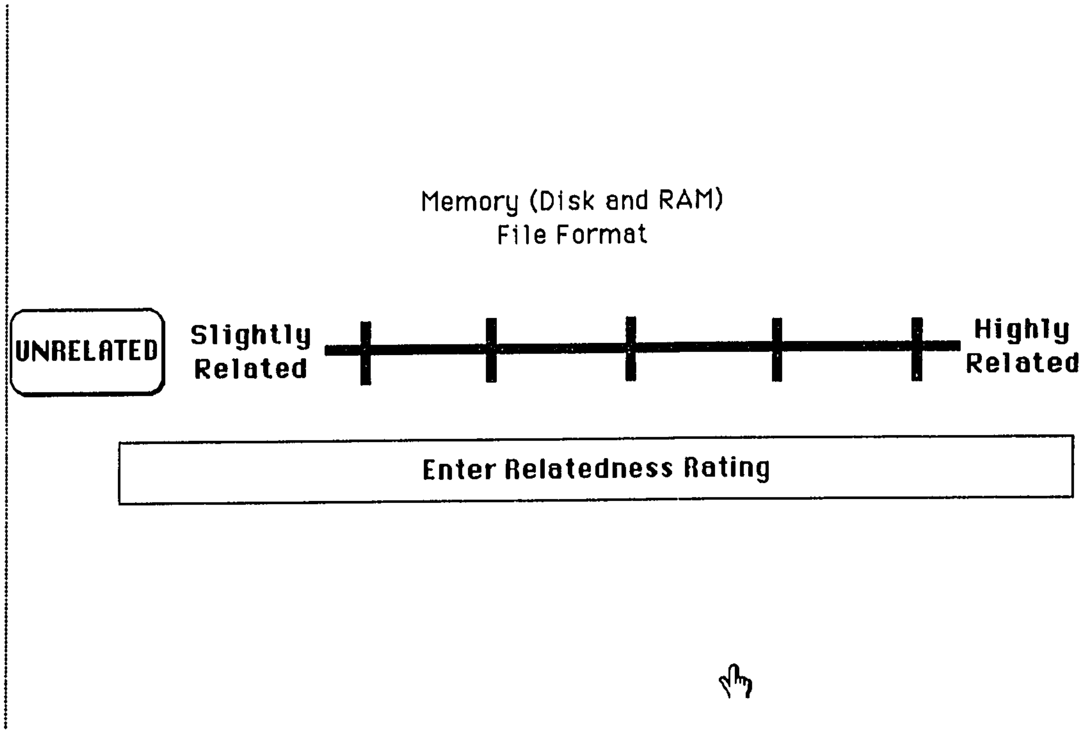
9. Predict the quality of a printed version of this document.

In this experiment you will be presented with 91 pairs of concepts and your task is to rate each pair according to the degree of relatedness of the concepts in that pair. Two items can be related in a number of different ways, but we are interested in your first impression of overall relatedness. To enter your ratings you will be using a scale like the one below.



You will use the mouse to point to a location on the scale. Note that you can also select "UNRELATED" if you feel that the items are not related at all. You may change your rating by simply clicking on a different location. The rating will be entered when you click on the button labeled "Enter Rating." If you have any questions at this time please ask the experimenter.
Before you begin making ratings you will be shown the entire set of concepts. Look them over to get an idea of the scope of the items you will be rating.

Continue


Memory (Disk and RAM)
File Format
UNRELATED
Slightly Related
Highly Related
Enter Relatedness Rating

Appendix B: Subjects

IBM PC Users

Subject PC1

Subject PC1 has about 6 years of computer experience. His first experience was with an IBM PC. He rated himself very familiar with the IBM PC and slightly familiar with the Macintosh and a Tandem computer. A former room mate of PC1's had a Macintosh circa 1986, and he made use of it to play games. PC1 feels that Macintoshes suffer from hardware limitations. PC1 uses an IBM clone, a 12 MHz 80286 based machine, at home. He uses 7 different application types and has take one computer related class. PC1 does not have Microsoft Windows, and has written batch files to create menus to facilitate access to software.

PC1 performs most of the tasks from the Task Checklist quite frequently (4.556), and finds them at least fairly easy (4.333). The only ratings below 4 that he gave on this checklist were for frequency of renaming a directory/folder and renaming a file, both of which he does occasionally.

On the Concept Definition task, PC1 rated himself unfamiliar only with Finder.

Program vs Document. Like PC5, he defined Program/Application in terms of "packages", but does appear to think of programs as files themselves, knowing that particular filename extensions indicate executable files (DOS question 5). PC1 seems to understand the distinction between program and document.

Storage Buffer/Clipboard. PC1 does not appear to possess the concept of Storage Buffer/Clipboard, although he can describe procedures related to a storage buffer, for example using Word Perfect's Move function. PC1 rated himself fairly familiar (4 out of 5) with Storage Buffer/Clipboard, but gave an explanation that appeared to combine the functions of the Files and Buffers statements from the DOS config.sys. file. PC1's Pathfinder network links Memory (disk and RAM) only with Storage Buffer/Clipboard, indicating that although he understands that Files and Buffers relates to system memory, he may have an exaggerated idea of their centrality. PC1's only other link for Storage Buffer/Clipboard was to Program/Application.

File Format. PC1 gave a fair definition of File Format as "the type of language (i.e., ASCII) used in the File." and his responses to DOS questions 12-14 seemed to indicate a grasp of procedures related to this concept. He was able to suggest that exporting as ASCII would facilitate using a file with another application. However, he later defined ASCII as the binary language of DOS.

Directory/Folder. PC1 defined Directory/Folder by making an analogy with filing cabinets and describing their use for separating "relevant" files. It is perhaps worth mentioning that in several cases PC1 misused more technical terms, for example describing initializing/formatting a disk as partitioning.

PC1 had little difficulty with most of the DOS performance tasks. Like several subjects, though, he used the DIR command to verify the effects of his file and directory operations. Called upon to open an unfamiliar file, he used the DOS TYPE command. Consistent with lack of a Storage Buffer/Clipboard concept, his method of moving a graphic within a word processor was to delete and re-import the graphic.

Subject PC2

Subject PC2 had had about nine years of computer experience at the time of the study. His first computer experience was with a Prime mainframe, and he rated himself very familiar (five on a scale of five) with both the IBM PC and a Hewlett Packard computer. The subject uses a DOS-based notebook computer at home. The subject regularly used nine different application types on the PC. PC2 had had five computer related courses, and a half day class on Microsoft Windows. Despite this latter class, PC2 is somewhat uncomfortable with Windows. For example, he noted that he found graphical word processors difficult to learn, and that he preferred to use a simple DOS text editor. He also complained of the lack of "a well-built database for Windows". PC2 worked as a systems analyst for 7 years, about 4.5 for Compaq, and in some respects seems to be the most technically oriented of the subjects. For example, on DOS question 10, PC2 was the only subject to mention taking care not to delete hidden files.

PC2's mean rating of the frequency on the task checklist was 4.333, and his mean rating of ease for those tasks was 4.167. That is, with a few exceptions he performs those tasks often and finds them easy. The only task that he rated difficult was inserting into a document something that originated in another program, though it is a task he performs fairly often. This may be related to lack of familiarity with the Clipboard, which he was, he said, just beginning to use. He complained that the lack of a good file manager in Windows, which reduces opening a previously created document to somewhat difficult/somewhat easy. Two tasks that PC2 also finds somewhat difficult/somewhat easy are using a brand-new disk and installing a new program, the latter because, he says, memory management takes special research. No other subject mentioned memory management in this context. The only tasks from the checklist that PC2 performs less than fairly often are rearranging directories/folders (seldom) and renaming a directory/folder (occasionally).

On the concept definition task PC2 rated himself as unfamiliar only with Finder, a Macintosh concept. The subject gave reasonably accurate definitions of most concepts, with the exception of Creator, another Macintosh concept, and Initialize/Format a disk, for which he admitted he knows how, but not what occurs.

PC2 was the only subject who was able to accomplish all of the performance tasks in DOS without difficulty.

Program vs Document. PC2 understands the distinction between program and document.

Storage Buffer/Clipboard. As noted above PC2 has made little use of the Clipboard, but he was able to describe use of the clipboard for cutting and pasting within a document (question 11) and between programs (question 12). PC2 was aware that items remain in the buffer and can be pasted in multiple locations. However, he thought, though unsure on this point, that to move a graphic between programs using the Clipboard it would be necessary for both programs to be running at the same time. In response to question 12 PC2 also described saving a Systat document in a file and later importing using a filter.

File Format. Despite having provided a very good definition of File Format in the Concept Definition task, PC2 stated that he was not really sure why an importing procedure was necessary, except that this is "as close as they got to having packages work together". However, in response to question 13 regarding a changed filename extension, PC2 was one few subjects to discuss in some detail the possibility of a changed file format and the steps this might necessitate for further use of the file. In response to question 14 PC2 mentioned the possibility that a colleague might export a file to your format, but did not specifically mention saving as text.

Directory/Folder. PC2 has a good understanding of Directory/Folder, which he described as "A theoretical place, for organization only...". He was one of the few subjects to suggest that a "Bad command or file name" message could be a consequence of typing a command in the wrong directory. He was the only subject to suggest, in response to question 7 regarding how to run a program, the possibilities of entering a complete pathname or of putting the directory of a frequently used program in the Path in the Autoexec.bat file, which specifies directories to search in an attempt to carry out a command.

Subject PC3

Subject PC3 has about eight years of computer experience, beginning on the Macintosh. He rated himself as fairly familiar with the IBM PC (four on a scale of five), and very familiar with the Macintosh (five). He uses an IBM PC compatible 80386/33 - based computer at home. He has had no computer-related classes, but uses ten different application types all with Windows on the IBM PC. He stated that he prefers the logic structure of the Macintosh, but likes the flexibility of DOS.

PC3 performs the tasks from the Task Checklist fairly frequently (mean of 3.778 out of 5), and finds them very easy (mean of 4.556). He very seldom backs up data or deletes files, and seldom saves a document to another format. He finds it difficult to insert into a document something that originated in another program.

On the Concept Definition task PC3 rated himself as unfamiliar only with Finder.

Program vs Document. He is aware of the distinction between program and document, but his definitions of File ("Work..."), Program/Application ("Software I buy that manipulates my data"), and Data File (Work files...) suggest limits to his understanding of these concepts.

Storage Buffer/Clipboard. PC3 has a good understanding of Storage Buffer/Clipboard, including the fact that it can be used between applications: "Used to move things around docs/files/and applications.", and invoked this concept at the appropriate times in answering the DOS and Macintosh questions (11-13). He is aware that some DOS programs have similar functions but that items left on the Clipboard in a DOS program are lost upon exiting the program, and not so in Windows or on the Macintosh.

File Format. He also has a good general understanding of File Format. He was one the four subjects to realize that a change of filename extension or icon might mean a change of file format (question 13). And he mentioned the possibility of saving to another format to facilitate using a document from another a different word processor (question 14).

Directory/Folder. On the basis of his definitions and responses to questions he has a good understanding of Directory/Folder.

