A FRAMEWORK FOR TESTING THE LEARNING

OF

COGNITION-BASED HUMAN-COMPUTER INTERFACES

By

MICHAEL JOHN MURPHY, B.A., B.Sc.(ENG), M.B.A.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

A FRAMEWORK FOR TESTING HUMAN-COMPUTER INTERFACES

DOCTOR OF PHILOSOPHY
(Management Science/Systems)

McMaster University
Hamilton, Ontario

TITLE: A Framework for Testing the Learning of Cognition-Based Human-Computer Interfaces

AUTHOR: Michael John Murphy B.A. (Queen's University)
B.Sc.(Eng) (Queen's University)
M.B.A. (York University)

SUPERVISOR: Dr. N. P. Archer

NUMBER OF PAGES: ix, 213

ABSTRACT

Successful implementation of Information Systems requires user acceptance. The old approach of adapting users to the system is no longer acceptable as more middle and senior professionals and managers are becoming system users. Due to the increasing people cost component of systems implementation, there has been a recognition that the human-computer interface must be easier to learn to use and recall for the individual who is both a novice and discretionary computer user.

From the cognitive psychology literature, various principles can be applied to the interface design to improve learning and recall. These principles can be used by interface designers to improve the usability of the human-computer interface. Models of human-computer interaction have been devised by other researchers. However, to date there has been little available in the way of satisfactory methodologies or tools to allow designers to measure practically how an interface implementation performs with respect to <u>both</u> learnability and <u>subsequent recall</u>.

This thesis develops a framework for testing human-computer interface learning. The framework differs from previous attempts in that it defines a new criteria for quantifying human-computer learning and recall, as well as providing a simple and effective tool for use by designers to determine such learnability metrics during the design process. In order to demonstrate the usefulness of the framework, it is used to experimentally test an original prototype interface design which attempts to improve human learning speed and memory retention using elaborative learning techniques and the "generation effect". The framework was able to measure significant differences between interfaces with respect to recall performance, and has demonstrated its utility as a contribution to the field of interface usability evaluation.

ACKNOWLEDGEMENTS

I would like to acknowledge the assistance I received in the preparation of this thesis. My thanks to my dissertation committee and in particular my supervisor Dr. Norm Archer for his guidance, teaching and patience. I am also grateful to the Management of Technology and Innovation Institute and Northern Telecom for their support, both financial and moral. In addition, I thank my parents for their support and encouragement. Finally, I thank my two children, Daniel and Katherine, who were born during the duration of this research and who often shared their father with a computer.

DEDICATION

I dedicate this thesis to my wife, Rachelle Laporte. In so many ways, this work has been a joint effort. It would never have been completed without her encouragement and love.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

One of the major barriers to entry and adoption of many office automation systems is that of user acceptance. The problem often centres around non-use by upper or middle management. As an example (Rockart et al 1984), if an organization is first introducing an office system by implementing electronic mail, non-use by upper management will send out the signal that it is acceptable to continue using conventional means of mail. However, if upper management uses the electronic mail system exclusively, it will very quickly become the standard throughout the organization.

It may be difficult to increase usage of computer-based systems by upper or middle management. Often, managers are computer illiterate and fear having to learn new skills in order to use computers (Shneiderman 1979). Also, many managers have a view of computers as 'data processing machines', and they 'have people who operate them' (Eason et al 1981). While these managers may still have secretarial help in order to enter electronic mail and other correspondence, it defeats the purpose of the system if these managers are not able to operate the system in order, at the very least, to read and access their own mail. Also, as the office system grows and other features are made available such as access to databases and decision support systems, the managers must use the systems or else there will be insufficient user support to insure that the system is adopted, has data integrity, is easy to use, etc.(Rockart et al 1984; Slator et al 1986).

I propose that any improvement in the initial learning speed and retention of a new computer system, primarily through improving the transfer of facts at the initial cognitive stage to long term memory, should result in faster learning and adoption of the system by novice and occasional users in general. In a business environment, these improvements should also result in faster learning and adoption by managers and discretionary users, and improved overall organizational implementation.

Designers are becoming aware of various methods to improve the learnability of new interfaces. However, as we will examine, there is not always a clear prescriptive way of doing things, as not all cognitive psychologists rely on the same theories. As well, much of the effectiveness or usability of an interface is based on implementation details difficult to identify or observe, and not simply the interface type or style used (Whiteside et al, 1985; Whiteside et al 1987). Designers are requiring and demanding practical models and tools to allow them to obtain quantitative evaluative data during the design phase in order to identify interface problems and areas requiring improvement (Whiteside & Wixon, 1987). In the specific area of learnability, Polson (1987) notes that "Recently, some authors (e.g., Bennett, 1984) have proposed that project objectives should also include usability criteria specifying training time and productivity. For example a new user should be able to complete a specific task after 2 hours of training, while an experienced user of an earlier version of the system should be able to perform the same task after a half-hour of training. Current human factors practice provides very limited tools for evaluating the reasonableness of such specifications or for effectively managing their achievement" (emphasis mine). Shneiderman (1987) states that "Scientific and engineering progress is often stimulated by improved techniques for precise measurement. Rapid progress in interactive systems design will occur as soon as researchers and practitioners evolve suitable human performance measures and techniques... soon, we will expect software packages to show learning time estimates and user satisfaction indices from appropriate evaluation sources."

Recent work in the area of usability testing and engineering is attempting to redress this problem with new evaluative techniques. This dissertation attempts to contribute to and build on this new approach of usability by examining the possible contribution of measures of recall within a practical framework of user-interface learning and testing, and proposes a systematic and scientific method and tool for evaluating interface prototypes in the area of learnability.

1.1 The Objectives of This Research

Although some of the principles and theories in this proposal are based in the area of cognitive psychology, the main result is hoped to be a direction for improvement in the use of information systems for novice, casual, and discretionary users.

The first major objective is to successfully synthesize from the existing body of work in human-computer interface modelling, human learning, and memory theory, a modified framework that specifically focuses on the issues of human learning and long-term retention of how to use a computer interface. The framework must identify the processes and variables of interest in order to make evaluative decisions useful in improving real life interfaces. In summary, it must have a practical and useful result in the real world of designing and modifying human-computer interfaces.

The second major objective is to apply the framework and demonstrate its usefulness. In order to do this, a new interface design utilizing multiple alternative techniques is implemented, based on principles relevant to improving human learning and memory. Actual working prototypes of the interfaces are experimentally tested on human subjects, in order to evaluate the effectiveness of the tool for comparing interface designs.

In order to operationalize the testing of the framework, it was required to implement a tool that will allow a designer to test the performance of an interface design in the early stages of development, based on the variables defined by the framework. The tool facilitates a practical, real world extension of the test framework and moves the thesis from the realm of cognitive science to the more applied domain of business information systems research and application, and specifically interface design.

The experiment performed tested the usefulness of utilizing elaborative learning strategies in learning to use a computer interface. Its principal contribution to the thesis is to refine the more general framework by providing a real-world interface prototype for testing purposes. This will demonstrate the potential utility of using the proposed framework in an actual interface design and testing exercise.

1.2 Importance of Topic

> "There is a basic tension in human-computer interface design. When users have difficulty with a system, there are generally speaking, two opposing solutions to the problem:
>
> 1. Adapt the user to the system, or
>
> 2. Adapt the system to the user."
> (from Good, Whiteside, Wixon and Jones 1983)

There was a definite trend in the late 1980's away from solution 1 to solution 2 (Thomas & Kellogg 1989). One of the basic motivators is the shifting cost curve of a total information system project. In the 1960's and 1970's, the major element of the cost curve was made up of hardware and software costs. This shifted dramatically in the 1980's with the distribution of hardware through individual workstations. Today the majority of the total cost is represented by software (including system maintenance costs) as well as the organizational costs ("orgware" Bjorn-Anderson 1988) which include determining information requirements, training, end user salaries and opportunity costs, restructuring work procedures and other personnel implementation costs (Bjorn-Andersen 1988). Other sources estimate that North American businesses are spending up to 40% of their capital investment dollars on computers, production automation, office systems and related systems implementation, double their 1978 level (Gist, et al, 1989). Faced with this cost structure, it makes economic sense to follow solution 2 and adapt the system to the user. As well, as information systems are no longer relegated to use by clerical staff but are used by the middle and upper managers who are also controlling and funding the projects, forcing users to adapt is no longer a viable option (Figure 1). In addition, these new classes of users are discretionary users. If the system cannot or will not adapt to these users' needs, it will fail from disuse (Rockart, et al, 1984)).

It is somewhat ironic that cost structure is the motivation for change in approach. Human factors and socio-technical systems have long been discussed as a proper direction on their own merits (Hirscheim & Newman, 1988). While these principles have achieved some limited application in Europe and in particular in Scandinavia, they have not been quick to gain a foothold in North America save for the use (overuse?) of the term "user-friendly" to describe many of the systems sold since 1974. However, the hard cost of system failure and the discretionary use of more senior organizational members have

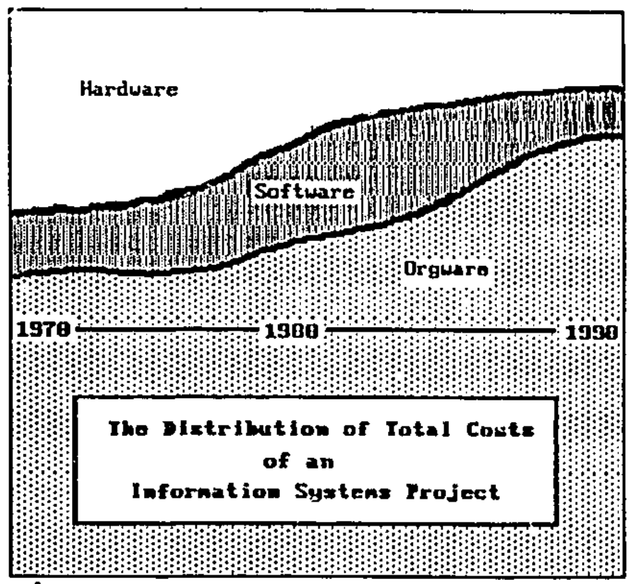


Figure 1
Cost Structure of Information Systems
(from Bjorn-Anderson, 1988)

made the principles based on solution 2 more widely accepted (Bjorn-Andersen 1988, Hirscheim & Newman 1988).

There has been much research in the area of modelling human-computer interaction. As designers attempt to create more appropriate user interfaces, new techniques and methodologies emerge. One such concept is that of interface "usability", which has shown much promise in providing "measurably" better interface designs. However, there is still much work to be done in providing designers with practical tools that allow measurement of the various usability criteria. Support for the requirement for new tools to measure learnability has been mentioned in the introduction above (Polson, 1987; Bennett, 1984; Whiteside & Wixon, 1987). In addition, Reisner (1987) states that "Most of the models, if not all, have not been tested as engineering tools." and continues "models that predict user performance or learning time usually or always assume error-free performance." Practitioners find that these limitations hinder using the models in real design and evaluation projects.

This proposal will attempt to contribute to this area by developing a framework and tool to measure the learnability of a computer interface which can be used in a practical way by designers during the prototyping phases of an interface design.

1.3 Outline of Dissertation

The dissertation will first examine the stages of learning that typical managers and professionals go through when learning to use a computer system interface. It then summarizes the relevant literature in cognitive psychology relating to learning and memory theories and models. In this context, currently used interface techniques are discussed relative to their applicability to learning and retention. The paper then reviews other studies of novice and discretionary computer learning, applying cognitive principles.

The dissertation then reviews existing models of human-computer interaction, the application of individual differences to testing interface learning, and selecting an appropriate cognitive learning style. Two specific learning models and taxonomies are discussed in detail.

A framework for measuring computer interface learning is then proposed, based on the theories and models discussed. The framework objectives, domains and variables are explained in detail with reference to its use in an interface design environment. Based on this model, a tool for implementing the framework is described.

In order to test the proposed framework for utility and the tool for practical usability, a human subject experimental study is then described and evaluated. The study tested five different interface implementations, based on cognitive theories of improving learning and retention, using the new framework and tool.

The dissertation ends with a summary of findings and conclusions regarding the applicability of the proposed framework.

CHAPTER 2

Literature Review of Applicable Theories and Designs

The following chapter reviews the literature in the areas of computer learning, cognitive psychology, interface design and previous studies and measurement techniques in the area of human-computer learning.

2.1 Computer Learning by Managers and Professionals

It has been recognized that even if managers overcome any existing biases against using a computer system, a problem of training remains. Managers often have little time available for learning to use a computer terminal. An ideal situation of a large block of uninterrupted time is usually not available in order to learn the system. Based on studies of managers (Mintzberg, 1973), the learning takes place on an 'hour here, hour there' basis, with perhaps weeks elapsing between sessions (see also Umanath & Scamell 1988, Slator et al 1986).

This type of learning is less effective than a concentrated period followed by extensive practice. In skill acquisition, it has been found (Anderson, 1985 p. 234; see

Figure 2) that three stages exist. The first is the <u>cognitive stage</u>, where people commit to memory a set of facts relevant

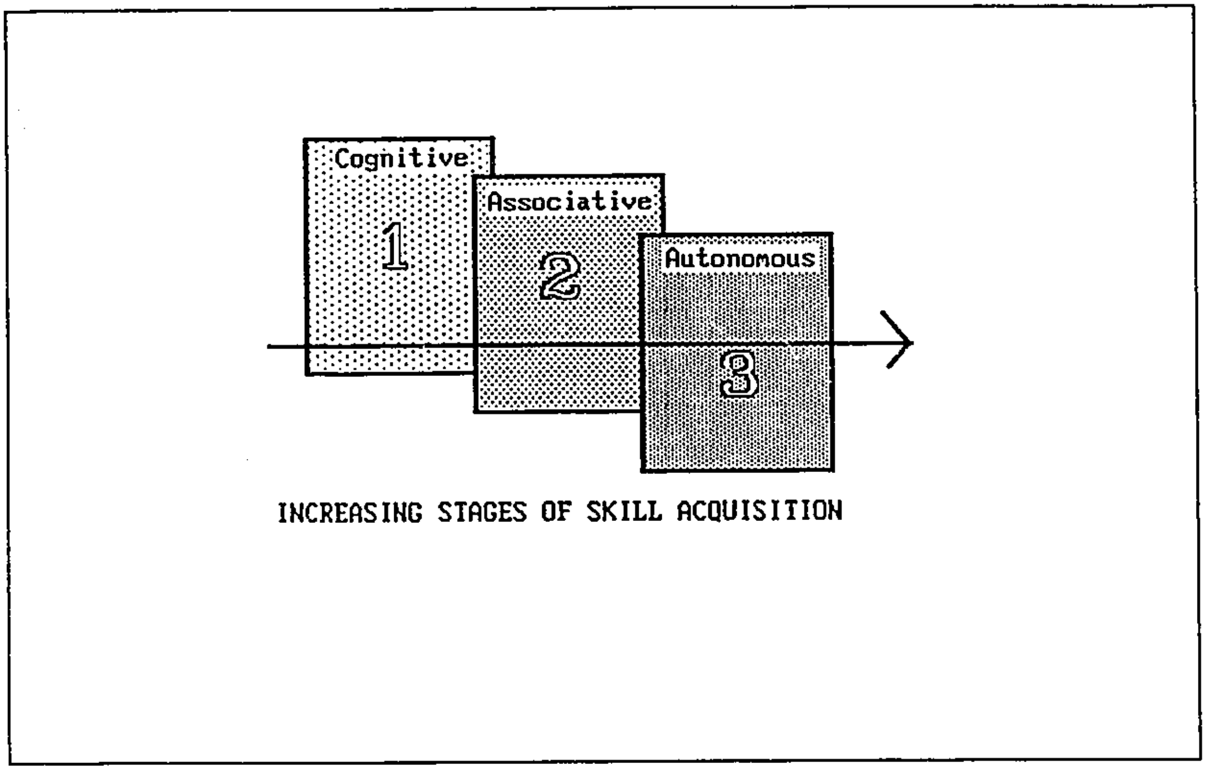


Figure 2
<u>3 Stages of Skill Acquisition</u>
(from Anderson, 1985 p. 234)

to the skill. Once complete, they move to an <u>associative phase</u>, where initial errors in their understanding are corrected and the association between different facts and actions for successful performance is strengthened. A third, <u>autonomous stage</u> may be thought of as an extension of the second phase. The skill becomes more and more autonomous as experience is gained, which has been shown (see Figure 3) to be represented by the Power Law of Practice.