Despite his apparent grasp of the relevant concepts, when called upon to use his knowledge he was often hesitant, and was unable to complete the tasks of importing a graphic and moving it within a word processor document. Trying to open a file of unknown format, he first looked for utilities, then attempted to retrieve the file into Lotus 1-2-3. When that failed he did not attempt to import, but decided to use Word for Windows "Because I can copy anything into that." Deciding that it looked like a word processor document, he finally elected to convert from text, and succeeded in opening the file.

PC3 had no difficulty creating a graph with Lotus 1-2-3, but was unable to import it into Word Perfect, despite recalling a "neat embedding feature" from work experience. He attempted to use Excel and Word for Windows, and created a graph, but, unfortunately, the experimental session was marred by the necessity that it be conducted at a public computing facility with networked applications. Inexplicably, at least on this occasion, the system runs one Windows application at a time, and to use another it is necessary to restart Windows. Therefore it was not feasible to use the Clipboard for this task. PC3 saved his graph to a file. Then he tried to import his previously newly created graph into and was unable to do so. He stated in explanation that he was accustomed to using Word 2.0 for Windows, which has a better Insert feature, and that he doesn't like Excel graphs.

Subject PC4

Subject PC4 has about six years of computer experience. His first experience was with an Apple II. PC4 rated himself at four on a scale of five in familiarity with the IBM PC, and unfamiliar with the Macintosh or with UNIX. He uses an IBM PC at home. He indicated that he uses five different types of application, and that he has taken one computer-related course, in programming.

PC4 had the lowest ratings of any subject for frequency and ease of routine tasks, 2.111 and 2.722 respectively on a scale of 5. PC4 does not engage in any of the actives related to directories named in this task, neither creating, deleting, rearranging, nor renaming directories. PC4 rated as difficult the following: inserting into a document, without retyping, something originating in another document created using the same program; inserting into a document something that originated in another program; and saving a document in another format.

On the Concept Definition task PC4 rated himself not familiar with Creator of a File, File Format, Finder, and Command Interpreter. He rated himself as unfamiliar (2 out of 5) with Filename Extension, Data File, Storage Buffer/Clipboard, and ASCII. PC4's responses on the Concept Definition task were brief and sketchy, suggestive of a relatively shallow knowledge, for example: "Clipboard concept is where information is stored in a different location than the program.", or for Memory (Disk and RAM) "Method of storing information. This can be used to retrieve information."

Program vs Document. PC4 appears to understand the distinction between program and document.

Storage Buffer/Clipboard. Although he does not use GUI-based software, PC4 was able to describe cut and paste operations within a program (question 11), but he did not mention a storage buffer/clipboard in this context, and seemed unaware that an item could be repasted without recopying. PC4 gave a fair if vague definition of Storage Buffer/Clipboard as "where information is stored in a different location than program." Curiously, his Pathfinder network shows Storage Buffer/Clipboard linked only to Directory/Folder, suggesting that his vague definition of Memory (disk and RAM), alluded to above, probably reflects a real confusion regarding the distinction between disk and RAM. He did not mention use of a clipboard to move a graphic into a document created in another program (question 12).

File Format. Although he later indicated unfamiliarity with the concept of File Format and defined ASCII sketchily as "Some sort of computer code", his Pathfinder network shows File Format linked to Program/Application, Directory, File, and Filename Extension. His responses suggested an ability to deal with differing file formats in practice. He justified the necessity of importing a graphic (question 12) by stating that graphics can't be created in Word Processors. He was, however, one of the few subjects to suggest that the file itself might have changed through use in another program to cause change of a filename extension (question 13). While stating that he is not very familiar with ASCII, he suggested that it might be a way to facilitate moving a document between programs (question 14).

Directory/Folder. Consistent with his ratings on the task checklist, PC4's responses the DOS questions suggested limited awareness of directories. He did not note that directory names have no extension (question 4), did not suggest searching subdirectories (questions 1, 2, 3, and 6), and did not suggest creating a subdirectory (question 9) in order to avoid confusion of files.

Despite the fact that he apparently does not spontaneously engage in activities directly related to directories, PC4 was able to create a subdirectory as one of the DOS performance tasks. Like many of the subjects in this study, he was baffled at how to open an unfamiliar file type, indicative perhaps of a partial grasp of file format or of difficulty in knowing when that concept is relevant. Cued to try a program, he used the retrieve command in Word Perfect. Seeing that the file contained numbers, he concluded that it was a spreadsheet file, and attempted to retrieve it in Lotus 1-2-3. After failing three times with the retrieve command he imported the file as text. He was able to carry out the remaining performance tasks without difficulty, with the exception of moving an imported graphic within Word Perfect. He did, however, demonstrate that he knew how to cut and paste text within Word Perfect.

Subject PC5

Subject PC5 has had about three years of computer experience. His first experience was with an IBM PC. He rated himself somewhat familiar with the IBM PC (3 out of 5), and slightly familiar with the Macintosh and VAX (2 out of 5). PC5 uses an IBM PC at home, and uses 3 types of applications. He has had one programming course. PC5 rated his ease of learning new programs somewhat easy/somewhat difficult, and was one of only three subjects to rate ease of learning below moderately easy. Despite having a PC for home use, PC5 noted that he feels "Mac is better for the common user.".

On average, PC5 performs the tasks from the task checklist fairly frequently, 3.944, and finds them fairly easy, 4.167. He only occasionally inserts into a document something that originated in another program, saves a document in another format, rearranges directories/folders, or renames directories/folders, all of which he finds somewhat easy/somewhat difficult. He very seldom inserts into a document something originating in another document created using the same program, seldom renames files, but often copies documents, and also finds these operations somewhat easy/somewhat difficult.

In general PC5's written responses were rather brief. PC5 rated himself unfamiliar with Creator of a File, File Format, Finder, and Command Interpreter. He rated himself as somewhat familiar with Filename Extension, and on the Concept Definition task confined himself to a description ("The three letter endings of files").

Program vs Document. PC5 described Program/Application simply as "Different software packages", but his answers to the DOS and Macintosh questions indicated a clear understanding of the distinction between program and document.

Storage Buffer/Clipboard. His definition of Storage Buffer/Clipboard, with which he rated himself somewhat familiar, mentioned only transferring text rather than more general copy, cut and paste operations. Nonetheless, he invoked cut and paste operations in response to Macintosh question 12 regarding bringing a graphic into a word processing document. However, he stated that both programs must be running at the same time. In PC5's Pathfinder network Storage Buffer/Clipboard is linked only to Data File and File.

File Format. Although he rated himself unfamiliar with File Format, he knew that something saved to DOS text is likely to be readily usable between programs (DOS question 14). However, on question 13 it did not occur to him that a document with a changed filename extension (DOS) or icon (Macintosh) might have a changed format. He did not invoke saving to a standard format on the Macintosh version of question 14, but noted that conversions between word processing programs had always been carried out automatically, in his experience. In PC5's Pathfinder network, File Format is linked only to File.

Directory/Folder. PC5 appears to have a good understanding of Directory/Folder.

When called upon to open a file of unknown format in the DOS tasks, PC5 used the DOS type command for text files. Called upon to open a text file with Lotus 1-2-3 he first tried unsuccessfully to retrieve, and had to be cued to try Import and then to import as numbers in order to make a graph, all suggesting limitations to PC5's grasp of procedures related to File Format. Unfortunately, repeated system errors interrupted performance of the remaining DOS tasks, and the experimental session continued with the Macintosh performance tasks. PC5 indicated that to move a graphic within a DOS word processor he would use cut and paste.

Confronted with an "Application Busy or Missing" on attempting to open a Macintosh file, PC5 hazarded a guess that it might be a word processor file, ran Microsoft Word, and was able to open the file.

Seeing what looked like a Spreadsheet, PC5 assumed that he had been able to open an Excel file with Word. PC5 was unable to complete the remaining Macintosh performance tasks due to unfamiliarity with Excel.

Subject PC6

Subject PC6 has about 10 years of computer experience, beginning with a workstation for a mainframe of a type he does not recall. PC6 rates himself as fairly familiar (4 on a scale of 5) with the IBM PC, and slightly familiar (2 out of 5) with the Macintosh. He uses an IBM XT at home, uses six different types of application, and has taken two computer related courses. He rates learning to use a new program as somewhat easy/somewhat difficult, one of three subjects to rate this below moderately easy. PC6 noted that all that he knows about programs he has learned on his own. He stated that he has been confused in Windows, and that he is not sure of the purpose of the little boxes in the corners.

On average, he performs the tasks on the Task Checklist at least occasionally (3.278), and seems to find them fairly easy (3.778). He very seldom backs up data or renames a directory/folder and seldom installs a new program or deletes or rearranges directories/folders.

On the Concept Definition task PC6 rated himself unfamiliar with Storage Buffer/Clipboard, File Format, Finder, Command Interpreter, and ASCII.

Program vs Document. He seems to understand the distinction between program and document. PC6's Pathfinder network was quite sparse, and perhaps should not be over-interpreted, but it seems noteworthy that in his network File has only one link, to Memory (Disk and RAM).

Storage Buffer/Clipboard. As indicated by his familiarity rating, he apparently has no concept of Storage Buffer/Clipboard, though he is aware of procedures in Word Perfect for blocking, moving, and copying (question 11). In fact he recalled the appropriate function keys as he did also for importing a graphic (question 12).

File Format. PC6 evinced no abstract understanding of the concept of File Format as such. As noted, he was unfamiliar with the term. He defined Filename Extension by "Designates type of file" but in answer to question five he mentioned only the filename extensions that denote executable files, saying that "The others are data files, I guess." However, he has some awareness of the practical implications of different file formats and the fact that they are associated with different programs and differ in how they are stored. In response to question 12, regarding importing graphics, he mentioned exporting to files readable by Word Perfect, and knew, in response to question 14 that many word processors are able to import files created using other packages, and that it is possible to "design a file stored in certain readable language", and even took a stab at ASCII (ACRS?). However, it did not occur to him in responding to question 13 that a change in filename extension might be associated with a change in format, and possibly create problems for use of the file. Consistent with a limited knowledge of file format, PC6 had difficulty opening a file of unfamiliar type in the DOS performance tasks, attempting to use Retrieve rather than Import. After a hint on this topic, despite some unfamiliarity with the programs involved PC6 was able to accomplish all of the tasks except number 6, moving the graphic within a word processor document. He attempted to block it and move the block, but was unable to do so. However, he demonstrated that he was able to use this method to move text.

Directory/Folder. PC6 has a good understanding of Directory/Folder, though it did not occur to him in response to question 4 that names without filename extensions might designate directories.

Macintosh Users

Subject M1

Subject M1 had had about four years of computer experience at the time of the study. His first computer experience was with a UNIX workstation, and he rated himself very familiar (five on a scale of five) with both UNIX and Macintosh, moderately familiar (three out of five) with DOS, and somewhat familiar (two out of five) with the VAX. M1 was the only subject in the study with no computer at home. He has had three computer programming courses and one CAD course. M1 listed five application types, but only one, desktop publishing, for the Macintosh, his computing platform of choice at the time of study. M1 noted that although the Macintosh was "what I prefer and know", most of his needs had been simple.

On the checklist of routine tasks M1 rated every task that he performs as very easy to accomplish on the Macintosh, but rated himself as performing some tasks seldom or never, resulting in the second lowest mean frequency rating for these tasks, 2.556, among the subjects in this study. In particular, M1 never installed a new program, or deleted or rearranged directories/folders, very seldom saved a document in a different format, and seldom backed up data or copied documents.

On the concept definition task M1 rated himself as unfamiliar with five concepts: Creator of a File, Data File, File Format, Finder, and Command Interpreter.

Program vs Document. On the basis of his definitions, M1 grasps the distinction between program and document, but appears to make a distinction between program and file, confusing file with data file and regarding programs as creating or modifying files. Program/Application is not linked to any of the file related concepts in M1's Pathfinder network, a fact that also suggests that M1 does not understand the relation between application and file format.

Storage Buffer/Clipboard. When asked about moving text within a word processor, he immediately invoked the idea of a buffer for both platforms, though he was more sure of procedures for the Macintosh. He also knew that an item remains on the Clipboard until another operation affects the Clipboard. However, M1's mastery of Storage buffer/Clipboard concept and related procedures seems uncertain or incomplete. He could not remember the name Clipboard when answering the Macintosh questions, and in answering question 12, regarding bringing a graphic into a document, first described "importing" from the Scrapbook. He then mentioned that if you have Finder, perhaps meaning MultiFinder, it is possible to copy and paste. However, he appeared to think that it would be necessary to close the application in which the graphic originated before pasting it into a document in another application. When called upon to perform this task in the final session, M1 transferred a graph from Excel to Microsoft Word via the Scrapbook, though MultiFinder was running on the computer used for the task.

File Format. M1 rated himself as very familiar with ASCII, and knew it as a standard set of symbols. However, he described it as useful for communication between computers, apparently not thinking in terms of communication between programs. He did not think of saving documents in other formats to facilitate use with another program (question 14). Furthermore, it did not occur to him that a change of suffix (DOS) or of icon (Macintosh) might indicate that a file had been changed and might no longer be able to be opened by a particular program (question 13). Nonetheless, M1 apparently is able to perform some tasks related to the concept of File Format. He was aware that many programs have translators that enable them to open files created by other programs (question 14). when called upon to open a file of unidentified provenance, M1 elected to run Microsoft Word "just for fun" and successfully used this program to open the file.

Directory/Folder. His grasp of Directory/Folder, which he described as "a subspace or space containing files or other directories" appears somewhat functional but not complete. Although in answers to several of the early DOS questions he mentioned using the DIR command and searching directories and subdirectories, which he referred to as folders, M1 did not think of creating a new subdirectory when asked how to avoid mixing up a new program with anything else (question 9). In answer to the same question among the Macintosh questions, M1 thought of putting a program in its own folder or directory, and mentioned that the same thing would go for IBM. His definition of Initializing/formatting a disk was similarly vague and incomplete, emphasizing the need to make the disk recognizable to the computer or drive.

Subject M2

Subject M2 has three to four years of computer experience, beginning with a mainframe. She rated herself somewhat familiar (three on a scale of five) with the IBM PC, slightly familiar (two on a scale of five) with the Macintosh, fairly familiar (four on a scale of five) with VM/CMS on an IBM 3090, and somewhat familiar with the VAX 880. She uses a Macintosh SE/30 at home, uses four different application types, and has taken two computer related courses. She was the only subject to rate her ease of learning to use new computer programs five out of five, very easy.

On the initial experience questionnaire M2 noted a few reasons for preferring DOS to the Macintosh, including the ability to concatenate files, a preference for keyboard control in, for example viewing directories, the ability to do wildcard deletes, and, on the Macintosh, irritation at switching between keyboard and mouse, and bafflement at how a file opens an application. However she noted several reasons for liking the Macintosh, including ease of learning/interactivity, ease of switching directories/folders and ease of moving files and folders. She stated that the more she used the Mac, the more she liked it. At the beginning of a later experimental session, M2 greeted the experimenter with the news that she was now "A committed Mac user."

On average, M2 performs the tasks on the Task Checklist fairly often (4.111 out of 5) and finds them easy (4.667 out of 5). However, she never saves a document in a different format and only seldom installs a new program or deletes a directory/folder. She only occasionally creates or renames a directory/folder. She often rearranges directories/folders, and finds it very easy on the Macintosh, but difficult on the IBM PC.

On the Concept Definition task M2 rated herself unfamiliar only with Command Interpreter.

Program vs Document. M2's definitions suggested that she understands the distinction between program and document. For example she described a File as "a document or product created through another program." But she went on to say that a file could be "Any unit of a program." While somewhat ambiguous, this last statement at any rate seems to indicate that she is aware that programs are also files.

Storage Buffer/ Clipboard. M2 also seems to have a good understanding of Storage Buffer/Clipboard, which she defined as "Temporary storage of an area of a file that has been defined by the user." In response to questions 11-12 she was able to accurately describe, and later performed, cut, copy and paste operations within DOS word processors, and within and between Macintosh applications, and was aware that an item remains on the Clipboard until the next operation involving the Clipboard. She noted that she was at first scared to cut, and would copy, paste, and then cut, not knowing "Where it went." Then when she learned that it went to the Clipboard, "It was

OK." She considers the Scrapbook confusing, because "You can't see all that's on it."

File format. M2's definitions indicate that she has a good basic understanding of File Format, and is aware that ASCII is a standard, but she is apparently not entirely able to apply the concepts in all appropriate circumstances. Although she knew in response to DOS question 12 that it would be necessary to import a graphic into a word processor document ("different texts" in her phrase), she could think of no response to question 14 for DOS regarding facilitating use of the same document with different word processors. For the Macintosh version of question 14 she suggested establishing parallels, e.g., "same icon, etc.", and admitted she did not know how to insert, for example, a Word Perfect document into Word. Along the same lines, she did not seem to fully understand the implications of question 13, thinking that she would somehow need to restore a changed filename extension or icon to the original form to be able to use the file, without noting the possibility of changed file format. She did, however, as an afterthought to the DOS version of question 13, suggest "making sure the word processors cross-compile.". When called upon in the performance tasks to open a file of unknown format, she used EDLIN in DOS and simply double-clicked on the Macintosh. It is perhaps worth noting that her method of exiting EDLIN was a reboot. When double clicking the unknown file resulted in the "Application busy or Missing" error message, she immediately and successfully tried to open the file using Microsoft Word. This could be an indication of past experience with text files, but as discussed under the section on misunderstandings, M2 seems to have a rule for such situations that opening the application first is more reliable. When later specifically requested to open a prepared text file to create a Word document, she first tried double-clicking on it, then opened it from within Word.

Directory/Folder. M2 gave a good definition of directory/folder, and was able to answer all questions and perform all aspects of tasks involving this concept.

M2 attempted both the DOS and Macintosh performance tasks. She proved to be unable to proceed with the DOS tasks beyond the point of attempting to import a previously created graphic into a Word Perfect document. She was able to complete the Macintosh performance tasks quite quickly, apart from the attempt to open a text file by double clicking on it. Called upon to predict the quality of printed output from a SuperPaint Paint layer document, she simply guessed "Probably pretty good."

Subject M3

Subject M3 has had about 8 years of computer experience, first using an Atari 800XL. M3 rated himself as very familiar with the Macintosh (5 on a scale of 5), somewhat familiar with the IBM PC (3 out of 5), slightly familiar with UNIX (2 out of 5), but fairly familiar with the TI 99 (4 out of 5). M3 began using a PC about 6 years ago, and the Macintosh, with which he now feels more familiar, about two years ago. M3 uses a Macintosh at home. M3 uses 5 different application types, and has had 3 computer related courses.

In general M3 performs the tasks from the Task Checklist frequently (mean rating of 4.389) and finds them easy (mean rating of 4.889). He only occasionally backs up data or installs a new program (3 out of 5). M3 seldom saves a document in a different format.

M3 did not rate himself unfamiliar with any of the concepts from the Concept Definition task, but rated himself only somewhat familiar (3 out of 5) with Data File, Command Interpreter, and ASCII. M3 was one of few subjects to give a definition for Finder that did not confuse it with Find File or the Operating System.

Program vs Document. M3 understands the distinction between program and document. M3 was the only subject to specifically mention in defining File that files can be executable (programs) or not (documents). Nonetheless, M3 described Data File as "...probably a DOS .dat file".

Storage Buffer/Clipboard. M3 appears to have a good understanding of Storage Buffer/Clipboard, though he mentioned in interview that he still gets Clipboard and Scrapbook mixed up, and has had problems with "strange things" happening when he pasted into a word processor from the Scrapbook. M3's Pathfinder network shows Storage Buffer/Clipboard linked to Operating System and Memory (disk and RAM). He is aware that the most recent item remains in the Clipboard, but stated in answer to question 11 for DOS that he would re-highlight to be safe if using a DOS word processor and repeating a paste.

File Format. M3's definition of File Format distinguished broad types of files ("executable, text, data, etc.") and noted that file format is characterized by extension on DOS machines and icon on the Macintosh. However, M3 did not mention any relation of format to creator or internal differences in files. On neither the DOS nor the Macintosh questions did his response to question 13 suggest that a file might be altered in format, in particular by being saved by a different program, though M3's Pathfinder network shows File Format linked to Filename Extension.

Directory/Folder. M3's discussion of Directory/Folder was rather procedural in tone, focusing on their use for arranging files and other folders. He stated on the Experience Questionnaire that folders on the Macintosh had enabled him, by analogy, to understand directories in DOS. In M3's Pathfinder network, Directory/Folder was linked directly only to Filename Extension.

M3 attempted both the DOS and Macintosh performance tasks. Unable to retrieve a text file in Word Perfect for DOS, he searched the menus and found and successfully used the Import option. M3 then used Import to open a spreadsheet file as text in Lotus 1-2-3, and attempted unsuccessfully to graph that text. M3's unfamiliarity with the relevant DOS programs led him to suggest moving on to the Macintosh performance tasks, and he was allowed to do so, completing all tasks very rapidly. Called upon to predict quality of printed output from a graphic that he had imported into the Paint layer of SuperPaint (Macintosh task 7), M3 mentioned only that he had seen poor output from MacPaint, but SuperPaint seemed a more sophisticated program, and he thought that it should print better.

Subject M4

Subject M4 has about 8 years of computer experience, beginning with a Macintosh. This can be taken as an upper bound, since the Macintosh came to market in 1984, and this study was conducted in fallwinter of 1991. M4 rated herself very familiar with the Macintosh (5 on a scale of 5). and somewhat familiar (3 on a scale of 5) with VAX and Epsom (sic). M4 uses a Macintosh at home, uses five different types of application, and has taken no computer related courses.

M4 performs the tasks from the task Checklist fairly frequently (4.167 out of 5) and finds them fairly easy (4.25 out of 5).She has never tried to save a document in a different format, and finds it difficult to somewhat difficult /somewhat easy (2 to 3 out of 5) to switch between programs.