Let us use as an example a senior manager who is initially not familiar with using a computer system. The organization has implemented a new electronic mail system, and all senior managers have been provided with desk terminals to allow them to compose, send, read, and file their mail. The manager then attempts to

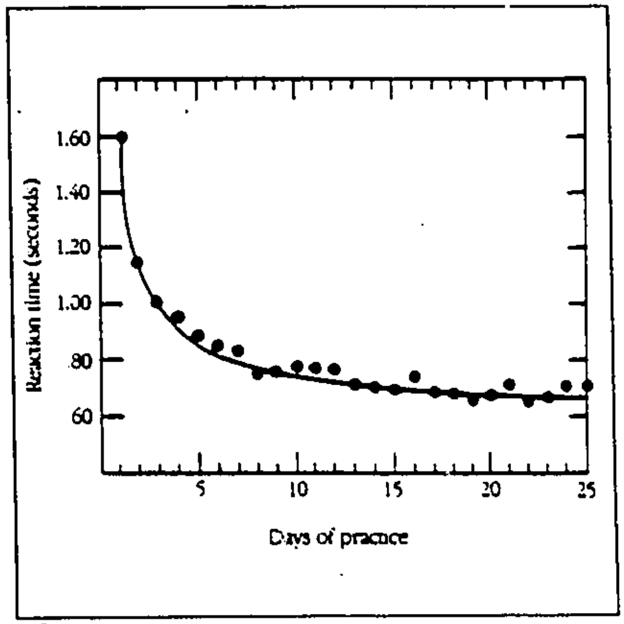


Figure 3
Power Law of Practice
(from Anderson, 1985 p. 238)

"learn" to use the system. In the first (Cognitive) stage, the manager typically learns a set of relevant facts, such as how to turn on the device, how to log on, how to access the mail system, what the command is for reading, filing etc. Once this stage is mastered to a degree, the manager then moves to the associative stage. Here, errors in the initial cognitive stage are gradually detected and corrected. As well, the connections among the various elements required for success are reinforced. For example, the manager may begin to realize that the system requires that an item of mail must be read before it can be filed, and that a file must be labelled before it can be used, etc. The third stage of skill acquisition, the autonomous stage, is reached when the manager achieves a level of skill allowing competent use of the facility. No sharp distinction is defined between the associative and autonomous stages, as the manager may reach a proficient skill level for certain aspects, such as reading mail, while other aspects are still at the associative stage, (e.g. editing mail and forwarding). Managers attempting to learn to use a computer system must move through the first two stages of learning before they can attempt to utilize the new skill in a problem solving application. The reality of the situation is that in a typical senior managerial environment as described by Mintzberg (1973), the manager only begins to learn the facts (stage 1), and then aborts the learning session before an effective transfer to long-term memory has taken effect through rehearsal. In the next learning session, which may be some days later, many of the facts must be re-committed to short term memory (stage 1) before any attempt can be made to progress to learning stages 2 and 3. This may result in frustration and inefficiency, and may reinforce any existing fears (Shneiderman 1979, Eason & Damodaran 1981).

The relevance of this section to this research can be summarized as follows:

1) Skill proficiency can be categorized as cognitive, associative and autonomous, which are useful design and evaluative benchmarks.

2) The amount or number of practice iterations increases skill proficiency.

3) As a computer system design objective when dealing with discretionary and novice users, it is desirable to quickly achieve the highest level of skill acquisition possible.

The concept of three stages of skill acquisition will be utilized in the framework developed in chapter 4.

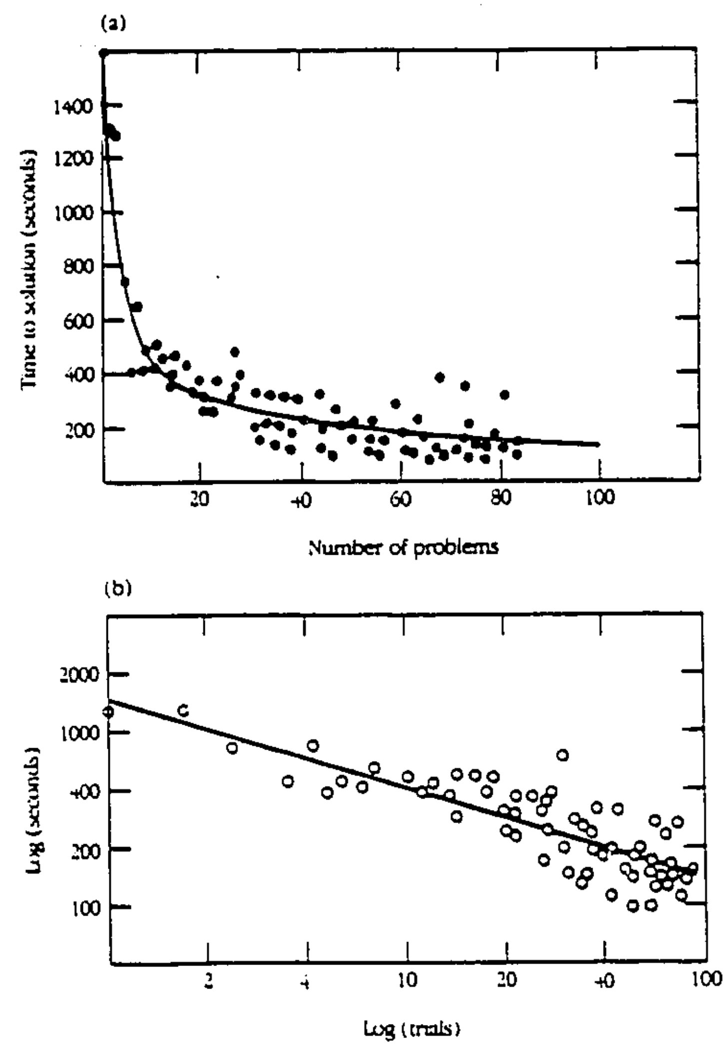
2.2 Cognitive Psychology and Interface Design

Current research in the cognitive psychology field provides clues to improving the learning speed and retention of that learning when a subject first begins to use computers, or a new computer system for the first time. We are starting with the premise that the typical user has a fixed amount of time to learn to use the new system, and that any technique to increase the amount of or quality of the initial learning will be of value. We are therefore looking for means of increasing the amount of Original Learning that takes place, and also improving the effectiveness of the transfer to Long Term Memory. This should reduce the amount of time spent in the first stage of skill acquisition, (the cognitive stage) as discussed earlier (Anderson, 1985). We should point out that there is currently some debate among cognitive psychologists concerning learning and remembering. Remembering is an awareness (recall or recognition) of a prior occurrence, whereas learning is a measured improvement in the performance of a task. For example, remembering that an icon has been seen before is different from being able to use the icon to perform a task, which would represent learning. Some researchers feel that there may be separate memory systems used for different tasks, but others think the same system is used (Gabel 1987). Recalling that the emphasis of this research is to apply theory, we will concentrate on using results that have shown improvements in either type of task, and avoid relying on any particular memory models, unless there is evidence that some models differ in the result of improving either remembering or learning.

2.2.1 Short and Long Term Memory

First, let us quickly review some theories of Short Term Memory and the means of transfer to Long Term Memory. Short Term Memory (STM) is an active, or "working memory" that can keep a limited amount of information on hand while performing a task. This information can be retained for recall indefinitely if the information is kept active by rehearsal. However, the accuracy of recall diminishes within a matter of seconds if no rehearsal takes place (Murdock 1961, Figure 4). Miller's work (1956) on chunking and the magic number 7 is well known, and many computer interface designs have attempted to use this result in limiting the amount of cognitive load imposed on the user. What is more germane to this thesis is the mechanism for transferring information to the user's long term memory.

Long Term Memory (LTM) is the more permanent memory individuals have for storing and retrieving information for indefinite periods of time without using continuous rehearsal. We first examine LTM within the context of a propositional network model of storing information. This model, proposed by



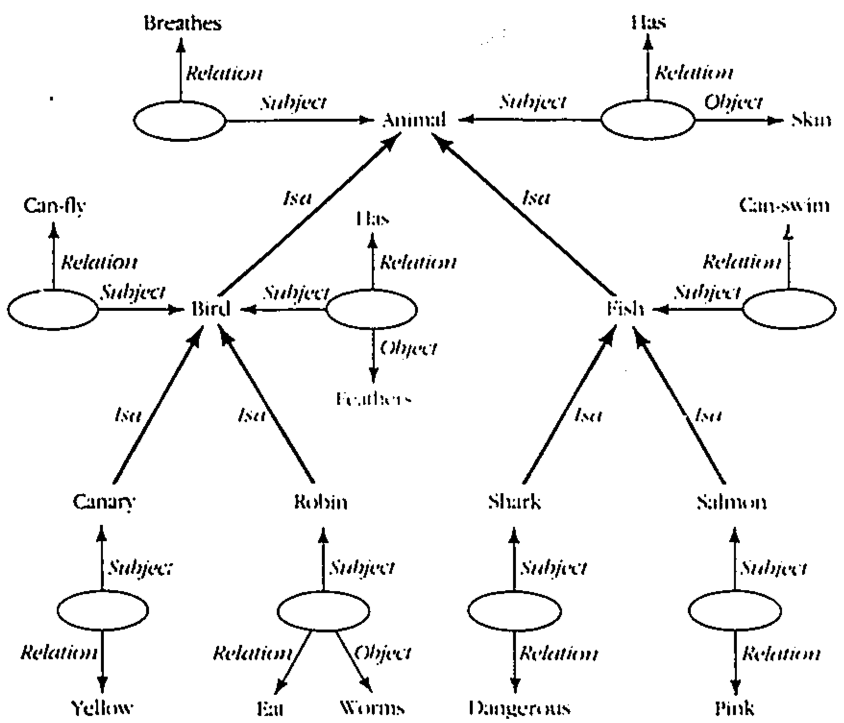
Time to generate proofs in a geometrylike proof system as a function of the number of proofs already done. (a) Function on a normal scale, $RT = 1410P^{-.55}$; (b) function on a log–log scale.

Figure 4
Effect of Rehearsal
(from Murdock, 1961 reproduced in Anderson, 1985)

researchers such as Anderson, (1981) and Collins and Quillian (1969), organizes all information stored in the mind in large hierarchical networks of propositions. These can be thought of as containing nodes which represent ideas, and links which are associations between these ideas. To illustrate this model, let us look at the associations used by Collins and Quillian (1969). Subjects were asked to judge the validity of statements such as "Robins have feathers" or "Robins have skin". Figure 5 represents the assumed hierarchical network representation. Their experiment measured the time required to come to a conclusion on various statements, and showed that the more links in the network required to determine validity, the longer the time required to complete the task, and that these links (or associations) appear to be additive. This model also has an extension called a schema which represents how propositions group together to define objects and episodes.

2.2.2 Propositional Network Model of Memory

The network theorists' view is that information stored in LTM can be retrieved by activating the nodes in the network, and spreading this activation (Anderson, 1985) through the network until the appropriate and desired node in the network is reached. In order to improve recognition and recall ability, this model proposes that increasing the amount of elaboration (providing additional redundant information at



The network of concepts assumed by Collins and Quillian (1969) in their experiment to compare reaction times in making true-or-false judgments about statements. A hierarchy of concepts and associated properties can be seen in the figure.

Figure 5
<u>Network Model of Memory</u>
(from Anderson, 1985)

the time of processing) at the time of learning will increase the number of possible retrieval paths available.

Let us look more closely at possible ways of providing this elaboration as it could apply to a user learning to use a computer interface. The following survey is based on Anderson (1985). Craik & Lockhart (1972) proposed that better LTM resulted if the material was processed "deeply" (depth of processing), which could roughly relate to increasing the number of elaborations. However, later studies indicate that it is not so much the number of elaborations, but more the appropriateness of the elaboration that improves LTM. One way to increase the depth of processing is to have the subject generate the elaboration. Slamecka and Graf (1978) found improved memory when subjects had to generate rhymes (generation effect), rather than simply read them. Nelson(1979) has shown that phonemic processing improves the memory trace. Kolers (1979) found that a subject's memory was better if sentences were read upside down. The point here is that if the amount of cognitive processing can be increased at the time of learning, the memory trace can be improved.

Additional evidence of improved recall has been demonstrated by Graf (1980, 1981) and Donaldson & Bass (1980) using other generation techniques. It would appear that a significant gain in recall can be achieved if subjects generate words or simple sentences that are meaningful to them, rather than some prescriptive or nonsense word or label. This would indicate that a useful feature of a computer interface would be the capability for the users to generate their own names or labels for the functions being utilized. These labels could be meaningful words or simple sentences that the user determines after understanding the function of a command. The system could then either provide these labels in verbal form back to the user at the time of command icon activation, perhaps as part of a help facility, or provide a text label superimposed on the icon on the screen, or perform some combination.

2.2.3 Recall, Recognition and Context

In section 2.3, we will discuss the desirability of using a recognition cue such as a menu rather than having subjects recall a command. As a general rule, studies have shown that recognition performance is better than recall (Anderson 1985, Wolford, 1971). Propositionalists contend that a recognition test is easier because it offers more ways to search memory. There can be conditions, however, where the opposite is true. Let us digress briefly and examine the effects that are a result of the context in which information is learned. Smith, Glenberg & Bjork (1978) have shown that re-introducing the physical context present at the time of learning,(for example the same room where the information was learned) improves recall. Bower, Monteiro & Gillian (1978) have also shown the same effects if the identical emotional context is recreated, in their case subjects being happy or sad. Similar effects have been shown if subjects are in the same physical state, e.g. drunk or sober. The propositionalist explanation is that again, recreating the context of the learning environment provides additional association paths that can activate the information to be retrieved. A further extension of this is encoding specificity (Tulving & Thompson 1973), which has demonstrated that memory for learned material can be affected by the context of the other learned material in which it is embedded (Anderson, 1985). It has been shown that if a test is sufficiently weighted in favour of recall, in these circumstances recall may exhibit better performance than recognition. Applying this knowledge to the design of a computer interface would result in the following rules of thumb for comparing two or more designs:

1) Use recognition of commands rather than forcing recall.

2) Provide a consistent visual screen context both when in the training mode, and in the operating mode (Some systems use a different screen and layout when training or demonstrating).