M4 is distinguished by an ability to accomplish routine tasks with little knowledge of computer concepts and intended procedures. M4 rated herself unfamiliar with Creator of a File, Filename Extension, Data File, Storage Buffer/Clipboard, Operating System, Finder, Command Interpreter, ASCII, and Memory (disk and RAM).

Program vs Document. M4 seems of all the subjects to have the least clear understanding of the distinction between program and document. She defined File as "A group of data/information saved in a category.", and Program/Application as "The data which allows a file to run-or be created and saved." When asked (Macintosh question 5) what icons and names might indicate programs, she responded that "it usually says on the next description thing," referring to viewing by name, apparently her preferred view. In a later, session, attempting to open a document of unknown format, she used Get Info and stated that she does not know what a document is. Prompted with the question "Is there an application that might be able to open it?", she replied that she didn't know how to go into a file from another file. However, she seems to understand the use of the term document in some contexts. For example, she was able to create a new Word document when instructed to do so.

Storage Buffer/Clipboard. M4 rated herself unfamiliar with Storage Buffer/Clipboard. In the preliminary interview she stated that she is just learning the Clipboard and Scrapbook. However, she knows how to highlight, which she calls darken, and copy and paste within applications, without apparently realizing that this involves the Clipboard (question 11). She uses the Scrapbook to copy between applications (question 12) and is also able to copy and paste from the Scrapbook without realizing that the Clipboard is involved. As M2 originally did, M4 stated that she always uses a sequence of copy, paste, then delete instead of cut and paste, "In case of mistake."

File Format. M4 appears to have at least some grasp of File Format, which she defined as "The way a file is created - and how it will appear on paper." It did not occur to her that a file might have changed along with its icon (question 13), but her answer to question 14 regarding using the same document with different word processors indicates at least an awareness of and experience with possible problems. She replied "If I put it into my computer would it be scaled down? Or would it say "Will print as data" and give funny squares and blocks?"

Directory/Folder. M4's knowledge of Directory/Folder, which she defined as "A list of files within which you can choose from." also seems limited. When asked (Macintosh question 4) the difference between pictures like a sheet of paper with a bent edge and pictures like a file folder, she replied that she would view by name. In response to question 9 she suggested avoiding mix-ups by putting it (a new program) in a folder, but did not suggest creating a folder. When asked in the Performance tasks to create a new folder, she was initially unable to do so, but found and used New Folder under the File Menu.

M4, despite her apparently limited knowledge of a number of computer concepts, and procedures, was able to accomplish the Macintosh performance tasks, after discovering how to create a folder and an initial hint to use an application to open a file of unknown format. As mentioned above, she used the Scrapbook to copy a chart from Excel to a Word Document. Finally, she was unable to predict the output quality of a graphic from the paint layer of SuperPaint, saying only that she thought it would look like it did on the screen.

Subject M5

Subject M5 has about seven years of computer experience, beginning with an IBM PC. He rated himself as very familiar with the IBM PC, somewhat familiar with the Macintosh (three on a scale of five) and slightly familiar (two on a scale of five) with UNIX, a Prime System, an IBM mainframe, and a VAX system. M5 uses a Macintosh IICi at home, uses 9 different application types, and has had five computer related courses. M5 considers DOS flexible and powerful, but considers some aspects, such as writing batch files, hard to learn and use. However, he considers the Macintosh difficult to program.

M5 performs the tasks from the task Checklist occasionally to fairly frequently (mean of 3.444 out of 5), and finds them very easy (mean of 4.833). He seldom saves a document in a different format, but finds it fairly easy, and seldom rearranges directories/folders, which he finds somewhat difficult/somewhat easy on the IBM PC, but fairly easy on the Macintosh.

On the Concept Definition task M5 rated himself unfamiliar with Creator of a File, Command Interpreter, and ASCII.

Program vs Document. M5 is aware of the distinction between program and document, as evinced by his correct answer to Macintosh question 6 regarding starting a program by clicking a document. His definitions from the Concept Definition task seem to emphasize programs and functions. For example, he defined File as "Smallest independent executable subunit. Components of programs and product of programs." And he defined Filename Extension as "Logical descriptor of what the file does. .BAT makes a batch file. Usually the extension implies a function."

Storage Buffer/Clipboard. M5's definition of Storage Buffer/Clipboard was very general, but his correct answers to questions 12-13 for both DOS and Mac erase any doubt that he understands the specific functions of the Clipboard or the buffers implemented in various DOS programs. M5's concept relatedness ratings, like those of several other subjects, produced a Pathfinder network in which Clipboard is linked to Finder, consistent with his definition of the Finders as the operating system on the Mac.

File Format. M5 gave a very good definition of File Format, though he was unfamiliar with ASCII. His explanation of the need to import graphics: "You can't merge files piece by piece" (question 13) made no mention of file format, but he did mention specifically saving in a different format as a way to facilitate using the same document with different word processors (question 14). However, it did not occur to him that a change of filename extension or icon might indicate a change of file format, despite a link between Filename Extension and File Format in his Pathfinder network. His response to question 13 was to use the file as is or rename it (DOS) or to change back the icon using ResEdit (Macintosh). When called upon to open a file of unknown format, M5 used the DOS TYPE command and, on the Macintosh, double-clicked the file. M5 misinterpreted the resulting Macintosh error message, but nonetheless immediately used MS Word to open the file.

Directory/Folder. M5 evinced a good understanding of Directory/Folder. He was one of the few subjects to suggest that a "Bad command or file name" message in DOS might be the result of a correct command in the wrong subdirectory.

Like a number of other subjects, M5 was unable to complete the DOS version of the Performance Tasks. He was unable to import the graphic into a Word Perfect document. However, he completed the Macintosh version of the task without incident, apart from misinterpreting the "Application busy or Missing" error message. Like the other Macintosh-using subjects, he was unable to predict print quality from a bit-map in SuperPaint.

Subject M6

M6 has about nine years of computer experience, beginning with a Commodore "Pet" computer. He rated himself somewhat familiar with the Commodore Pet, the Apple IIe, and the IBM PC (3 out of 5), and fairly familiar with the Macintosh (4 out of 5). M6 uses a Macintosh IICi at home. He uses five different kinds of software and has had two computer classes. Although he rated himself unfamiliar with UNIX, one of his classes involved programming on a SUN, and programming in Scheme on a SUN is one of the five application types that he listed.

M6 performs the tasks on the Task Checklist occasionally to fairly often (mean of 3.556 out of 5), and finds them all very easy (mean of 5). He seldom backs up data, saves a document in a different format, or renames a file.

M6 rated himself fairly familiar (4 out of 5) with Filename extension, Data file, Storage Buffer/Clipboard, and Command Interpreter. He rated himself very familiar with all other concepts from the Concept Definition task.

Program vs Document. M6 appears to understand the distinction between program and document, and gave one of the more interesting definitions of Program/Application as "A processor or transformer of information."

Storage Buffer/Clipboard. He has a good understanding of Storage Buffer/Clipboard, and was the only subject to mention that it "can be shared globally", which applies to the Macintosh or to Windows, but not necessarily to the DOS programs. However, he was unsure whether Windows supports cut and paste for graphics as well as text. In M6's Pathfinder network, Storage Buffer/Clipboard is linked only to Finder, which he defined as "Operating system." His definition of Memory (disk and RAM) suggests that he is aware of the importance of memory, yet in his Pathfinder network Memory is linked only to Initialize/Format a disk.

File Format. He has a good understanding of File Format, as "the manner in which the file is saved. Pertains to the application it was used on, usually.", and invoked the idea of file format in response to the appropriate DOS and Macintosh questions, 13-14. However, he did not mention the possibility of importing a graphics file into a DOS word processor in response to question 12.

Directory/Folder. M6 has a good understanding of Directory/Folder.

M6 attempted both the DOS and Macintosh performance tasks, but had considerable difficulty with the DOS tasks. Asked to open an unfamiliar file, he used EDLIN, the clumsy precursor of the present DOS text editor, but was unsure how to use it. Given the hint to use an application, he concluded on the basis of a meaningless filename extension (.use) that it was probably a Word document. Seeing numbers, he concluded that it was a spreadsheet file, and imported it as text into Lotus 1-2-3. Cued to use a real Lotus file, he retrieved the file and created and saved a graph, but was unable to import the graph into a Word Perfect document. The session continued with the Macintosh tasks.

Asked to open an unfamiliar file, he double-clicked it and got an "Application Busy or Missing" message, and decided to try MS Word. Seeing numbers, he once again, without being instructed to do so, switched to a spreadsheet. He was able to complete the remaining tasks without incident.

Unclassified User

Subject B1

Subject B1 has had about 8 years of computer experience, beginning with an Apple II. She rated herself somewhat familiar (3 on a scale of five) with both the IBM PC and the Macintosh, and slightly familiar (2 out of five) with UNIX. She uses a PC compatible at home, but stated that it was not her choice and she would prefer a Macintosh. She uses eight different types of applications, and has had courses in two programming languages. She rates her ease of learning new programs as somewhat easy/somewhat difficult (3 out of 5), one of three subjects to rate ease of learning less than moderately easy. Her mean ratings for frequency and ease of performing the tasks from the Task Checklist are the third lowest, 3.111 and 3.778, respectively. According to her ratings, she seldom backs up data, installs new programs, creates directories/folders, rearranges directories/folders, or renames directories/folders.

On the Concept Definition task B1 rated herself unfamiliar with Creator of a File, Finder, and Command Interpreter.

Program vs Document. B1 understands the distinction between program and document, but B1 doesn't appear readily to think of programs as files themselves. She invoked "data that a program can work with" in her definition of File, and mentioned only extensions identifying the type of program that created a file in discussing Filename Extension. Similarly, on DOS question 5, she did not mention extensions that typically identify executable files, nor did she speculate about finding such files in her answer to question eight, regarding installing a new program without the aid of documentation. In answer to question seven, asked how to run a program she did not mention finding the name of the executable file. Rather, she said that you "Have to know what to type". Still, her definition of Data File was "Just contains data-not executable," at least suggesting that she does know that programs are files.

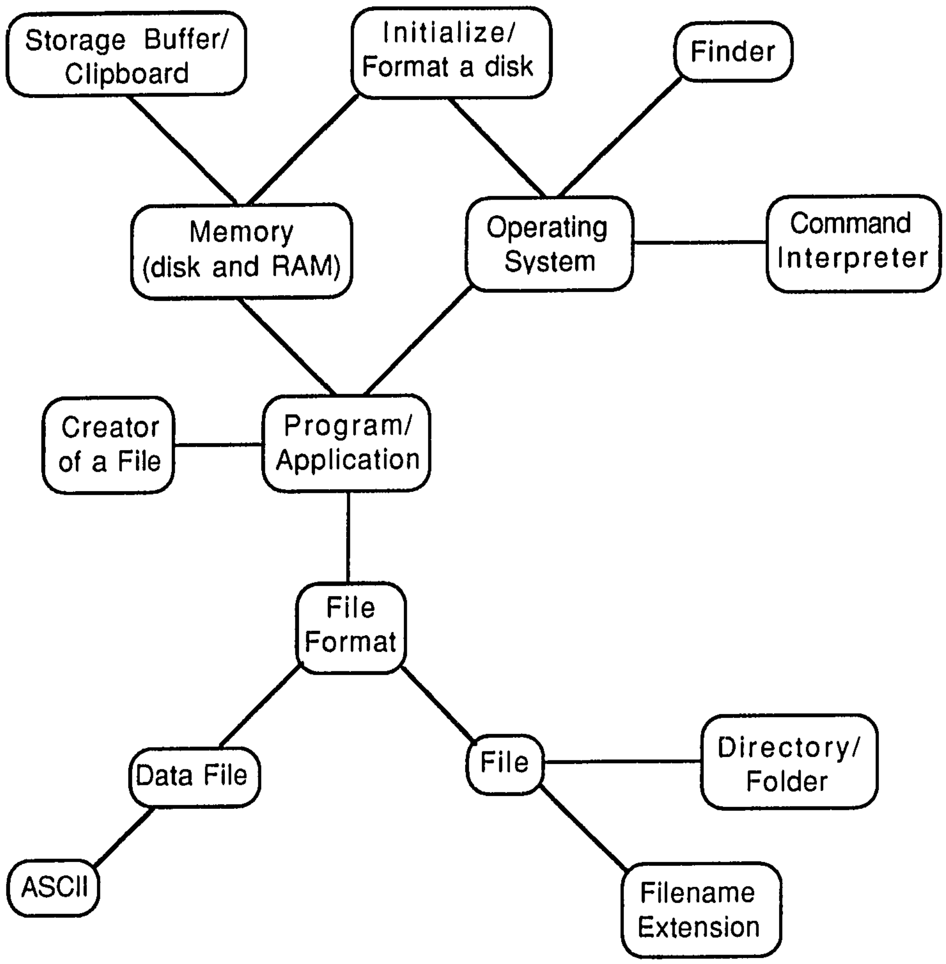
Storage Buffer/Clipboard She seems to have some understanding of the concept of the Storage Buffer/Clipboard, but may think of it as a feature of particular applications, and may think of it as limited to text. On DOS question 11 regarding copying or moving text within an application she mentioned blocking text and moving or copying to multiple locations, then as though discussing a completely separate concept, mentioned a "Scratchpad type thing in Word on the Apple...Mac or Windows sticks it off on a Clipboard that you can access." When asked about bringing a graphic into a word processor document (question 12) she mentioned only importing and dealing with possible difficulties and a necessity to "decode" if the graphic was created in another package. Even on the Macintosh questions she did not indicate that it might be possible to accomplish this using the Clipboard. However, She did just this in the performance tasks.

File Format. As may already be apparent, she seems to have a good abstract grasp of File Format, and knows that ASCII is a standard format that might be used to facilitate using documents with multiple word processors (question 14). But she was at a loss as to how to open a file of unknown format in the performance tasks, until given a hint. This is in spite of the fact that she stated in the preliminary interview that normally opens files from within an application on the Macintosh. On question 12 she was aware that a change of filename extension or icon might be accompanied by a change of format.

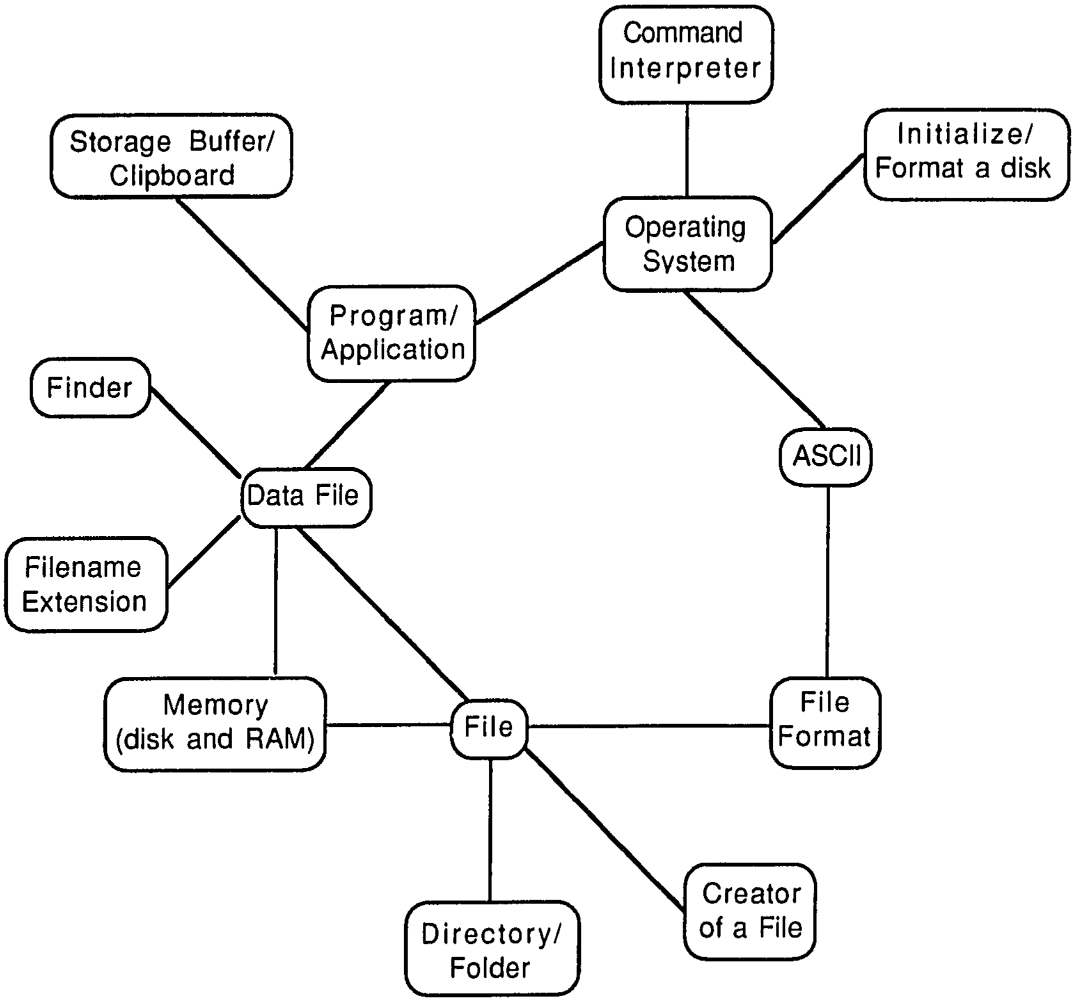
Directory/Folder. Her grasp of Directory/Folder is somewhat problematic. She defined the concept as "Different areas..." and seems able to use pre-existing directories for organization, for example adding that she would put all data files for Systat in the Systat directory. As previously noted, she indicated on the task Checklist that she seldom performs routine operations directly involving directories, and as a matter of fact was unable to create a new directory or folder for the performance tasks. Further, she did not suggest creating a directory in order to avoid confusion with files related to other programs for question 9. Nor did she reply that names without extensions in DOS might indicate directories in DOS. In response to Macintosh question 1 she thought that when you click on a disk you "get little icons and folders", but continued that she would need to "sit and play" to be certain. Similarly, in response to Macintosh question 3, She thought that folders might be like data files on the Macintosh, but said that she would have to sit in front of the screen to be sure. Nonetheless, when she did sit in front of the screen for the Macintosh Performance Tasks, she was able to open an Excel folder in order to find and open Excel without hesitating. Despite the limitations of her ability to manipulate directories, Directory/Folder is a central concept in her pathfinder network, with links to Data File, File, File Format, Finder, Filename Extension, and Program.

B1 used Windows to run DOS programs for the DOS performance tasks. Unable to complete the tasks using DOS programs, she did so using Windows and the Clipboard. B1 was able to complete the Macintosh tasks once given a hint to open a file of unknown format using an application. Interestingly, she closed Excel prior to opening MS Word, remarking that she doesn't know how to open multiple applications. She thought that printed output from the Paint layer of SuperPaint would look just like the screen display, though she later stated that she had heard of vector versus bit-mapped graphics.