3) Subjects should be in the same physical environment as during the initial learning session.

4) Test at the same time of day to eliminate any physical state context effects.

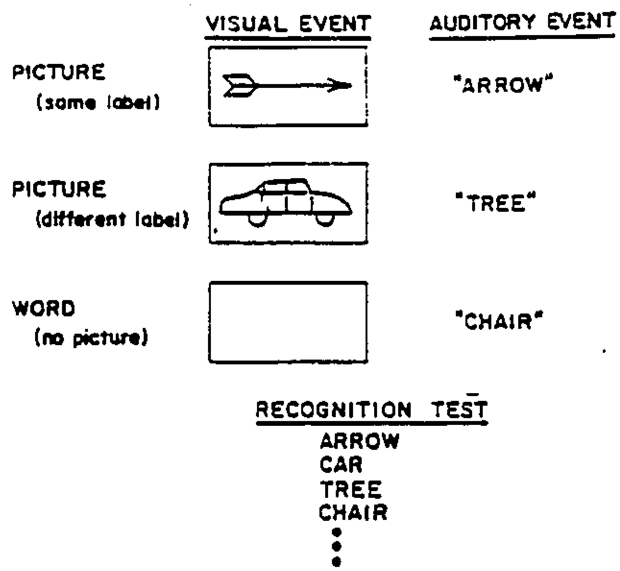
2.2.4 Dual-Code Theory

We now examine results from some cognitive psychologists who favour a different model of human memory and learning.

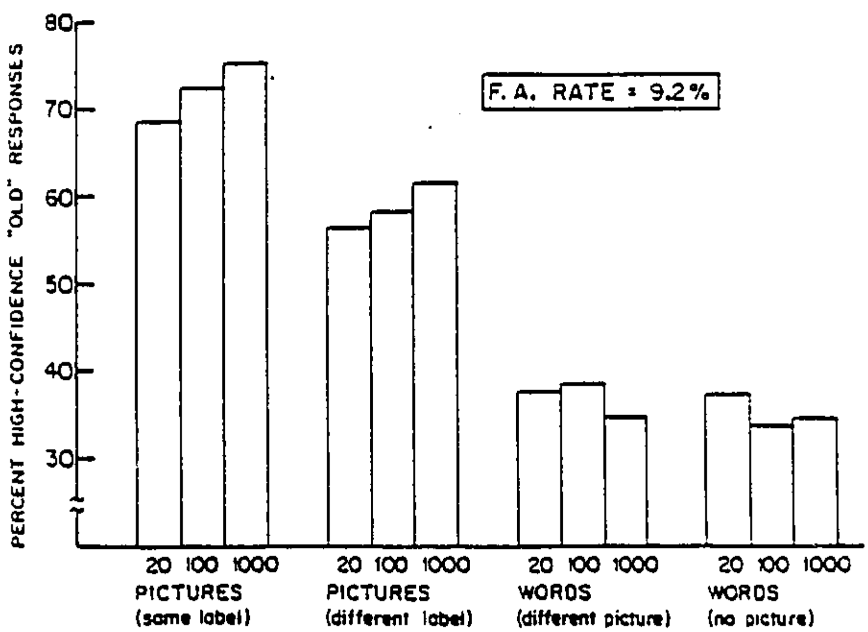
Memory capacity for visual information appears to be much greater than for verbal information (Shepard 1967). Evidence is that people do not remember the exact visual details of the picture, but rather some representation of the picture's meaning (Bower, Karlin and Dueck, 1975). This is an underlying theory of icon-based computer interfaces used on devices such as the Xerox Star workstation and the Apple Macintosh (Smith et al 1981). Users appear to find it much easier to move a direct manipulation device such as a mouse around a screen and to activate icons (pictures) such as a filing cabinet (Wilton & McLean 1984). Icon representations have meanings that seem to be easier for the user to recall than equivalent commands in a command-language based system. In fact, an icon-based system utilizes memory recognition of the picture (Oh, I know what that means), whereas a command language utilizes memory recall (Now, what was that file command again?). Icon-based systems have been shown to be generally easier and faster to learn to use than other systems (Smith et al 1981).

Paivio (1971) developed a dual-code theory to account for imagery effects in verbal learning studies. "Dual code theory... is based on the assumption that memory and cognition are served by two separate symbolic systems, one specialized for dealing with verbal information and the other for non-verbal memory". The two systems are presumed to be interconnected but capable of functioning independently. Interconnectedness means that representations in one system can activate those in the other, so that for example, pictures can be named and images can relate to words (eg. Mona Lisa). Independence implies, among other things, that nonverbal (imaginal) and verbal memory codes, aroused directly by pictures and words or indirectly by imagery and verbal encoding tasks, should have additive effects on recall (Paivio and Lambert 1981). Using this theory, Madigan (1983) found that recognition of a visual event augmented with a synonymous verbal label was better than if a conflicting verbal label was used, and was much better than if words only were used (see Figure 6).

Bower, Karlin and Dueck (1975) had subjects study pictures called droodles with and without an explanation of their meaning. Subjects who were given verbal (text) labels showed better recall of these pictures (70%) than those who were not given labels (51%). These droodles are similar to simple icons used in typical systems (see Figure 7), although



Examples of visual and auditory events



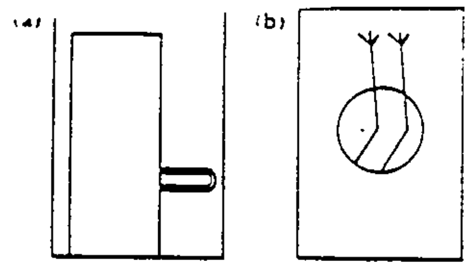
Recognition performance as a function of imput modality, labeling, and exposure items.

Figure 6
<u>Pictures versus Labels</u>
(from Madigan, 1983)

it might be argued that icons are more representative and easily identifiable. This may be an important point. If the icon brings with it a meaning to a user that is inconsistent with the actual activity (Bott, 1979) some negative interference may result. (This is discussed further in section 2.3.2 below). Perhaps icons should be more symbolically transparent and similar to droodles. It would appear that an interface that augments icons with self generated verbal information using verbal word labels would be consistent with the findings discussed above including improved meaning recall, verbal/non--verbal additive recall, increased (elaborated) processing, the "generation effect", and generation of meaningful elaborations.

2.2.5 Caveat & Relevance

It must be noted that there is far from an agreed general theory on human learning and memory in cognitive psychology. Hintzman (1989) reviewed the competing theories and models of learning and memory. Of particular note are his observations on the various theories available to prove or disprove the generation effect discussed earlier. He concludes that there may be not one but many "generation effects" (see also Greenwald et al 1989, also McDaniel, et al 1988). As stated earlier, this research proposal can only utilize the "effects" resulting from cognitive psychology which appear to be reasonably consistent. An understanding of the various psychological models and theories that may or may not be responsible for these effects is useful, but this research does not pretend to make definitive conclusions about which of the currently competing models is more correct.



(a) A midget playing a trombone in a telephone booth.
(b) An early bird who caught a very strong worm.

Droodles used by Bower et al. (1975).

Figure 7
Droodles
(From Bower et. al. 1975)

We can summarize the relevance of this section to the research as follows:

1) Elaborative processing facilitates more effective transfer to Long Term Memory (LTM).

2) Self generation appears to improve retrieval.

3) The use of both images and verbal information can lead to improved recognition and retrieval.

These results will be used in the development of the interface prototype described in chapter 6 and form the basis for the proposition P1 (later developed in section 6.5) that human recall performance will be better for interface designs which incorporate user self-generation.

2.3 Current Trends in Human-Computer Interface Design

Interfacing to computers has progressed significantly from the early days when the only means of communicating with the machine was using banks of toggle switches and push buttons on the computer control console. Instructions and input were often fed directly in binary, hexadecimal or octal. This was acceptable as long as the user of the computer was a specialist.

Today, the objective of a computer interface is to provide non-computer experts with easy access to computing power to solve problems and perform tasks. The following examines the current techniques in use.

2.3.1 Interface Classification

We will look at the various techniques available from the perspective of suitability for use by casual users with little or no previous computer use, as this is the primary population that this study is attempting to address. A good classification of these techniques is provided by Baecker & Buxton (1987). They list nine major categories of interactive style (see Table 1). Most modern interfaces use some combination of more than one member of this set.

TABLE 1
INTERFACE CLASSIFICATION
From Baecker & Buxton (1987)

NAME	DESCRIPTION
COMMAND LINE	the user types instructions to the computer in a formally defined command language
PROGRAMMING LANGUAGE	the command language allows its own extensions through the definition of procedures
NATURAL LANGUAGE	the user's command language is a sufficient, well-defined subset of some natural language such as English
MENU SYSTEMS	the user issues commands by selecting in sequential choices from among menus of displayed alternatives
FORM FILLING	the user issues commands by filling in fields in one or more forms displayed on the screen
ICONIC	user commands and system feedback are expressed in pictograms instead of words
WINDOWS	the user's screen is divided into a number of possibly overlapping rectangular areas, each of which handles a specific function or is itself a "virtual terminal"
DIRECT MANIPULATION	the user manipulates, through a language of button pushes and movements of a pointing device such as a mouse, a graphic representation of the underlying data
GRAPHICAL	the user is defining and modifying sketches, diagrams, renderings, and other two- or three-dimensional images and pictures.

2.3.2 Applicability to Novice/Discretionary Users

We now briefly review each interface type with regard to its suitability for use with novice or discretionary computer users.

The most promising technology for use in the long term, for simplicity of learning, is probably a natural language interface that uses a language such as English. Combined with a speech input capability that would be robust enough to correctly interpret most human speech, it is difficult to imagine a more congenial human-computer interface. However, it would appear that we are still a long way from being able to provide effective full natural language interfaces with computers (Hayes & Reddy 1986; Baecker & Buxton 1987 Chapter 10), although some progress has been made with limited, restricted language interfaces for specialized functional uses.

Command Line dialogues have been the traditional method for computer communication, and typically elicit fear, apprehension and non-use from the novice and praise from the expert user. These systems are inherently artificial languages with their own special grammar. Much of the research into this area has focused on naming conventions for the commands. There is very little consistency between various command interpreters in naming common commands. Instead, designers use what they feel is a "natural" language representation.

Bott (1979) found evidence (Black 1981) that real world preconceptions also provided a kind of proactive interference effect when non-computer-literate subjects attempted to learn a computerized text editor. As an example, users were attempting to use the 'PRINT' command of the UNIX operating system which is an example of a command line dialogue. The help procedure started off with the statement "How to print text:". Since 'print' commonly refers to printing presses and 'text' commonly refers to a book, many computer naive subjects interpreted this statement to mean 'how to print a book on a printing press'. In a separate example, users found it surprising that a copy of a computer file was retained in the storage media even though they were working on an active copy in their work space. In the office environment, a file is physically removed from the filing cabinet (storage media) while it is being used at the desk (active workspace). Bott determined that the users' own definition of commands were useful in facilitating learning by non-computer-literate individuals. Therefore, although a command line dialogue is generally felt to be a poor interface for novice users, Bott's determination of the value of self-generation of commands helps to support its use as a technique to augment other, more appropriate, dialogues.

Programming Language dialogues are really extensions of the command line dialogue, with the additional capability of defining new procedures and commands. Again, this technique is more appropriate for the expert user community for the same reasons discussed above.

Menu-driven interfaces have proven to be popular with novices or people who are infrequent users because the systems use recognition-based memory cues instead of requiring memory recall. These systems can be awkward for the expert user if there is no facility to jump clear of the menu nesting that many systems utilize. However, an adaptable combination of menu and command line is often used in modern, good quality software to provide utility for both the novice and expert (Shneiderman 1986, Perlman 1985, Baecker & Buxton 1987). Much research has been done on the design of menus, and Shneiderman (1986) surveyed the evolving design parameters that appear to provide the most effective menus. Perlman (1985) has found that menus are most useful if users, even experts, do not know what options are available to them, or if available choices change over time. As well, Card et al (1983) and Shneiderman (1986) have found that as a general rule, users prefer broad, shallow menu trees instead of deep, narrow structures.

Form Filling interfaces are similar to menu systems in that, according to Perlman (1985), menus are displays of alternatives, while forms are displays of requirements. Form systems are particularly useful for capturing input parameters which do not have a small set of alternatives as part of another overall system, such as a database or report generator.

Iconic Interfaces were introduced extensively to the marketplace in the last decade first in the Xerox Star workstation, and later by Apple Computer in their Lisa and Macintosh computers. The success of the Macintosh, the amount of imitation (eg. Microsoft Windows and OS/2 for the IBM family), and the general popular press acknowledgement of its ease of use appear to indicate that an iconic interface offers much promise as a means of increasing user learning speed and skill acquisition. However, there is little hard experimental evidence documenting the advantages of icons in the human-computer interface. Most literature in the computer science and systems field are prescriptive and list intuitive advantages, but borrow heavily from cognitive psychology models of human learning and image processing capability (Baecker & Buxton 1987, Hemenway 1982).

The use of windows allows the visual work area of the user to be divided into multiple activities. Windows can be useful to the novice in acting as a help facility available on screen while the user is engaged in some activity. Research in this area concentrates on differences between overlapping and tiled windows under various conditions (Bly & Rosenberg 1986). Windows are also particularly useful in providing users with an external memory extension to their own short term capability that is useful when attempting to deal with multiple concepts at one time (Card, Pavel & Farrell 1984).

Direct Manipulation has achieved widespread market success with spreadsheet products and computers such as the Macintosh. With direct manipulation, the user manipulates, through a language of button pushes and movements of a pointing device such as a mouse, a graphic representation of the underlying data. Shneiderman (1983) claims direct manipulation to be particularly effective for novice users as it includes the following applicable attributes:

-Novices can learn basic functionality more quickly
-Intermittent users can retain operational concepts better
-Error messages are rarely needed
-Users can immediately see if their actions are furthering their goals, and if not, can modify their actions
-Users experience less anxiety, actions are easy to reverse
-Users gain confidence and mastery as they initiate actions, feel in control, and can readily predict system responses

These purported advantages over other interfaces have not been proven scientifically (Norman, et al 1986). However, direct manipulation appears to be appropriate for the novice user based largely on the sales and popularity of products that utilize it. Also, direct manipulation concepts are consistent with the principles of cognitive psychology discussed in the previous chapter.

Graphical Interaction systems are really a specialized subset of a direct manipulation system dealing specifically with diagrams and graphs, and would include visual art and computer aided design (CAD) systems.

2.3.3 Interface Selection for Novice & Discretionary Users

At this point, it would appear that for the purposes of this study which is focusing on novice and discretionary computer users, the most appropriate interface technology available would be Direct Manipulation utilizing Iconic, Menu and Window elements. We recognize that there is no body of quantitative research that proves the superiority of direct manipulation. The choice is based largely on the observational and theoretical studies discussed, as well as the practical issue that they have achieved widespread market penetration. Command and Programming language dialogues should be avoided for this type of user, and Form based systems do not appear applicable for the initial types of tasks we hope to teach these users. Finally, a Natural Language interface would be desirable, but we must wait for significant advances in artificial intelligence and parallel processing technologies before such a system is economically viable.

2.4 Studies of Novice or Discretionary Computer Users & Design

Although we have not come across studies directly addressing the exact topic of this thesis, there have been many other attempts and theories on improving interfaces designed for the novice user.