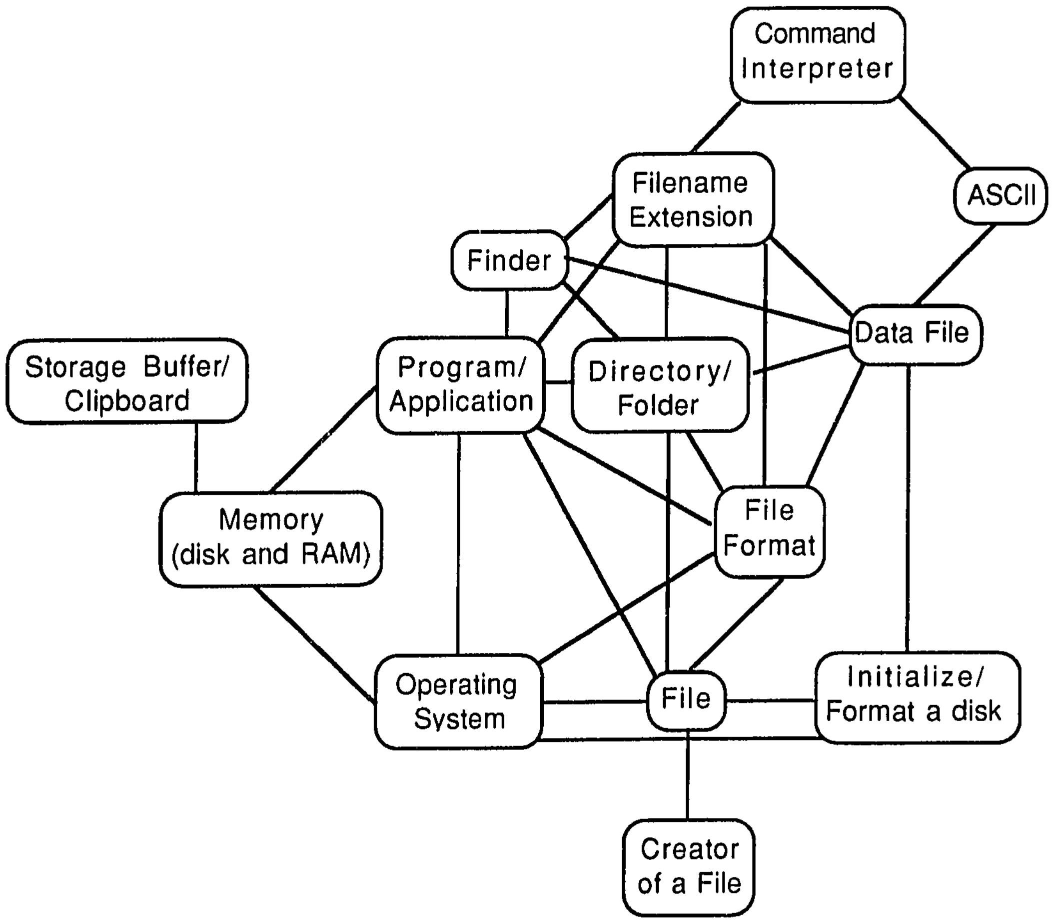
Appendix C: Pathfinder Networks



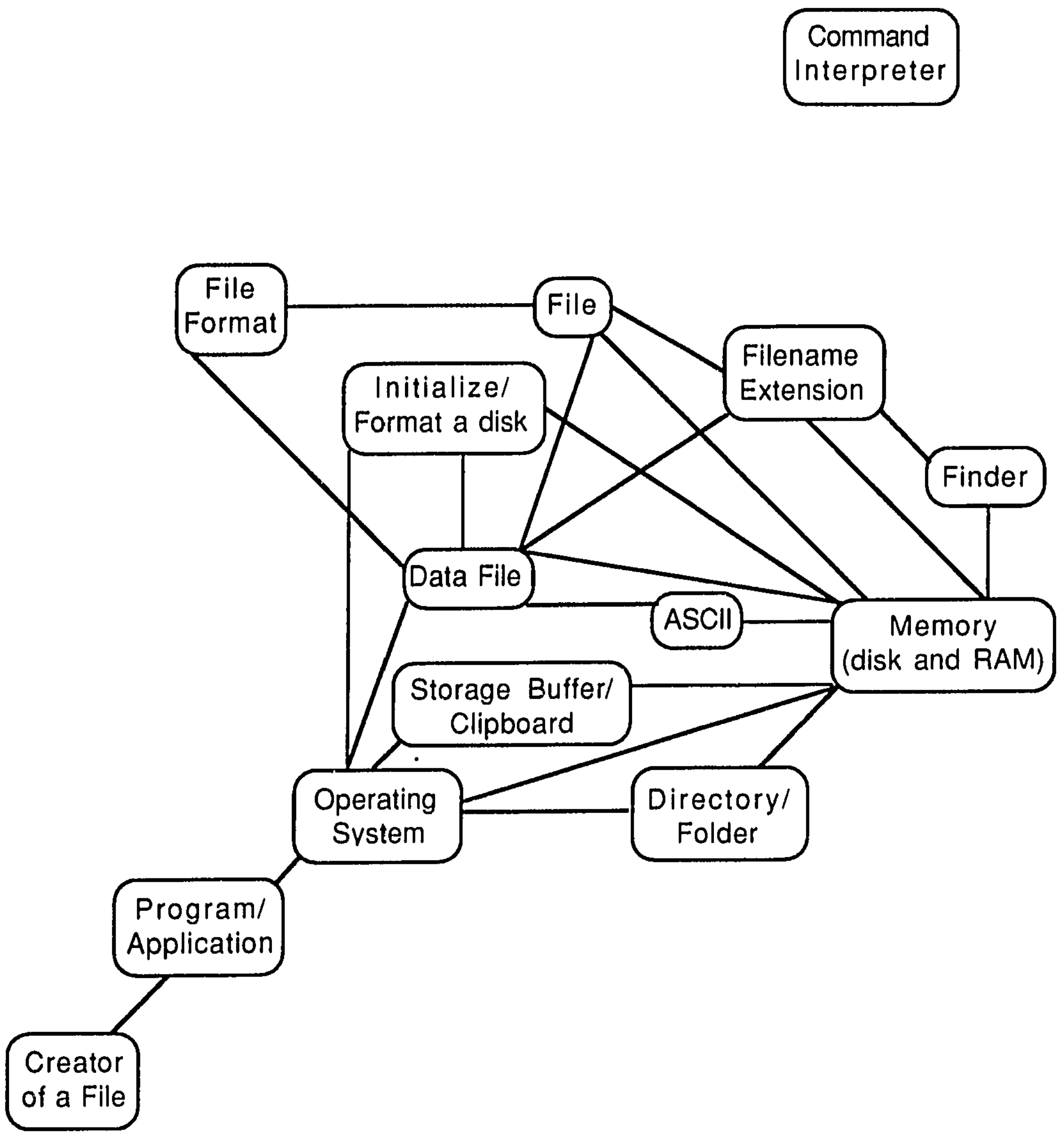
Pathfinder Network for Macintosh Users' Mean Relatedness Ratings



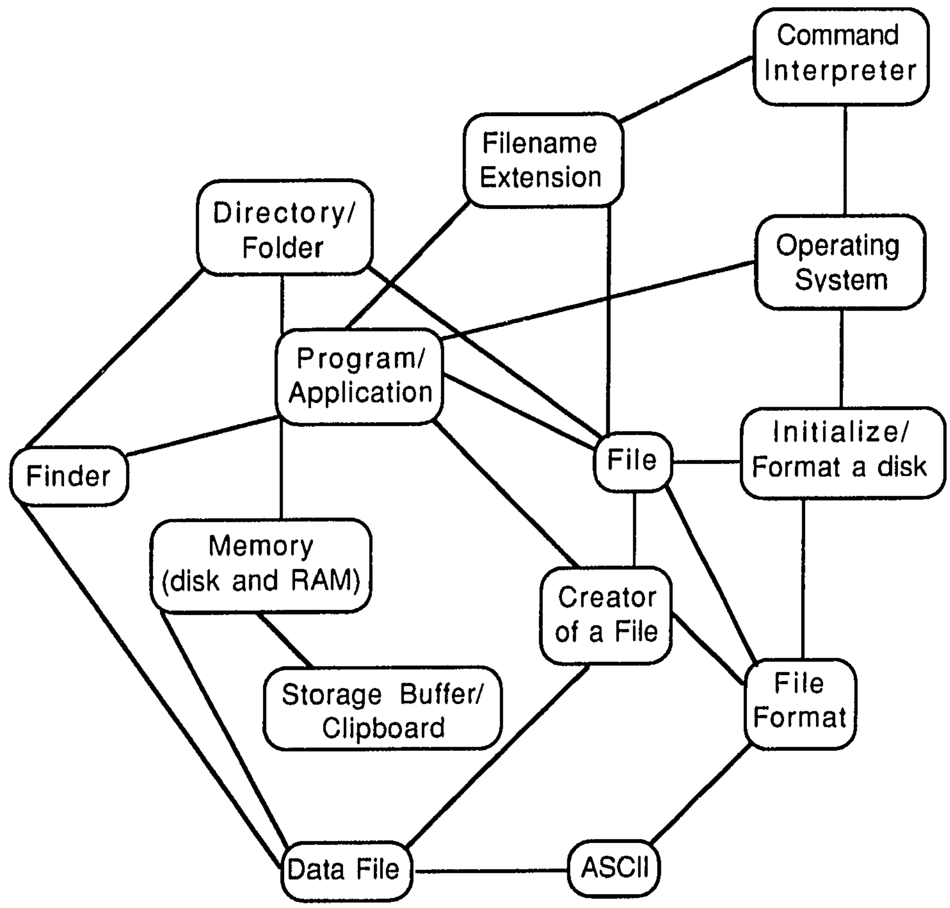
Pathfinder Network for PC Users' Mean Relatedness Ratings



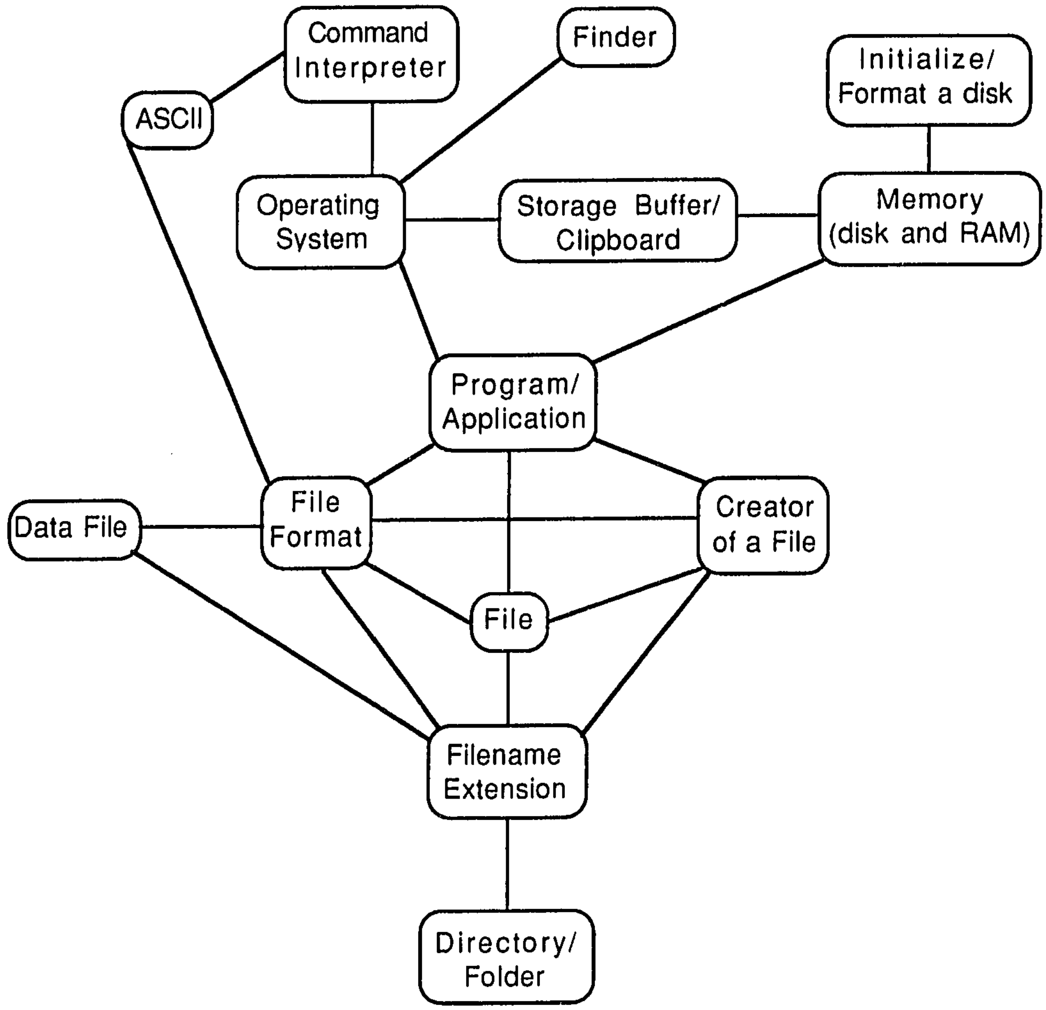
Pathfinder Network for Subject B1



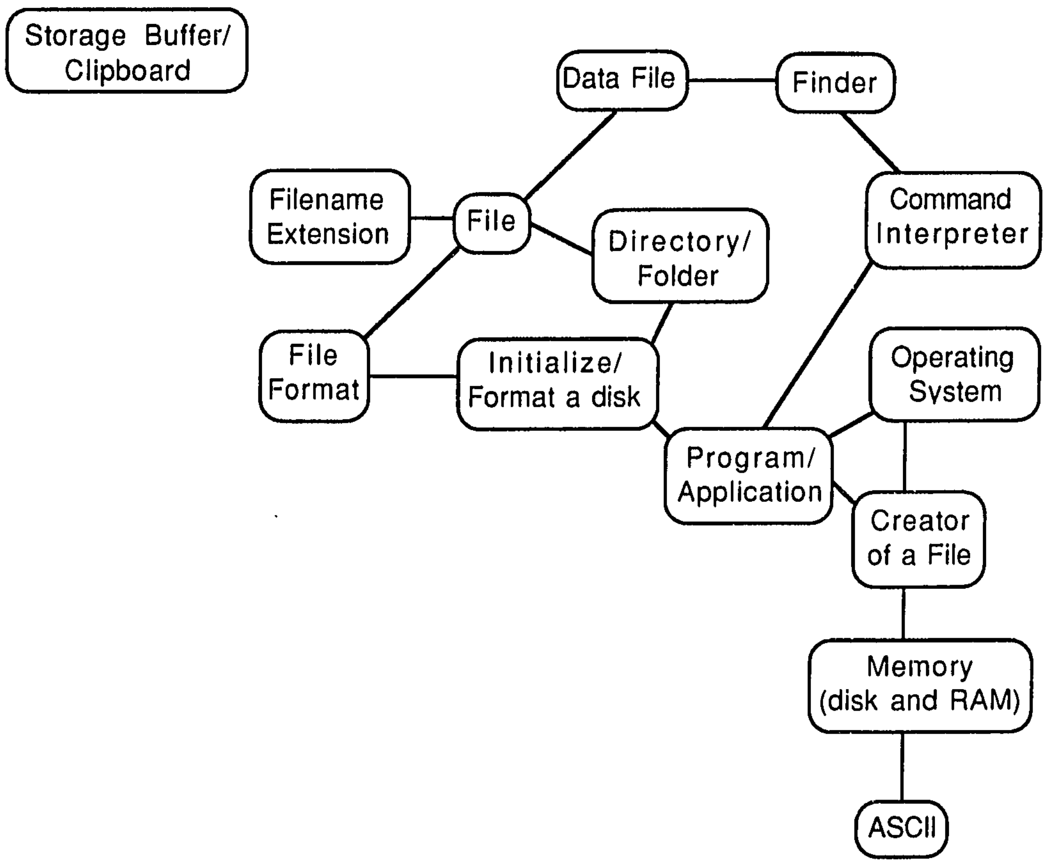
Pathfinder Network for Subject M1



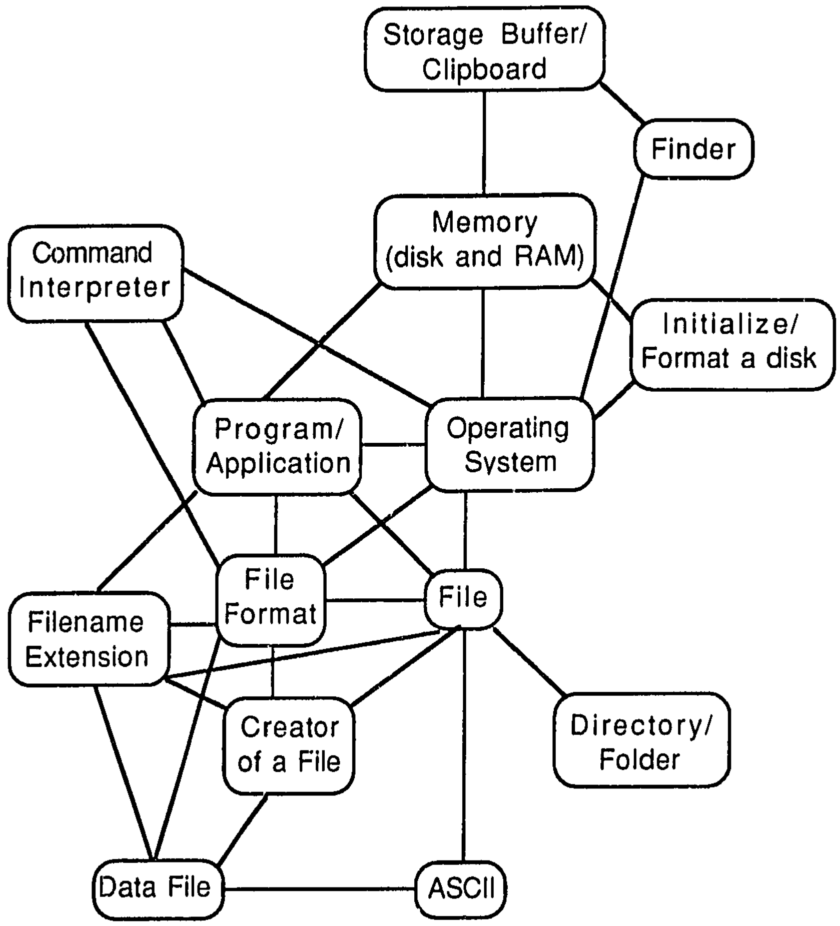
Pathfinder Network for Subject M2



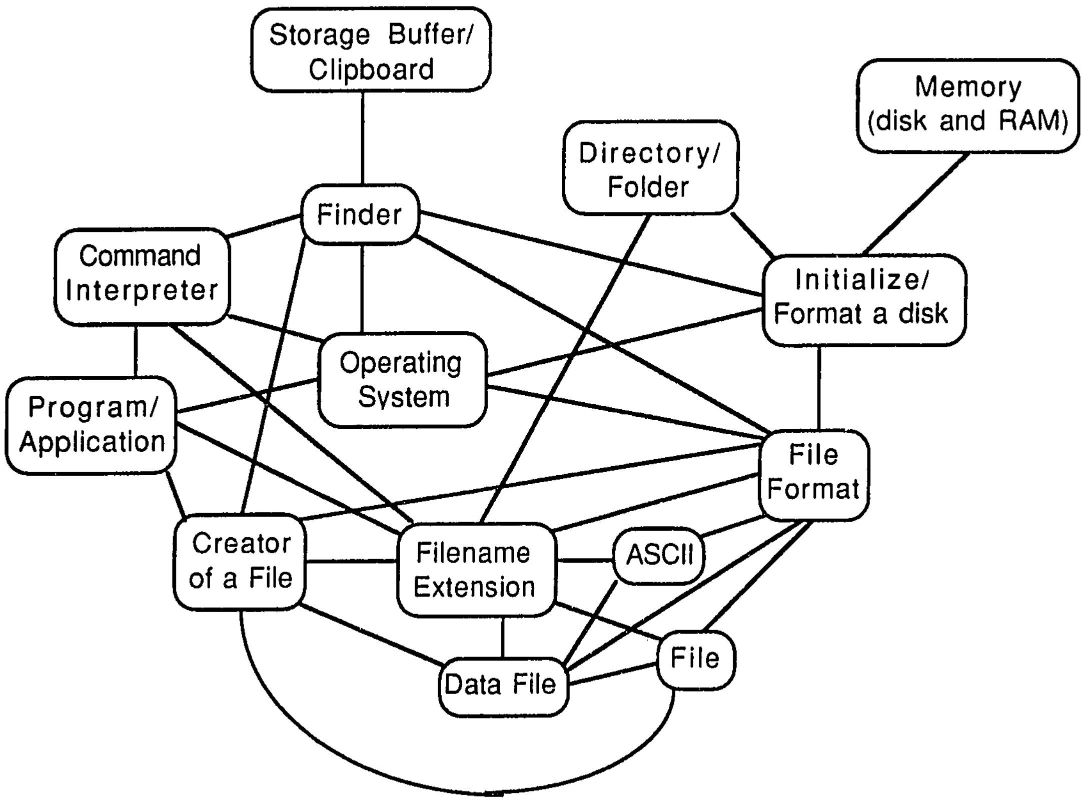
Pathfinder Network for Subject M3



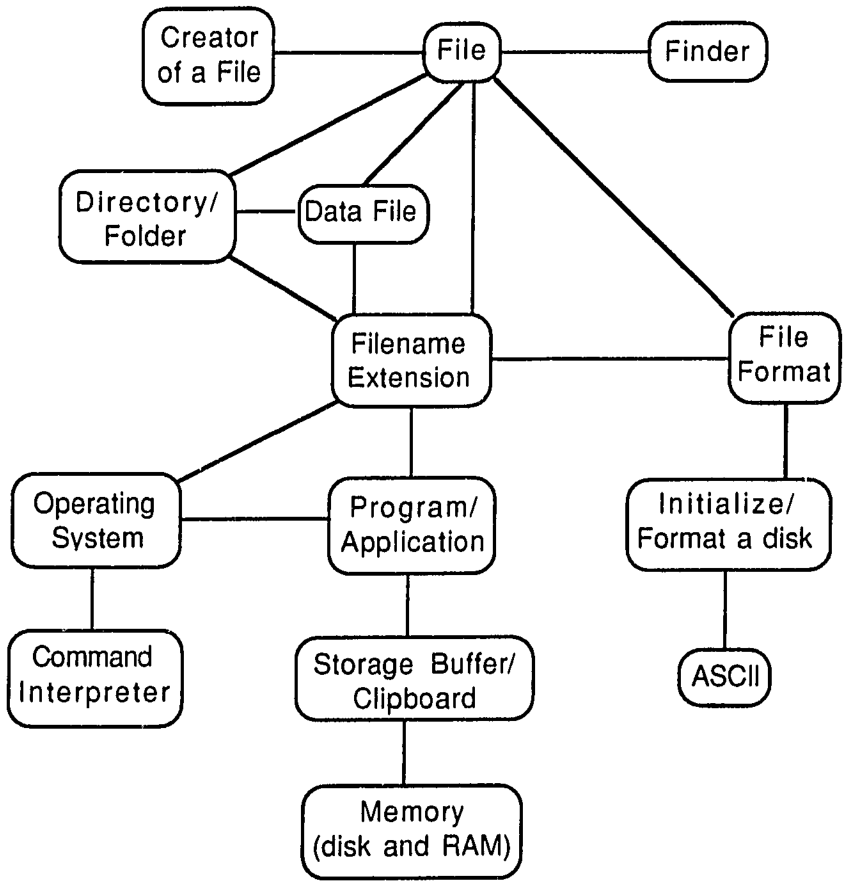
Pathfinder Network for Subject M4