2.4.1 Human Factors

Paxton & Turner (1984) provide a review of the literature to 1984 on applications of human factors to the needs of novice computer users. Much of the material is concerned with socio-technical aspects of implementing computer systems and logistic and implementation studies. Many are based on questionnaire studies asking users for their views on how things could improve. While this material is useful in a general sense for computer designers and implementers, there is little application of cognitive principles. For example, the review cites as "Psychological Factors" the following synopsis of recommendations (Shneiderman 1979, Eason & Damodaran 1981):

1)Reduce anxiety by clear instruction

2)Instill positive attitudes toward computers

3)Create approachable computers.

Allwood (1986) reviewed literature on novices and observed the great variability among novice users on different performance measures, and an issue that needs future attention is that of individual differences and what factors contribute to these differences.

2.4.2 Command and Icon Generation

There have been more applicable studies done more recently that are related to the concept of applied cognitive theories. Lindgaard & Perry (1987) (see also similar study by Grudin & Barnhart, 1984) studied a group of 72 subjects, half of whom were computer novices with no computer experience, and half of whom were moderate computer users (less than 1 year experience). The experiment used a set of four simple tasks using an electronic mail system simulated on a personal computer. The interface dialogue used was a forms-based system, where users had to fill in the blanks. Some users were first presented with a choice of labels to be used to describe the function to be performed, while others were forced to use a predetermined label. The list of labels is reproduced in Table 2. They found that novices using self-picked labels learned to perform the tasks faster than the predetermined label group. They also found that the results held after an interval of four weeks, providing some support for the value of this approach in improving long term retention. It should be pointed out that these users simply picked which label they preferred from a set of available labels. An interpretation of why their performance improved relies on previous studies by

Table 2
(from Lindgaard & Perry, 1987)

The three lists of vocabulary generated on the basis of preference ratings obtai[n]
an earlier experiment.

Element	LABELS List A	List B	List C
1	Sender's name	Sender	Originator indication
2	Sent on behalf of	Authorization	Authorizing user's indication
3	Message number	Message identification	Sender's message number
4	Message description	Subject indication	Subject
5	Time sent	Submission time	Submission time stamp indication
6	Time received	Delivery time	Delivery time stamp indication
7	Primary and copy recipient's indication	Original or copy	Original or copy recipient's indication
8	Reply requested	Answer requested	Reply request indication
9	Replying to previous message	In reply to	Replying message indication
10	Related messages	Cross-referencing indication	Cross-reference
11	Forwarded message	Message sent on	Forwarded message indication
12	Multiple recipients	Delivery to more than one person	Multi-destination delivery
13	List of other recipients	Disclose names of other recipients	Disclosure of other recipients
14	Speed of delivery (express, normal or low priority)	Delivery priority (high, standard or low priority)	Grade of delivery (urgent, normal or non-urgent)
15	Delivery confirmation	Delivery notification	Advice of delivery
16	Inability to deliver	Advice of non-delivery	Non-delivery notification

Hammond (1983) and Black & Moran (1982) which suggests that user choice can prevent interference from previous understanding of the label as used in their own, non-computer environment, as discussed earlier (see section 2.3.2). This dissertation will later propose (see P1, section 6.5) that self-picked labels provides for additional elaborative processing at the time of learning and so would be expected to provide better recall, consistent with propositional network theories.

2.4.3 Adapting the Interface

Further in this theme, Good, Whiteside, Wixon & Jones (1984) (see also Benyon & Murray, 1988) attempted to allow a novice interface to "organically" adapt itself by including common error command names that users attempted. This use of user synonyms made the interface more usable on the first attempt by 76% of subjects as compared to 7% for the unmodified original system. This study shows the power of adapting the system to the user's pre-conceived schema and labelling conventions.

Umanath & Scamell (1988) investigated recall performance using different data display formats. They found that recall for information possessing a spatial orientation, such as a sales forecast trend, is best recalled if presented in a graphical format rather than a tabular format. This result is consistent with cognitive psychology theories such as verbal-picture memory systems. However, this study result can be useful in improving the display techniques used in management decision support systems.

2.4.4 Hypertext and the Information Retrieval Task

Another related topic currently enjoying significant research activity is in the area of hypertext systems for improving retrieval and search techniques. A study by Lansdale (1988) was based on the findings of Bower and Winzenz (1970) that the active process of generation produces more effective memory traces (see also Graf, 1980). It found that if users generated their own icons, recognition performance was improved compared to when the system itself generated the icons. More interestingly they also found that recall performance improved as well, which is arguably the process requiring more additional support. These icons were used to retrieve documents contained in a filing system. Lansdale et al (1987) extended this work in a comparison of using either icons or words as retrieval cues, either system-provided or determined by the user. The study did not find any difference between visual and verbal modes, but rather that either is most effective when the user is able to make a meaningful association with the material to be retrieved. It is important to note that the user in both of these studies did not "generate" an icon or word completely unaided, but rather selected one he or she found most appropriate from an available list.

Jones and Dumais (1986) attempted to test for differences in retrieval of documents using name labels versus spatial (location) information. The results were largely inconclusive, although initial results indicated roughly equivalent performance when a small number of objects were to be remembered. However, this deteriorates in favour of using names when a larger object population is used.

Egan et al (1989) use a technique called "rich indexing" to improve the user retrieval performance of their hypertext-based "superbook". As part of this technique, and because they have found that users (even subject area experts) often use many different terms to describe the same activity or concept, the search system builds up a repertoire of aliases to improve the search operation. As users use various words to search a topic, the system keeps track of the terms used, and when the user eventually finds the desired information, the system allows the introduction of a new "alias" based on the frequency of users attempting to use the same term. This technique results in markedly improved first attempt retrieval by users.

2.4.5 Implications of Previous Studies

The implications of the studies discussed in section 2.4 and their relevance to this research can be summarized as follows:

1) Some previous studies have focused on novice user performance and results of selection (not free generation) of command names but neglected to use any underlying cognitive theory of memory or learning.

2) Research into adaptive interfaces and hypertext systems indicates that user recall improves significantly if user synonyms (rich indexing) for command names or index terms are allowed.

3) Recall performance improves for information if the system representation is consistent (eg. graphical (pictorial) for spatial information).

4) Self-selected icons provide better recall performance in a file retrieval task than system provided icons. However, the extension to command names or functions has not been studied.

CHAPTER 3

Development of a Framework for Human-Computer Interface Learning

Chapter three now looks at various classifications of models of human-computer interaction, individual differences and their potential effect on learning, as well as two particular models of learning which are most applicable to the later development of a learning strategy and measurement model.

3.1 Literature Review of Models and Measurement of Human-Computer Interaction

The decade of the 1980's witnessed a "race between (computer) function and usability. New technologies and new capabilities became available to users faster than user problems can be studied, understood and addressed" (Carroll, 1989). A review of the research into understanding and modelling human-computer interaction reveals a controversy revolving around the "level of approach" to the problem (Frese, 1989). Frese defines the two extremes of approach as a molecular (low-level) approach, also referred to as a quantitative approach (Williges, 1987), which analyses human action on the lowest level, and a molar (high-level) approach, also referred to as a conceptual approach (Williges, 1987), which studies human interaction in terms of high-level ideas, metaphors and paradigms rather than detailed quantifiable concepts. For a time in the mid 1980's, each camp seemed to justify the benefits of its approach by criticizing the other (Newell and Card, 1985 and Shneiderman, 1984). However, by the end of the decade, many researchers were adopting a more catholic view, and including elements of both approaches (Williges, 1987 and Carroll, 1989). We will now examine the approaches in more detail.

3.1.1 Molecular (Quantitative) Models

Williges (1987) provides a classification of quantitative models that groups models into the following types : Performance Models, Ergonomic Models, Simulation Models, and Statistical Models.

3.1.1.1 Performance Models

A performance model is one which attempts to evaluate an interface based on direct empirical measurement. This area has been strongly influenced by research practice in experimental psychology, with emphasis on tightly controlled laboratory approaches (Carroll, 1989). Probably the most influential of this model category is the keystroke-level model to evaluate user keying performance (Card, Moran, & Newell, 1983). The model was based on Fitt's Law (Fitts, 1954) and was used to optimize interface keyboard layout. It is largely based on an idealized view of tasks to be performed, and some argue that it is too steeped in the work of Taylor (Frese, 1989). Others argue that it fails to predict performance in a real world environment, and that indeed its lack of capability to factor in errors makes it naive and impractical (Shneiderman, 1984 and Greif, & Gediga 1987). However, the authors (Newell & Card, 1985) and others (Carroll & Rosson, 1985) argue that direct empirical measurement is, at the end of the day, the only adequate means of assessing the comparability and usability of an interface design. Indeed Newell & Card (1985) argue that psychology will be driven out of the mainstream of human-computer design by computer science unless it can develop predictive models, and coined the term "hard science drives out the soft".

3.1.1.2 Ergonomic Models

Ergonomic models are concerned with workstation layout and design and utilize biomechanical data to optimize the interface. This type of model has deep foundations in traditional human factors research, and is widely used in specialized human-machine design such as aircraft cockpit design. It is not widely used in the area of software interface design (Williges, 1987), although it is currently gaining some acceptance as part of expert systems used to design workstations (Gawron, Drury, Czaja & Wilkins, 1989).

3.1.1.3 Computer Simulation Models

Computer simulation models of human-computer performance are typically task-network models which allow simulation of tasks and subtasks. They provide a predictor of time to complete tasks and other fundamental variables (Chubb, Laughery & Pritsker 1987). The use of simulation models is recent, and appears not to have been widely used in interface design or modelling to date. Simulation may prove useful in assisting in early design work, to be later verified with human studies once a prototype is available (Williges, 1987).

3.1.1.4 Statistical Models

Williges (1987) classifies a last category of quantitative models as statistical models. These use statistical procedures to build models of user behaviour. Typically, clustering algorithms are used to postulate structures representing the user's model of the interface and task. One such example is a study by Tullis (1985) which used clustering procedures as a means of capturing users' perceptions of relationships among system functions (Williges, 1987) It could be argued that this category is simply a step in a procedure used to develop one of the other types of quantitative models.

3.1.2 Molar (Conceptual) Models

The opposite "mainstream" of human-computer interface modelling is the so-called Molar or Conceptual type of model. Most of this work starts with a general model of the human cognitive processes involved in information processing. It then progresses to develop methods and paradigms to infer the processes and apply the principals to the interface design (Williges, 1987). These model types have been classified by Williges (1987) based on cognitive psychology definitions proposed by Mayer(1981) as follows: Cognitive Processes, Cognitive Structure, and Cognitive Strategy. Definitions and examples of each type are now reviewed.

3.1.2.1 Cognitive Process Models

Cognitive process models are concerned with identifying the procedures individuals use in performing an information processing task. One method that is widely used to capture these procedures is known as verbal protocol analysis

(Ericsson & Simon, 1984; Simon & Simon 1978; Newell & Simon 1972; Simon 1980). Subjects are asked to continually describe aloud their mental processes as they are performing a task (such as using a computer interface). These reports are reinforced by the experimenter during the test by probing questions, using a non-evaluative tone of voice, as the subject is performing the task(such as "What are you looking at now?", "Why did you do that?" etc). This information is typically recorded on a tape recorder (technique from Newell & Simon 1972; Evans 1988).

An example of a very general information processing model that has been used as a basis for additional procedural refinement was developed by Wickens (1984) (Williges, 1987). This model is not to be taken literally, but rather is used to infer a possible model of human information processing (see Figure 8).

The work of Norman (1986) has proven to be influential. His model identifies seven stages of user activity when using a computer for an information processing task. These stages are : establishing a goal, forming an intention, specifying the action sequence, executing the action, perceiving the system state, interpreting the system state, and evaluating the system state with respect to the set goals and intentions. Frese(1989) acknowledges the fact that Norman's concepts

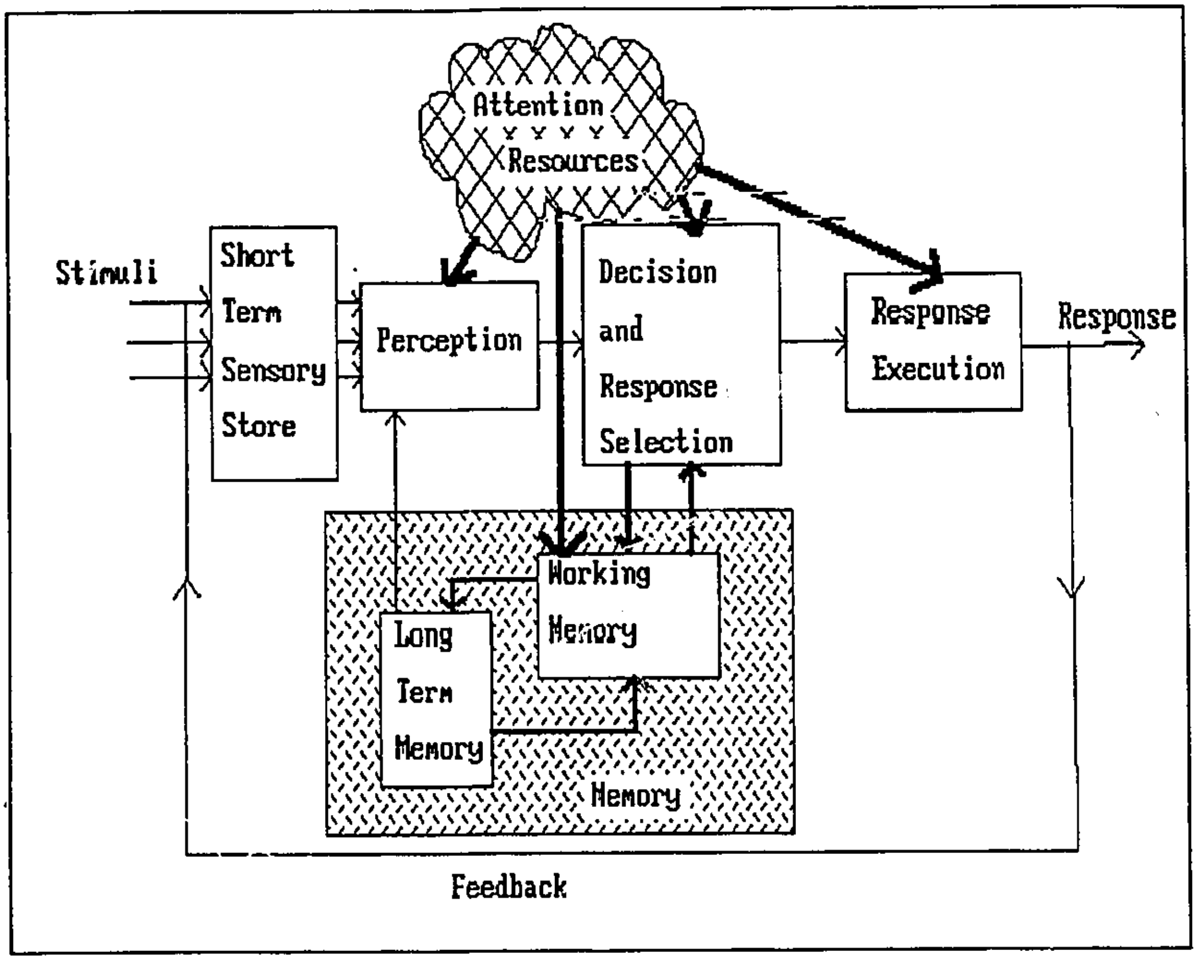


Figure 8
General Information Processing Model
(from Wickens, 1984)

"cannot be measured in milliseconds" but argues that they may represent a more life-like representation that provides an interface designer with a useful user model to contrast against the designer's own model.

3.1.2.2 Cognitive Structure Models

Cognitive structure models attempt to represent the knowledge utilized in a task as a hierarchical network. An example is a model of task complexity by Kieras and Polson (1985) (also Kitajima, 1989). This model attempts to decompose a user goal using a collection of production rules. They propose that a designer can map this user model of the task structure to a user defined model of the device structure. These models have not yet achieved a significant amount of testing in a design environment, but are proposed as a hypothesis for improving task-to-device mapping (Williges, 1987).

3.1.2.3 Cognitive Strategy Models

Models that allow for the selection of multiple approaches in task completion are classified as cognitive strategy models by Williges(1987). An ambitious and comprehensive attempt at a complete cognitive strategy model (Carroll, 1989) is the GOMS model (Goals, Operators, Methods and Selection rules) developed by Card, Moran and Newell (1983). This method allows users to hierarchically decompose their goals into subgoals. These are then matched to a basic set of methods which represent a strategy appropriate to the user. Indeed, it is from this overall GOMS model that the quantitative keystroke-level model discussed earlier appears as a submodel.

3.1.3 Criticism of Molecular vs. Molar Models

As discussed in the introduction to this section, much of the debate in the 1980's centred around the pro's and con's of each modelling technique. The molecular (quantitative) camp could be summarized as follows (Frese, 1989) (based on Card, et al 1983):

a) an approximate quantification is better than none,

b) only a hard science (quantitative laws) will be accepted by system designers,

c) this approach allows for analysis of design alternatives before they are actually designed.

The molar (conceptual) researchers countered point a) by questioning the usefulness of the quantification. Landauer (1987) (reported in Carroll, 1989) observes that while experimental psychology routinely "focuses on the significance of effects, it typically disregards the size of the effects." He goes on to argue that it may not matter if experts reliably chunk information 2% more than novices."

Shneiderman(1984) and others (Greif and Gediga, 1987; Frese, 1989) argue as well that concentrating on such micro-variables is not exact but "pseudo-exact" because important considerations such as comprehensibility and memorability are ignored for the sake of minimizing keystrokes.

A more satisfying view has been proposed by Williges (1987) and Carroll(1989). Williges argues that "no single, complete model of the human-computer interface exists...and both conceptual and quantitative models are needed to provide an adequate description of the user... A rigid adherence to one approach or model is counter-productive to understanding the true complexity of the user interface." In 1989 Carroll recognized that "an orderly evolution of HCI (human-computer interaction) work has produced a paradigm that builds upon...human factors evaluation and cognitive description and at the same time redresses their limitations with respect to design impact and the ecological validity of empirical work".

3.1.4 Criticism of Existing Models

While the various modelling camps discussed above appear to be reconciling their differences to a degree, there is yet another community represented primarily by system practitioners and designers that have reservations about much of the modelling activity to date. Booth (1989) argues that although models attempt to identify potential difficulties that users might encounter, they do not inform the designer of the real problems that users experience, may tempt designers to use models or computer simulations to direct design, and omit actually involving users in the design process. In addition, he argues that a major difficulty of almost all modelling is the question of what granularity of analysis to employ and uses the example of the breakdown of the task of making a cup of coffee. It may be possible to break this task down into 13 subtasks, 30 subtasks, or even 50 subtasks depending upon the extent to which the task is broken down (granularity of analysis). Many models (example:Kieras & Polson,1985) equate cognitive complexity with the number of rules required to describe a task. Booth argues that such a model may predict a complexity level based not only upon the true complexity of the task, but also on whatever granularity of analysis was employed.

Another recurring theme among critics of existing models is that they simply do not fit into the design process and are too complex to use and understand on an everyday basis (Carroll & Rossen, 1985). Moran (1986) states that "Models (using analytic techniques) play a role as elements of the design argument, helping to support certain decisions. Thus, models are best applied locally to small aspects of the design, where they fit the grain size of the design decisions." Whiteside & Wixon (1987) suggest that future research be motivated by practical interface problems. They contend that "most designers begin with an existing design and then seek to improve it; or they may take elements from a number of designs and combine them in unique and new ways...In our experience (at Digital Equipment Corp.) empirical evaluation of existing interfaces has frequently had a profound impact on new designs." They argue that if future models would adopt a similar, applied approach, they would be both powerful and more useful.

3.1.5 The Usability Approach to Interface Design

In order to better understand how new models and tools might be more useful in design, we now look at the concept of usability.

There are many design methodologies based on the concept of making the computer interface more usable (Shackel, 1986; Shackel, 1988; Bennett, 1984; Bury, 1984). The Gould & Lewis (1985) definition is representative and is as follows: A usable system is "any system designed for people to use...(that is) easy to learn (and remember), useful, that is contains functions people really need in their work, and is easy and pleasant to use". They recommend three principles (steps) to achieve usability:

1) Early Focus on Users and Tasks-- by studying user cognitive, behavioral, anthropometric, and attitudinal characteristics, as well as the nature of the work to be accomplished.

2) Empirical Measurement-- early in the design process of prototype designs.

3) Iterative Design-- when problems are found in user testing, they are fixed and the process continues iteratively, repeated as often as necessary.

While there is no generally agreed list of empirical measurements to be performed on an interface prototype (Brooke (1990), there does seem to be a consensus that variables such as usefulness, effectiveness (or ease of use), learnability and attitude (or likeability) are important metrics in determining the degree of usability of an interface. A survey by Booth (1989) lists various measures that have been used to determine usability scores, including time to complete a task, errors that occur, etc. A variety of methods exist to determine these variables, such as attitude measurement, verbal and visual protocol analysis, patterns of use, structured walk-throughs, and both laboratory and field trials.

A particular weakness in existing test methods, however, is in the area of determining measures of learnability. Bennett (cited in Shackel, 1981;see also Booth,1989) suggests that it would be possible to measure learning time, but that this measure is confounded by the fact that people often learn at different speeds and in different ways. A more recent assessment of the problem by Shackel (1988) states that "the problem with these definitions is that they are conceptually satisfactory but still only generalized in form: they do not specify what is usability in quantifiable or measurable terms". He then proposes that objectives should be set for learnability measures for the following metrics:

1) Learned within some specified time from commissioning and start of user training.
2) Based upon a specified amount of training and user support.
3) Relearned within some specified relearning time each time for intermittent users.

However, Shackel does not provide details on how or what to measure to determine these metrics and states "..there is still much to be done in modifying, developing and testing usability evaluation procedures for human/computer interaction."

Farooq & Dominick (1988) surveyed tools and models for developing user interfaces. They note that "although one should generally be sceptical of questionnaires and interview studies, these are often the only direct measures of user assessments available to researchers." They later state that "From this survey, a number of directions for future research have become apparent" including "Automated monitoring of experimental data: The utilization of automated user/system interaction monitors should be absolutely essential for supporting serious and objective evaluations of alternative user/system interface prototypes and designs. Without the availability of such monitored data, assessments of interface effectiveness often become subjective conjectures, rather than statistically defensible results derived from objectively measured phenomena. Considerable additional research attention needs to be given to the utilization of such monitoring mechanisms within user interface behavioral experiments" (emphasis mine).

There are those who criticize usability testing and indeed prototype testing in that it focuses on low-level details and tasks. However, Gardiner et al (1987) argue that "while it is important to realize that this may indeed be the case, it is hardly a crippling criticism of the method; no one claims that prototype testing does the complete job. Evidence from user testing must always be interpreted, and the interpretation may well reflect more architectural issues; in addition, the aim of testing at earlier stages ...is to provide insight into just such issues."

Although the concept of relearning performance is discussed in the literature, there still appears to be a continued focus in recent research on using measures of learning speed and initial error performance as a surrogate for learnability performance. Whiteside, Bennett and Holtzblatt (1988) suggest a possible usability measurement criteria checklist of 22 items which includes time to complete a task, percent or number of errors etc, but does not focus on recall performance. Karat (1990) reports on a usability testing and cost-benefit analysis study of error performance. However, the test focus is on error performance during the learning session only which spanned a matter of minutes. No attempt was made to test the performance of the interface impact on long term memory or retention.

It should be stressed that usability is a new and evolving design and evaluation technique, and that none of the previous authors suggest that their list of measures is totally exhaustive. Indeed, it is argued (Whiteside, et al 1988) that these checklists are only meant to aid in the creative process of developing usability attribute definitions that are situation, task and context driven. We suggest throughout the next chapters that there may be additional utility to adding the concept of recall as a fundamental means of testing learnability.

To summarize this section, designing human-computer interfaces within a usability framework promises to provide interfaces which are useful, easy to learn and easy to use.

Many evaluation techniques have been proposed. However, there are only beginning to emerge clear methods and definitions available to help designers measure the learnability metrics related to a usable interface in a practical and useful manner. In addition, there is a paucity of existing objective evaluation tools available to the research and design community relating to learnability using both learning speed and recall measures as evaluators.

3.1.6 Summary of Existing Models

The relevance of this review of existing models to the research can be summarized as follows:

1) A high level conceptual model is useful in guiding interface design by providing an underlying theory base of both the user and the system.

2) A low-level quantitative model is useful in providing a basis for evaluation and measurement of an interface design.

3) The design community finds many existing modelling attempts non-practical and not designer-oriented.

4) A general framework and approach to measure interface learnability and recall would appear to be consistent with a usability approach to design, and may fulfil an existing practical methodology need among system practitioners.

We will develop a framework to measure computer interface learning and recall in chapter 4 which exhibits elements of both conceptual and quantitative model types, and is intended to be practical and consistent with a usability approach to interface design.

3.2 Literature Review Pertaining to Individual Differences

Any modelling of human-computer learning must take into account the area of individual differences. We will examine the types of differences from two perspectives: differences affecting individual performance, and differences that identify a preferred cognitive style or individual trait.

3.2.1 Task Performance Differences

It is well known that individual differences in specific abilities is a dominant predictor of task performance (Morrison & Nobel, 1987). Indeed, one of the major problems identified by Bennett (1984) discussed above is that people learn at different speeds and in different ways. Therefore, one of the keys to determining a metric of learnability is a method of normalizing out these individual differences in learning time. One way to accomplish this is of course to have a large random sample of people in a test of learning time. This is counter to actual use in a practical "real-world" development environment. However, a potentially promising direction is to make comparisons of exhibited retention levels. Farr's (1987) survey concluded that higher ability learners tend to achieve higher levels of learning than less able individuals (measured using a broad measure of general intelligence). As a result, when decay occurs over time, the more able learners will maintain their learning advantage. However, there is no retention difference between the differing-ability groups if they have both learned to the same mastery criterion (Hurlock & Montague in Farr, 1987). This distinction is an important one for the purposes of the model proposed later in this paper (chapter 4) in that for purposes of comparison, it is important to measure whether or not a user has achieved a benchmark level of skill acquisition (i.e.-cognitive, associative or autonomous). Then, it will be possible to make more meaningful comparisons of user performance in skill retrieval after some fixed delay, without as many confounding effects of individual ability, although the absolute effects of individual ability will never be totally "normalized" (Farr, 1987 p. 94).

There are many other categories of individual abilities besides general intelligence that may also affect performance. Sein et al (1987) reported that subjects that have a high score on visual ability performed better in a computer learning task that required manipulation of graphical models, which is consistent with other studies. Visual ability is simply one of many specific individual differences that will affect individual performance. It is important to recognize that these differences must be measured or controlled by the experimental procedures used if a test is attempting to isolate learning effects due solely to a computer interface.

In addition to ability, it has been suggested that other experience and attitude variables <u>may</u> impact the level of task performance (Gist et al, 1989). Additional individual differences potentially confounding task performance include (Browne (1990)): psycho-motor skills, understanding, expectations, motives, temporal changes, preferences and situation specificity.

Experience has been shown to affect computer task performance, and one would expect users with a high degree of experience to perform tasks more quickly. However, depending upon how the task is measured or scored, the opposite may be true. Lewis & Norman (1986) point out that error types such as slips (A person establishes an intention to act. If the intention is not appropriate, this is a mistake. If the action is not what was intended, this is a slip) may occur more often with experts than with beginners due to experts' highly automated performance and lack of focused attention. This points out the necessity of identifying between these error types in any measurement of human-computer task performance.

Parasuraman & Igbaria (1990) have shown that among managers, differences such as gender, personality (using Meyers-Briggs, 1962 classifications) , demographics, education and mathematics anxiety all have relationships to computer anxiety and computer attitudes. These individual difference effects point to the necessity of measuring or controlling any experimental study of computer task performance in order to mitigate for their potential confounding influence.

3.2.2 Learning Style Differences

We now turn to the area of individual differences that classify the user's personality traits, problem solving approach, or so-called cognitive style. There are various measurement instruments that have been used to determine what these differences are. Two widely used and verified instruments are the Meyers-Briggs Type Indicator and the Learning Style Inventory (LSI). Jung's psychological types are the basis for the Meyers-Briggs instrument (Meyers-Briggs, 1962). This uses a self-reporting questionnaire to determine individual scores along the following four personality continuums: extrovert-introvert, judging-sensing, sensing-intuition, and thinking-feeling. The Meyers-Briggs instrument has been widely used in studies throughout the literature. However, some argue that it may not correctly represent the Jungian Types (Kolb, 1984).

A second indicator, the Learning Style Inventory (LSI) (Kolb, 1984), attempts to measure the cognitive style of the individual when learning or problem solving. It uses four measures or types of learning as its variables:

<u>Convergent</u>- Knowledge is organized and focused to a result.
<u>Divergent</u>- Generates multiple creative result paths
<u>"Gestalt" Assimilation</u>- Idea-centred deductive results
<u>Accommodative</u>- People-oriented action results.

Both of these tests are simple to administer, and can provide valuable insight into the effectiveness of some of the proposed interface alternatives with various individual cognitive types. The LSI instrument exhibits strong reliability and each of the four learning styles is highly correlated with the personality types used in the Meyers-Briggs Type Indicator (Kolb, 1984). Use of the LSI would allow for comparison to the body of previous literature in many areas that used the Meyers-Briggs instrument for subject typing.

TABLE 3

<u>User Cognitive Features</u> (from Murray, 1988)
Learning Style
Cognitive Style
Spatial Ability
Task/system expertise
Short Term Memory (STM)
Working Memory
Learning Strategy
Risk-aversion
Levels of Attention, Confidence, & Aggressivity

Recent work in the area of designing adaptive systems has focused on methodologies for determining individual differences that will affect the way the system will adapt, in order to optimize user performance (Browne (1990)). Murray (1988) has surveyed the literature as it applies to variables of potential importance to user profiles for use in embedded user models in adaptive interfaces. These are summarized in Table 3.

Murray proposes that, although each of these variables could be individually scored, a more realistic implementation would be to use an artificial intelligence knowledge base that would be controlled by initial input parameters from a user profile.

3.2.3 Summary of Individual Differences

The impact of individual differences relevant to this research can be summarized as follows:

1) Individual differences in ability will greatly affect speed and error performance of a learning task. However, in measuring retention, the confounding effects can be minimized to some degree if subjects have learned the task to a benchmark mastery criterion.