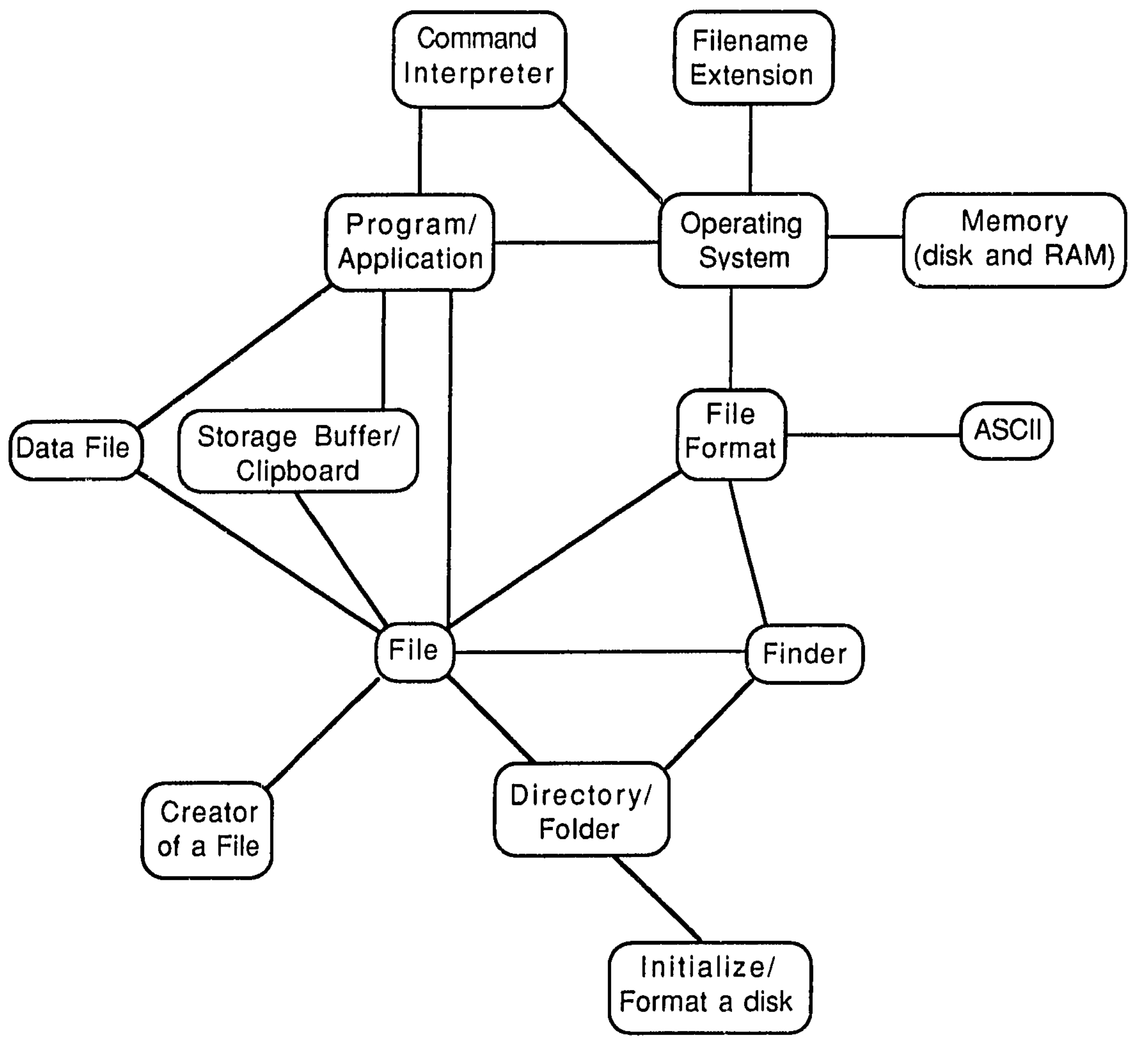
Pathfinder Network for Subject M5



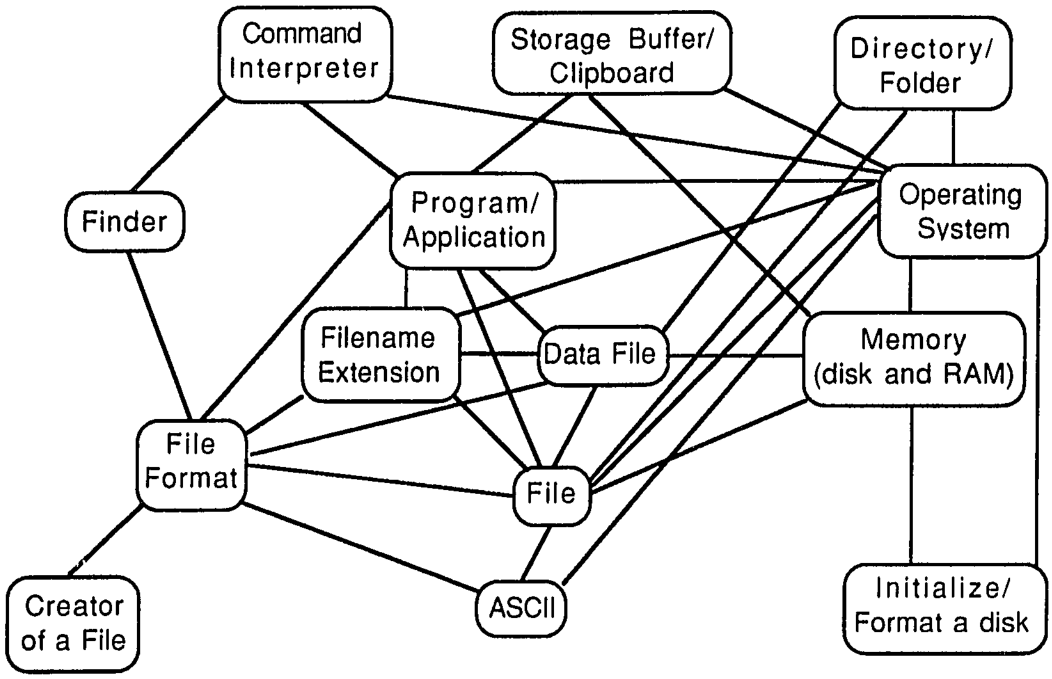
Pathfinder Network for Subject M6



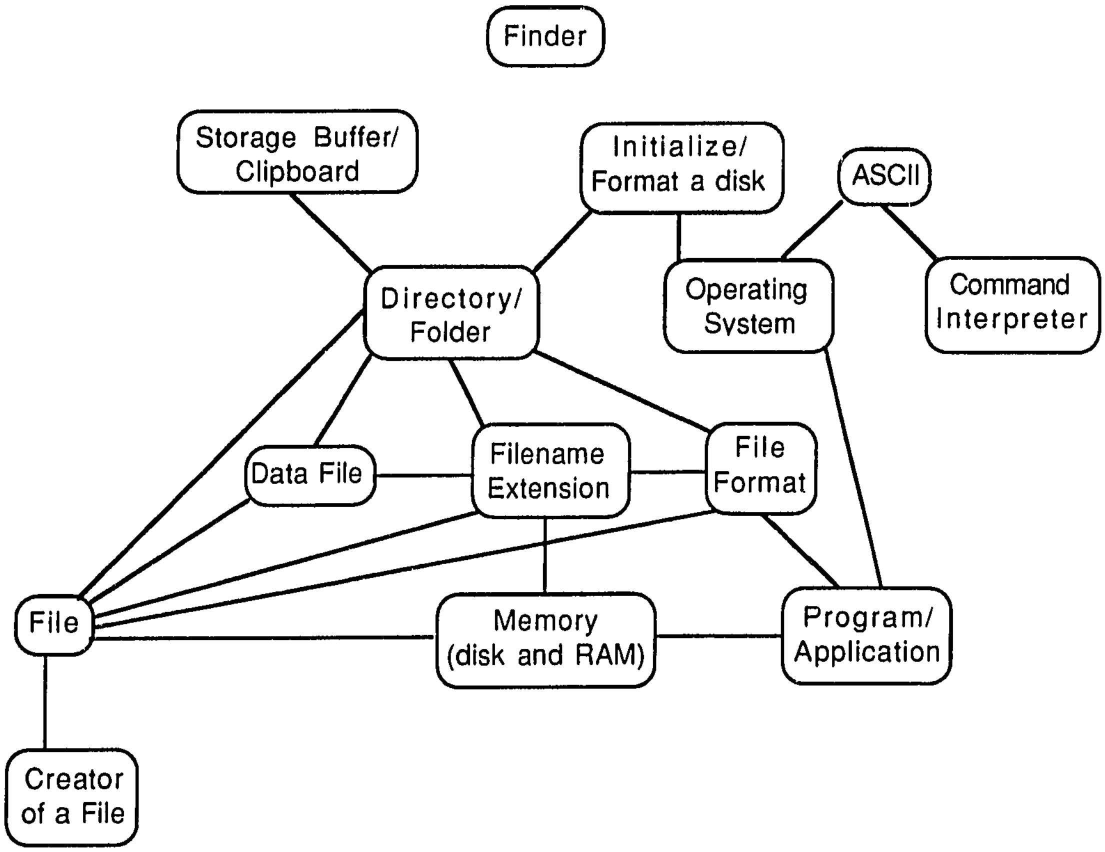
Pathfinder Network for Subject PC1



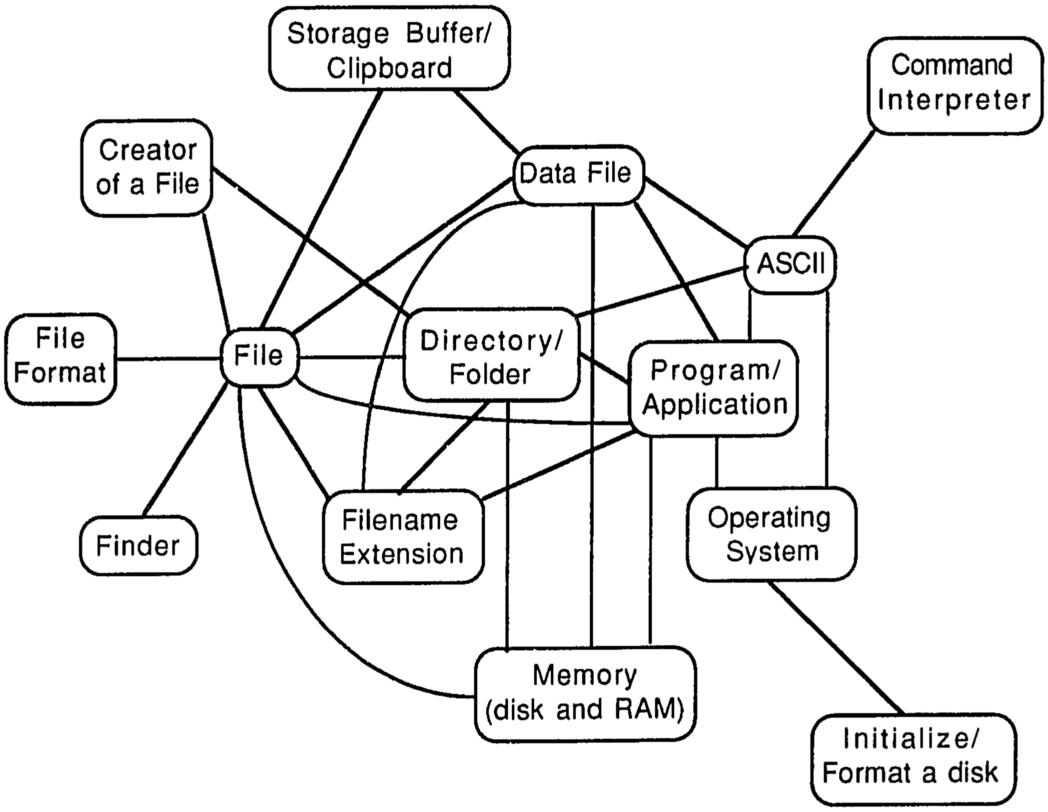
Pathfinder Network for Subject PC2



Pathfinder Network for Subject PC3



Pathfinder Network for Subject PC4



Pathfinder Network for Subject PC5