2) Many other individual differences, including experience and attitude, may impact the level of task performance, although not always in an easily predictable way.

3) The Learning Style Indicator appears to be one of a number of relevant instrument for use in measuring subjects cognitive preference in a learning task.

We will utilize the LSI and measures of experience, attitude and other individual differences easily measured in the framework and test discussed in later chapters. In addition, the framework proposed in chapter 4 recognizes the importance of ensuring all subjects achieve a common task mastery level, in order to make meaningful evaluations of interface implementations.

3.3 Literature Review of Relevant Learning Models and Theories

We now examine two models relevant to learning. One model provides a useful framework of learning strategies appropriate for various tasks to be learned. It has elements most appropriate for application in an information processing environment. The second model is a general model of long-term retention of knowledge and distills a set of variables useful in determining the effectiveness of learning.

3.3.1 Generative Learning Strategies

The first model we examine is Generative Learning Strategies (Jonassen, 1988) based on the generative hypothesis (Wittrock, 1974) which is that the meaning for material learned is generated by activating existing knowledge structures in order to interpret what is being presented. Generative learning activities require learners to "consciously and intentionally relate new information to their existing knowledge rather than responding to material without using personal, contextual knowledge" (Jonassen, 1988). Jonassen (1988) lists the following examples of generative learning : generating mnemonic memory aids, notetaking, underlining, paraphrasing, summarizing, generating questions, creating images, outlining, and cognitive mapping. He asserts that by use of these activities, information is elaborated into a more personal form that is more memorable to the user. Jonassen then defines learning strategies (similar to cognitive strategies), based on definitions by Rigney (1978), as "Mental operations or procedures that the student may use to acquire, retain, and retrieve different kinds of knowledge and performance". (see also discussion of GOMS model by Card et al in section 3.1.2.3 above). Jonassen provides a taxonomy of learning strategies synthesized from other classifications (Weinstein et al, 1985; Dansereau et al 1979; Rumelhart et al, 1977; Norman et al 1976). This taxonomy is shown in Figure 9.

The area of the Jonassen taxonomy (see Figure 9) relevant to this study is the primary strategy classification "information processing strategies". Jonassen expands on each of the four sub-categories by identifying the relevant cognitive processes involved, and identifies appropriate learning activities consistent with these processes. This is summarized in Table 4.

Jonassen's taxonomy has applications in the area of computer based training of any topic, as well as training in the use of the computer interface itself. It appears to be a very applicable and practical taxonomy relating the theories discussed earlier on appropriate cognitive learning approaches. We now turn to a second framework by Farr (1987) which focuses on learning variables.

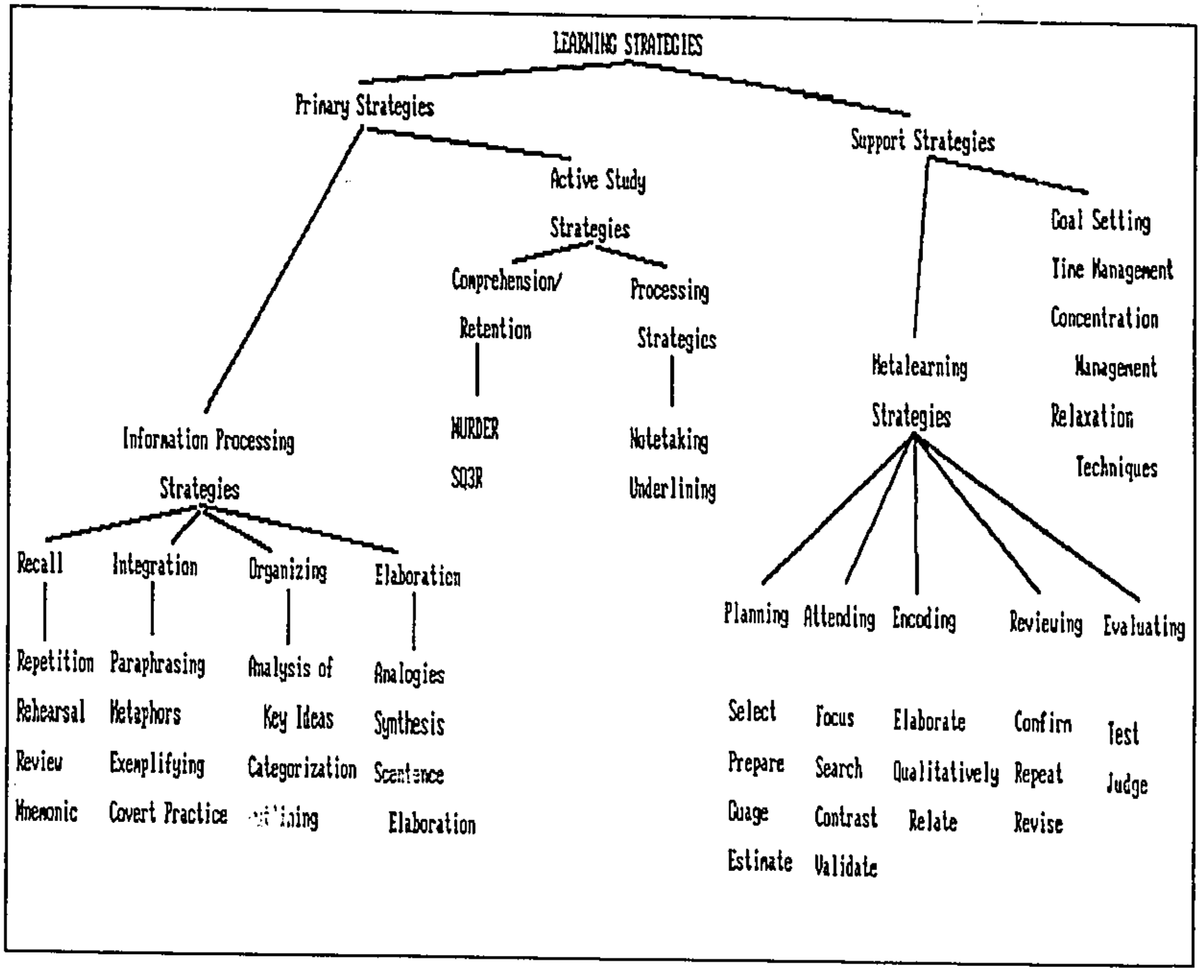


Figure 9
Jonassen Taxonomy
(from Jonassen, 1988)

TABLE 4
INFORMATION PROCESSING STRATEGIES
(from Jonassen, 1987)

LEARNING STRATEGY	COGNITIVE PROCESS	LEARNING/INSTRUCT. ACTIVITY
RECALL		
Repetition Rehearsal Review Mnemonic technique	Behav. theory Verbal learning Chunking Connectionism -practice effect	Repeat processing, such as re-reading a text passage. Repetitive practice in recalling information without reorganization.
INTEGRATION		
Paraphrasing Metaphors Exemplifying Covert practice	Schema theory Network memory theory	Restate information in your own words. Generate additional examples of the idea. Mentally practice learning behaviours. Complete practice activities (exercises).
ORGANIZATION		
Analysis of key ideas Categorization Outlining Networking Pattern noting Cognitive mapping	Associative memory Node-link model of long term memory	Identify key concepts, develop definitions, and compare with oth. concepts. Classify concepts according to taxonomy / class scheme. Develop hierarchical list of concepts in materials. Identify nodes, classify links, and diagram, linking ideas to each other producing a map of concepts. Pairwise comparison strength of relationships and statistical scaling to produce a map.
ELABORATION		
Imaging Analogies Synthesis Sentence elaboration Implications Drawing inferences	Dual coding Encoding variability Schema theory Problem solving Higher order thinking skills	Mental images /create drawings that describe content. Complete analogies or generate own. Add descriptive information - make material meaningful. State implications for personal fulfillment. Infer causes for outcomes/ events.

3.3.2 Long Term Retention of Knowledge and Skill

A relevant literature review and analysis of learning theories and measurement variables was done by Farr (1987). This study was an attempt to review the variables affecting the long-term retention of knowledge and skills. While it does not concentrate on computers, it does provide a good summary of the research as it pertains to training individuals with no prior experience in particular tasks. The study focus was on describing and ranking the major variables that affect long term retention. It then maps these variables to the relevant cognitive process theories, identifies strategies to promote long term retention, and recommends methodologies to retard decay attributable to little or non-use of a learned knowledge or skill. Farr organizes his conclusions into six variables that are a useful framework in approaching the study of skill acquisition. These are summarized in Table 5.

3.3.3 Summary of Learning Models

The Jonassen taxonomy will be used in the model of computer learning developed in chapter 4 as the basis for selecting an appropriate learning strategy based on the task type and complexity. The Farr variables will be used to contribute to the definition of measurement and evaluation variables in the same model.

TABLE 5
VARIABLES AFFECTING LONG TERM RETENTION
(from Farr, 1987)

Variable 1: Degree of Original Learning

The single most important determinant of knowledge and skill retention is the degree of original learning (OL). The amount of decay can be reduced by enhanced learning or over-learning.

Variable 2: Task Characteristics

Characteristics are 1) TYPE and 2) COMPLEXITY. Continuous-control tasks such as a motor task (eg. tracking something on a screen with a mouse) are retained better, even for extended time periods than a discrete or procedural task (usually verbally mediated or executed). Also, the complexity of the task determines acquisition time and long term retention. The more cohesive or integrated a task is, the less it will decay. This also applies to the degree of learner-imposed organization on the task.

Variable 3: Retention Interval

In general, the longer the period of non-use, the greater the decay. However, the amount of decay is sensitive to the task characteristics, degree of original learning and the method of training. Also, decay can be reduced during the retention interval by mental rehearsal or "imaginary practice".

Variable 4: Instructional Strategies/ Conditions of Learning

Programmed instruction leads to better retention. As well, the degree to which the material received elaborative processing will affect retention.

Variable 5: Methods for Testing - Conditions of Retrieval

Recognition generally yields better retention than recall.* Skill retention scores will be increased if the measurement is conducted in a context similar to the original learning.

Variable 6: Individual Differences

Higher ability individuals tend to reach a higher degree of original learning faster than lower ability individuals. However, once the same level of learning is achieved, both high and low ability individuals appear to exhibit similar retention ability.

*Note:There are some exceptions. See section 2.2.3 discussion of Tulving & Thompson, 1973.

3.4 Normalized Performance Ratio

Moffat (1990) has proposed a metric referred to as the Normalized Performance Ratio (NPR) which is defined as the value of the mean of the periods of time required by a group of people to complete an identical task with the system (the mean of the completion times), divided by the sample standard deviation of those completion times. In other words, the NPR is the reciprocal of the completion times coefficient of variation. Moffat proposes the following steps in determining an interface NPR:

1) An experimentally-unbiased sample of human subjects is assembled who have no experience with a given human-computer interface (HCI).

2) The subjects are trained (it would appear in a classroom setting using paper documentation) until they represent an evenly distributed range of understanding about the HCI's expected behaviour from no understanding to an expert level of understanding of the operation of the HCI.

3) The amount of time (the completion time) required by each subject to complete an equivalent processing task with the HCI is measured.

4) The mean of completion times is divided by their sample standard deviation to obtain the NPR value of the HCI.

Moffat contends that NPR scores for different HCI's and different tasks can be directly compared because the calculation of the NPR is task complexity independent.

Discussion of the NPR

Moffat's proposal of the NPR is heavily steeped in the Tayloristic belief that speed of task execution is a surrogate for the quality and value of an HCI, or more specifically the amount of variability in a subject population of the speed of task execution. It is similar in approach to the quantitative models and in particular the performance models discussed in section 3.1.1.1. In a mechanistic task such as document editing or database look-up, speed of operation and hence efficiency is an important aspect. However, the author appears to contend that because the NPR is independent of the complexity of the task and of the complexity of the human-computer interface it can " thus provide the basis for the unbiased comparison of all MMIs (Man-Machine Interface)".

It is difficult to agree with this assessment for a number of reasons. First, the NPR ignores the differences in task types between two measurements. As discussed in sections 3.1.4 and 3.1.5 many tasks involve utilizing an HCI in a decision support mode. In these tasks it is difficult to envision how variability of speed of execution can help predict the quality and hence the effectiveness of task execution. Second, the model takes into account task success or failure. However, it lacks inclusion of measures of degree of success or failure. For example, it fails to distinguish between or measure error types occurring during task performance.

The NPR is a commendable attempt to define a quantifiable and measurable metric of an HCI concerning task completion time. This is consistent with the industry demand for applied measurement and evaluation techniques discussed in section 3.1.4 and 3.1.5. As of this date, the NPR has not been implemented or tested. Moffat proposes a methodology for implementation in his paper. However, it utilizes a manual classroom training method using teachers and paper documents. After training, users then exhibit task proficiency on a computer interface and are measured using a stopwatch. This methodology could prove to be problematic with regard to confounding effects, particularly the context effects as discussed in sections 2.2.4. In addition, section 3.1.5 discusses the desirability of utilizing automated measuring techniques for providing statistically defensible results. An improvement to the methodology would be the provision of initial training using computer-aided techniques on the same HCI to be used for the measured task. This automated system could also provide all learning and task measurement without requiring external human intervention and stopwatches, etc.

One of the weaknesses in utilizing the NPR approach for the focus population of this dissertation (business managers) is the major assumption that "If the training and completion-time measurements involved in an implementation of the NPR metric occur within the space of a few hours or a few days then subjects may not lose much information through memory loss. However, if the implementation occurs over the space of several weeks, then the problems associated with memory loss, and need for review, may be significant." As mentioned in chapter 2, users can spend days or weeks between learning sessions and task implementation. For this reason, the NPR approach would be unsuitable for use unless long-term memory effects were added to the model and methodology.

Summary of NPR

1) The NPR is a first attempt at providing a generalized measure of HCI performance and is based on variability of task performance times of a subject population.

2) NPR would appear to have more value in evaluating highly structured, processor manipulation tasks than tasks resulting in decision support or task output quality.

3) The NPR methodology remains untried and untested. However, potential problems concerning context and measurement might be overcome using an automated teaching and measurement approach.

4) The NPR model ignores long-term memory and decay effects which limits its applicability to the business population of this dissertation.

CHAPTER 4

A Framework & Methodology for Selection & Measurement of a Computer Interface Learning Strategy

We will now develop a framework and methodology for selecting a learning strategy and measuring the degree of original learning and long-term skill acquisition for human-computer interaction related to the theories and models described previously.

4.1 Objectives of the Framework

Recalling the objectives mentioned in section 1.1, the objective addressed by this chapter is the first one, as follows:

Objective 1: to successfully synthesize from the existing body of work in both human-computer interface modelling, human learning, and memory theory a modified framework that specifically focuses on the issues of human learning and long-term retention of a computer interface. The framework must identify the processes and variables of interest in order to make evaluative decisions useful in improving real life interfaces. In summary, it must have a very practical and useful result in the real world of designing and modifying a human-computer interface by system designers and implementors.

Specifically, given that a designer has already completed the system and information requirements analysis for a proposed new or modified information system and has decided the tasks to be performed by the user on the proposed human-computer interface, the framework must:

1) Identify a procedure that will allow a system designer to select an appropriate learning strategy for the task to be implemented. This would be done before system design, and would guide the design of the task implementation.

2) Identify a limited number of metrics that will be easily testable in order to provide an evaluative, quantitative score of the implementation of the interface task. This score could be used to determine the quality or effectiveness of the particular task, or as a comparison tool for different alternative implementations of the same task. This testing is intended to be performed at the time of interface prototyping, and before final design and implementation.

The methodology is specific in its attempt to evaluate an interface along lines of effective learning and long term retention. In relation to the classification of models discussed in section 3.1, the framework exhibits elements of a "conceptual model" in its attempt to match and select an appropriate cognition based learning style, but has elements belonging to the "quantitative models" classification in that it attempts to measure a number of variables to provide a score of learnability and recall.

In developing a methodology for use by system designers and implementors, we recognize that a tradeoff is required between rigor and usability. An extremely detailed test methodology would require a vast battery of test variables that would make the testing of an interface far too time consuming. However, sufficient complexity is required to provide a mechanism to select among the various available strategies.

4.2 Development of Framework

As discussed in section 3.1.5, the usability approach appears to be consistent with the requirements of the design community (Gardiner, 1987) and has been gaining acceptance by this community (Vainio-Larsson, 1990; Lewis et al, 1990; Whiteside et al, 1988; Brooke et al, 1990). As discussed, Shackel (1988) proposed the following objectives for measures of learnability:

1) Learned within some specified time from commissioning and start of user training.

2) Based upon a specified amount of training and user support.

3) Relearned within some specified relearning time each time for intermittent users.

We propose to explore the concept of relearning required through the use of recall metrics. In addition, the methodology must be implemented in an automated fashion to minimize contextual interference and measurement errors which occur if human testing using external apparatus is used (Farooq & Dominick, 1988).

Finally, the methodology must above all be practical and usable itself as a design aid in order to meet the requirements of the design community (Gardiner, 1987, and Whiteside & Wixon, 1987).

The framework we now describe utilizes four functions. The first function is the method of selecting an appropriate learning and test strategy, and is called the Strategy Selection Function. The second and third functions are measurement functions used in the initial learning and the learning retrieval domains respectively, and are named the Initial Learning Measurement Function and the Learning Retention Measurement Function. The last function provides the designer with a quantitatively derived metric of interface performance and is named Task Learning Effectiveness.

We start with a modification of a general model of information processing proposed by Wickens (1984). The Wickens model is an attractive starting point, as it is a general model of human information cognitive processing and is not specific to computer applications but represents any human processing activity. It has also been used by others as a starting point for graphically segmenting the cognitive operators at work when performing computer tasks (see Williges, 1987) and represents a generally accepted view by many cognitive psychologists (Anderson, 1985, also Wessells, 1982). Also recall from the discussion of the model in section 3.1.2 that the operations outlined are not physical locations, but are rather representations of functional transformations (see Figure 8). We use the Wickens model as a visual metaphor rather than a literal schematic of the fundamental cognitive processes involved, and overlay the specific tasks and variables of interest to the measurement of computer task learning and strategy selection for the improvement of task learning. Because of the desired focus on task learning, we will represent the task and response as the only elements of Wicken's decision response and selection function. The modification of Wicken's general model to focus on computer task learning is represented in Figure 10. There are two condition domains of particular interest. The first we will call the Initial Learning Domain. This we define as the conditions present at the time of first or initial learning of a particular interface task by a user. Note that we allow that the user does not usually learn all the tasks available in an interface during one session, and therefore make the distinction that this learning domain is task dependent. The second condition domain we define is the Learning Retrieval Domain, which represents the conditions present at a time subsequent to the initial learning where the user attempts to retrieve and use (or apply) the particular task knowledge and skill learned in the initial learning session. Again we emphasize that these domains are task specific as a user may, in the

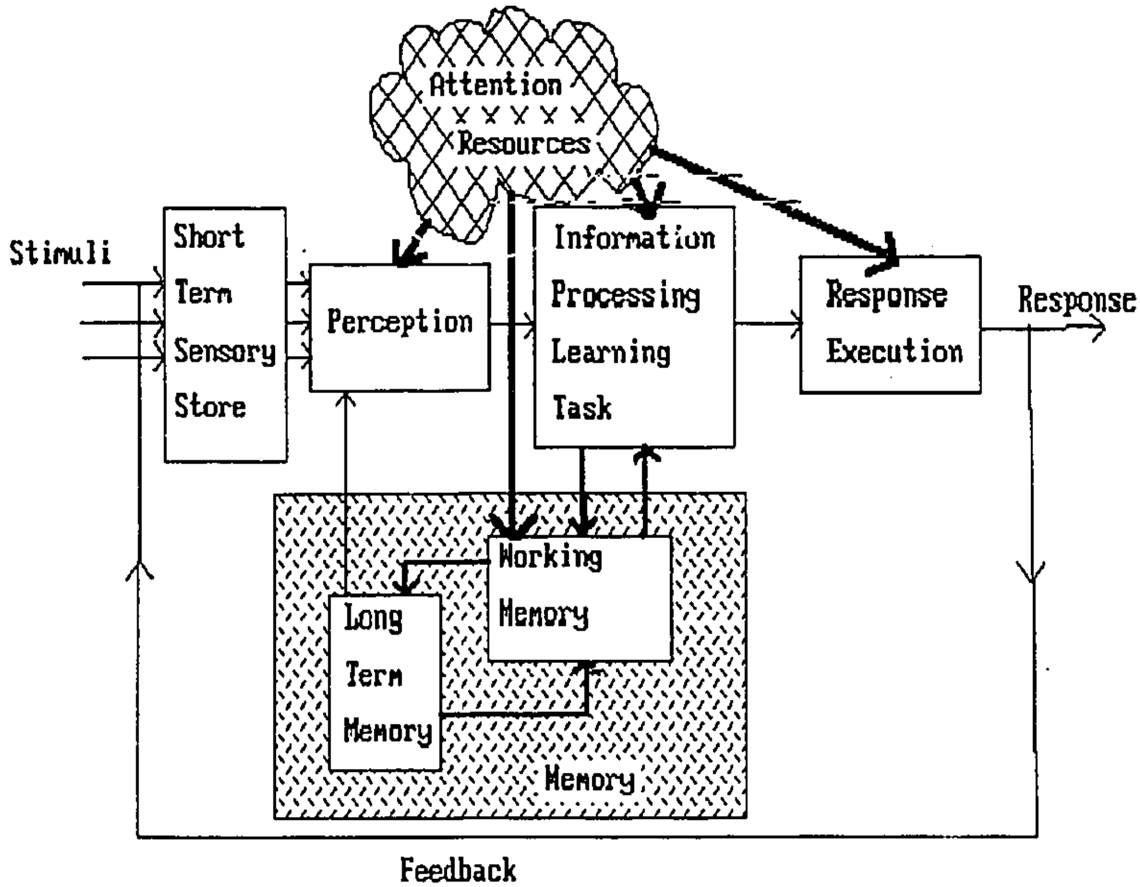


Figure 10
Modified General Task Model
(adapted from Wickens, 1984)

same user session, be in a retrieval domain and utilizing a previously learned skill (e.g. login procedures), and then move into an initial learning domain and learn a new task (e.g. an advanced editing function).

4.2.1 Initial Learning Domain

During the time of initial learning, we desire to improve the learning and memory of the task by use of an

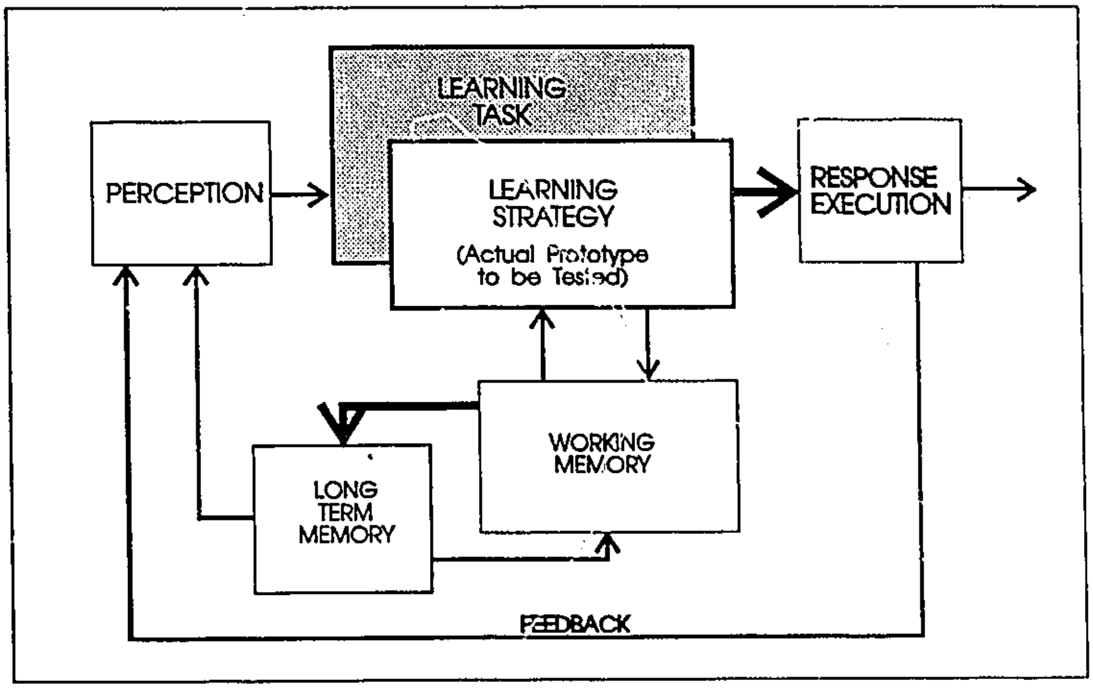


Figure 11
Initial Learning Domain

appropriate learning strategy that matches the task characteristics and the individual differences of the user or user group population. Figure 11 represents visually the areas we are attempting to improve. By utilizing the most appropriate learning strategy, we are attempting to increase the amount and quality of original learning achieved during the initial task exposure as discussed in section 3.3 (Jonassen, 1988 and Farr, 1987). This is represented graphically by the increase in flow (non-literal) from the task to the response execution functional representation. We are also attempting to increase the ability of the user to retrieve the knowledge and skill learned from long term memory at some time in the future (e.g. improved memory trace, "deeper" processing) (Farr, 1987). This is represented graphically by the increased flow (non-literal) from the task in working memory to the representation in long term memory.

We now overlay on this initial learning domain the following descriptors, variables and evaluation metrics as shown in Figure 12. First, we define the variables illustrated in Figure 12 and listed in the first part of Table 6.

ID Variable Set (Individual Differences)

The ID variables represent the variable scores obtained to identify individual differences of the subject population. As discussed in section 3.2, individual differences may impact task performance and must be measured or controlled in order to attempt to minimize their confounding effects on task learning performance. The instrument we used for this measurement is the Learning Style Inventory (LSI) (see chapter 3.2). The Learning Style Indicator appears to be appropriate for use in this task, based on its demonstrated use as a tool and its cross correlation with other instruments such as the Meyers-Briggs personality type identifier (Kolb, 1984). The variable

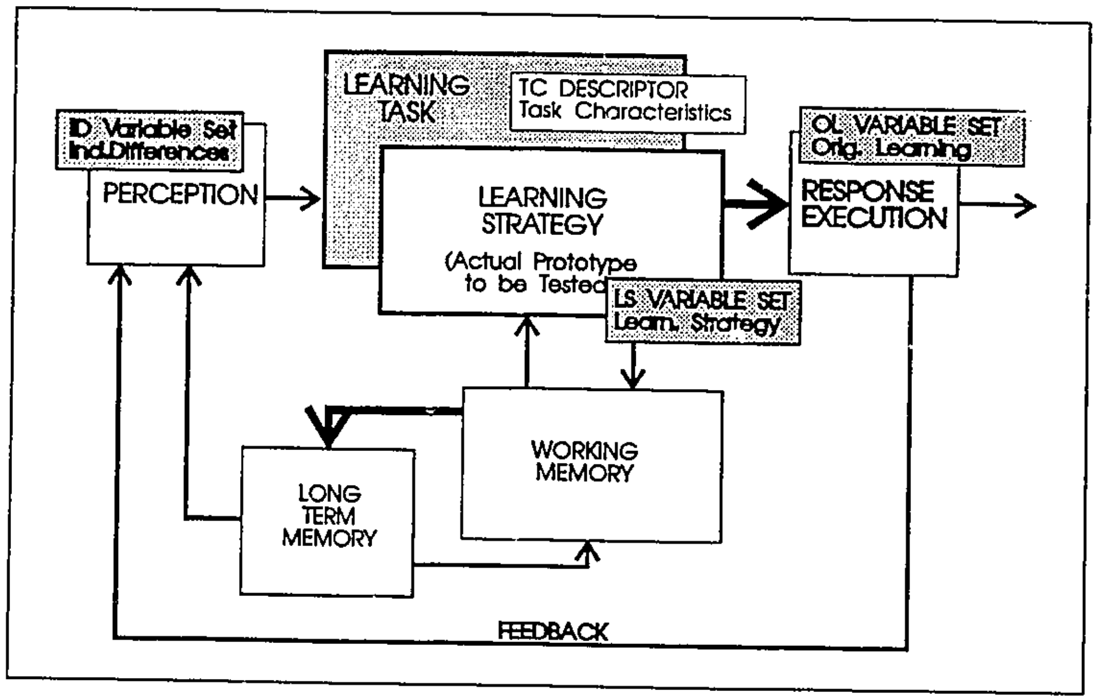


Figure 12
Initial Learning Variables

is a measure of the preferred learning approach of either the individual user, or can be used collectively to represent the dominant learning style of the total user population of the proposed interface. Other variables of individual differences may impact task performance as discussed earlier (section 3.2) such as attitude, experience, age and sex. These four variables are also measured in addition to the LSI to form a complete user ID profile for use in mitigating potential influences in interface performance testing.

Table 6

Framework Variables

INITIAL LEARNING DOMAIN VARIABLES

Individual Differences Variable Set (ID Variables)
* Attitude
* Experience
* Age
* Sex
* Learning Style (LSI)

Task Characteristics Descriptor (TC Descriptor)
* Task Type
* Task Complexity

Learning Style Variable Set (LS Variables)
* Appropriate Learning Style(s)

Original Learning Variable Set (OL Variables)
* OLtime
* OLerrors
* OLlevel
* OLpass
* OLiterations

LEARNING RETRIEVAL DOMAIN VARIABLES

Retained Learning Variable Set (RL Variables)

* RLinterval
* RLtime
* RLerrors
* RLlevel
* RLpass
* RLiterations

We are not proposing to apply the framework initially to multiple user groups, or refine its application to specific users on a dynamic use basis, as a so-called adaptive user interface (see section 2.4.3 above). Our intention for this research is to use the individual difference variables to control for possible individual difference effects when measuring individual performance on a specific interface task application. These variables are therefore concomitant or moderating variables. In addition to the above individual difference variables which the literature has stated may impact task performance, there remains the single individual difference which it is agreed will impact task performance, that of individual ability or general intelligence. We are not proposing to administer lengthy general intelligence tests, in an effort to keep the test methodology efficient and practical (consistent with other practical implementations-see Browne, P. 197). Instead, we will examine other measured variables as suitable surrogates during the exploratory experimentation described in chapter 6.

In addition to using the ID variables for test variance reduction, we will explore the use of dominant group LSI scores as potential designer enrichment cues, because LSI types have been shown to be highly correlated to professions and jobs (Kolb, 1984; Smith & Kolb, 1986). For example, Engineers are highly correlated to the LSI type "Converger", Journalists to "Accomodator", Police Officers to "Assimilator" etc. We propose that an additional use of LSI scores may be to assist the designer in identifying the dominant user LSI of the target intended audience (if one exists). By matching this information to the Kolb learning style descriptions, the type's preferred learning approaches may provide additional creative clues to prototype implementation with perhaps better learning and user enjoyment. However, we do not intend to test user satisfaction, or likeability scores within the scope of this research and are not in a position to make any conclusive statements on the utility of this potential use of group LSI scores.

TC Descriptor (Task Characteristics)

The type and complexity of the particular task to be learned will influence the selection of an appropriate learning strategy. As a result, we propose the use of a Task Characteristics Descriptor as a representation of the task type as well as the task complexity. This will be determined by matching the task type first to a type taxonomy based on the Jonassen (1988) compilation (Jonassen, 1988; Dansereau, 1978; Rumelhart & Ortony, 1977; Norman et al 1976) of information processing strategies (see Figure 9), as well as Farr (1987).

The task complexity will be quantified on the three characteristics described by Payne (1982), and this will complete the task profile contained in the Task Descriptor. It has been suggested that task complexity can be determined by evaluating the task on a number of dimensions such as number of alternatives, number of dimensions of information used to define an alternative, and the amount of time available to complete the task (from Payne, 1982).

Other, more complex task mapping and decomposition techniques have been proposed and could have been used. Other attempts at task analysis and mapping include Command Language Grammar (CLG) (Moran, 1981), Action Grammars (Reisner, 1981), Soft Systems Methodology (Checkland, 1981 & Browne, 1990) and Task-Action Grammars (Payne, 1984). However, these techniques have been criticized for being overly-complex for design application and easy specification, and some are not suitable for non-command language based designs (Browne, et al 1990 p. 107 & p. 196). Indeed, it would appear that the field of task complexity decomposition is still very much evolving, especially for higher order tasks which involve cognitive complexity. For this reason, we will at this time recommend that designers utilize the Payne framework for low-complexity procedural tasks, and that only tasks of similar complexity be compared using the evaluation metrics. As task complexity decomposition techniques become better defined and easier to use, then we can contemplate using the proposed framework in extended complexity situations.

Identifying the TC Descriptor has two intended results. The first result, obtained from matching the task type to the Jonassen taxonomy, is intended to aid the designer in selecting a learning strategy for use in the interface design. It is far from totally prescriptive, but is simply a way to enlarge the number of design options by suggesting a strategy that appears appropriate for the task. It is also far from being a panacea, as many tasks will map to multiple strategies. We simply suggest that its use is a modest contribution to the designer's tool kit in helping to approach an interface design where the usability dimension of learnability has been given a high priority as a desirable objective, or indeed a specific target (such as 80% of the target population must learn the task in 2 hours or less).

The second intended result of the TC descriptor is an approximate identification of task complexity. Task complexity is difficult to calculate with any precision. Therefore, the Task Characteristics descriptor is only useful as a guidepost for initial design implementation. As well, it can be used in an attempt to check that if two or more interface implementations are being compared that the tasks or subtasks being tested are either identical, or of very similar task type and complexity. Until more practical and implementable techniques are developed for measuring task complexity in practical design situations, then as a guiding principle designers should attempt to hold complexity relatively <u>constant</u> when comparing interface performance. This is consistent with the conclusions of Browne (1990 p. 196) who states "much better techniques are required...task analysis techniques are inadequate and none has yet gained wide acceptance. Analysts have many techniques to choose from. The choice is probably best made on the basis of the attributes of the application". We will also treat the Task Characteristics Descriptor as a control mechanism. Designers must ensure that different interface implementations under comparative test must have roughly equivalent task type and complexity within the application context if meaningful comparisons of other performance test measures are to have any validity.

<u>LS Variable Set (Learning Strategies)</u>

The learning strategy variable represents the most appropriate or promising set of learning strategies for the interface task, based on the task type and complexity delineated in the Task Characteristics (TC) descriptor, as discussed above. The initial set of suggested design strategies is extracted by matching the task type (identified in the TC Descriptor) to the most appropriate learning strategy (strategies) described in the appropriate subsection of the Jonassen (1988) taxonomy. In an interface design environment, this Learning Strategy Set is operationalized by an actual working interface implementation(s). As such, in a later evaluation and test environment, it represents the test treatment(s) and is therefore the independent variable of any experimental test suite.

OL Variable Set (Original Learning)

As discussed in section 3.2.1, it is most important to ensure that various subjects have learned a task to the same mastery criterion if meaningful comparisons of task retention and learning decay are to be performed at a later time. Also Farr (1987) (see Table 5) has determined that the degree of original task learning is the most important determinant of knowledge and skill retention. For these reasons, the degree of original learning achieved is an important quantitative metric. This is intended to be measured in an automated fashion while a human subject is actually learning to use the computer interface.

As discussed in section 3.1.5, most existing measures of learnability utilize time for task completion as a measure of learnability. For consistency, we will also define the variable OLtime (Original Learning Time) as the measure of how much elapsed time the task requires for the user to reach completion. However, we will also include in this variable set a benchmark criteria in order to satisfy the previously described (see section 2.1) importance of identifying the level of task proficiency exhibited (cognitive, associative or autonomous--recall Anderson, 1983; Fitts & Posner, 1966).

We will determine if a benchmark has been reached by measuring the number of errors exhibited by the subjects on their paths to achieving task completion. We therefore define <u>OLerrors</u> (Original Learning Errors) as a variable which measures the number of errors which occur during the user interface original learning session. This variable is again consistent with many learnability measurement definitions (Whiteside, 1988, also see section 3.1.5). The designer determines the maximum number of user errors to be exhibited for performance to have passed the defined benchmark.

The threshold <u>OLlevel</u> will represent the task proficiency benchmark level set for the task to be evaluated and tested by the designer (one of cognitive, associative or autonomous) and the variable <u>OLpass</u> will indicate if the benchmark was indeed achieved (exhibited fewer errors to completion than the benchmark threshold set by the designer).

At this time there is no generally accepted or well defined theory of error types and their effects. A desirable extension to the framework would be the later inclusion of error type definitions. However, until such a time as a theory becomes available, the recording of simple error events will be used in the framework, consistent with other usability practice.

Finally, we define the variable <u>OLiterations</u> as the total number of task completion iterations a user exhibits before the <u>OLlevel</u> threshold is met. This is a potentially important variable as discussed earlier (section 2.1, power law of practice, overlearning etc) but is not often discussed in other usability measures of learnability. As an example, this variable is an attempt to differentiate between two different users both spending ten minutes on a task before the task performance threshold is met. However, one user has actually repeated the execution of the task (or subtask) many times (with perhaps a few errors per trial), while the second has slowly completed the task once but with no errors.

The <u>OL Variable Set</u> then consists of the variables <u>OLtime</u>, <u>OLerrors</u>, <u>OLpass</u>, <u>OLiterations</u> and the threshold <u>OLlevel</u>. The <u>OLtime</u> variable provides an indication of the total amount of time to reach or complete a defined task benchmark (<u>OLlevel</u>), which will be useful in making relative judgments on the time efficiencies of interface prototypes with respect to initial task learning. The <u>OLpass</u> variable has a binary result; either the subject completed the task with fewer errors as defined in the task threshold OLpass, or they failed. These variables are intended to be automated system-measured dependent variables to be used for interface comparison purposes.

4.2.2 Learning Retrieval Domain

We define the learning retrieval domain as the condi-

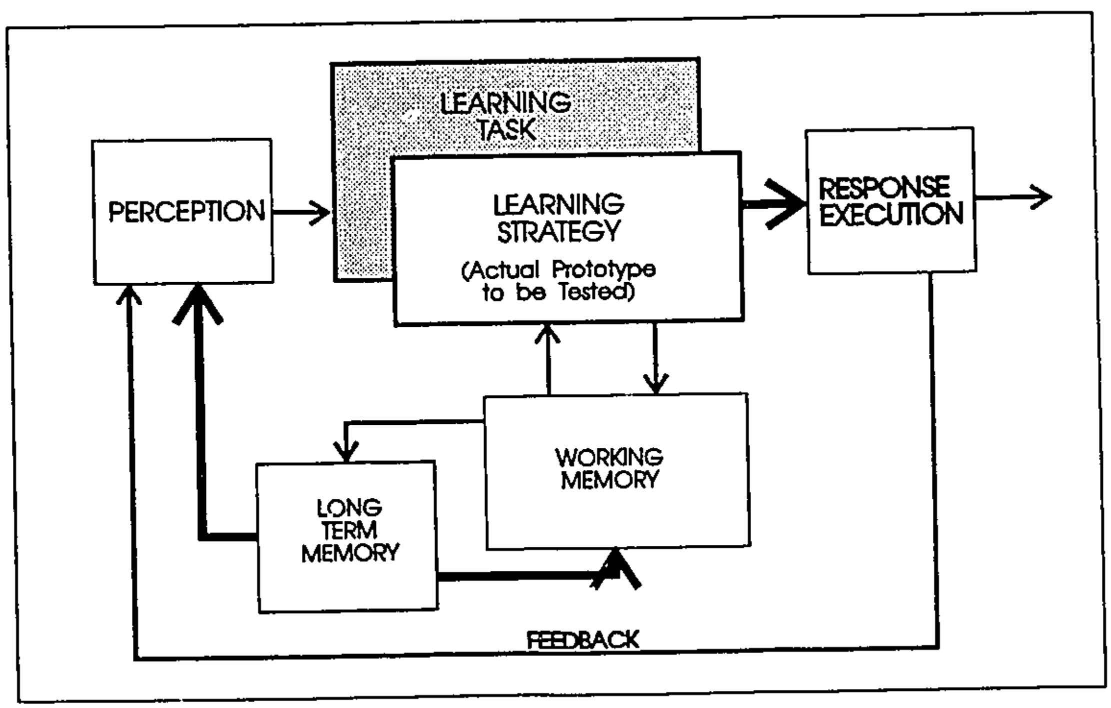


Figure 13
Learning Retrieval Domain

tions present when the user is attempting to apply or utilize the previously learned knowledge or skill at some time subsequent to the initial learning. Figure 13 represents graphically the functional representations we are attempting to improve.

By utilizing the most appropriate initial learning strategy, we wish to decrease the amount of decay over time of that learning. This is represented graphically by the increased flow (non-literal) from the long term memory to perception and working memory functional representations. We now overlay on this learning retrieval domain the variables and evaluation metrics as shown in Figure 14. First, we define the variables illustrated in Figure 14 and listed in the second part of Table 6.

RL Variable Set (Retained Learning)

In order to determine the effectiveness of the interface under test with regard to long-term memory retention achieved, we require a measurement of the amount of task proficiency decay that has occurred since the subject first learned and demonstrated the task to a benchmark level of mastery. Recall from Farr (1987) (see table 5) that over the retention interval the amount of decay is sensitive to the task characteristics, the degree of original learning and the method of training. So far the framework has attempted to guide the interface designer to an appropriate interface

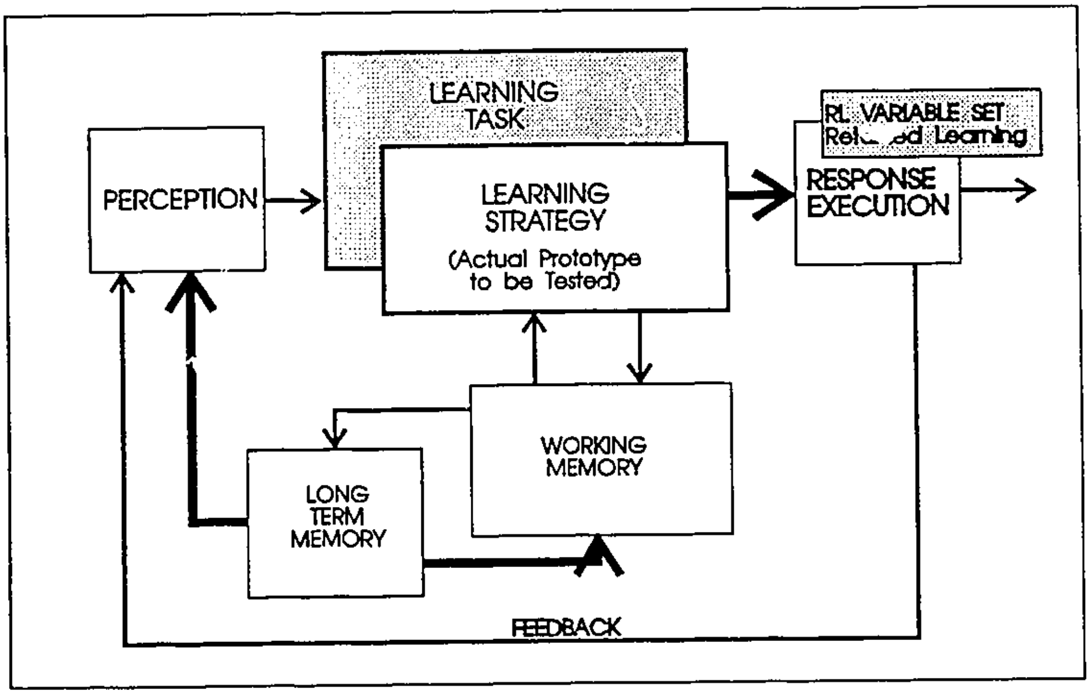


Figure 14
<u>Retrieval Domain Variables</u>

design(s) based on a mapping of the task characteristics to a set of appropriate learning strategies. The framework requires identifying the complexity of the task to be performed on the interface and holding it relatively constant across any set of multiple interfaces to be compared under test. The methodology also requires measuring the degree of original learning exhibited during the initial learning session and ensuring that task proficiency has been exhibited to a defined benchmark level. To this point, many of the measures obtained are similar to those obtained by other learnability evaluation methodologies. It is at this point that we introduce a new concept of the Learning Retrieval Domain, in that we wish to introduce measures of effective transfer to long term memory.

First we define the threshold RLinterval (Retained Learning Interval) as a designer defined metric of time lapse between initial task learning (Initial Learning Domain) and demonstration of recall performance (Learning Retrieval Domain). The designer sets the threshold to match the specific task and environmental context of the target user group. For example, in designing a system with intended users who are initially computer novices, and highly discretionary users, with no external control or pressure to use the system on a regular basis, a representative interval may be weeks or months. For an alternative target system intended to replace manual clerical processing functions by users with little discretionary power and little choice but to use the system, a representative interval may be one or two days. It is suggested that the designer set the variable to an appropriate level, perhaps based on interviews or surveys depending upon the size or focus of the application.

Consistent with the Original Learning Domain, we will define the variable <u>RLtime</u> (Retained Learning Time) as the measure of how much elapsed time the user requires to again reach task completion.

We will again determine if a benchmark has been reached by measuring the number of errors exhibited by the subjects on their path to achieving task completion under conditions of recall. We therefore define <u>RLerrors</u> (Retained Learning Errors) as a variable which measures the number of errors which occur during the user interface retrieval learning session.

The designer determines the maximum number of user errors to be exhibited for performance to have passed the defined benchmark.

The threshold <u>RLlevel</u> will represent the task proficiency benchmark level set for the task to be evaluated and tested by the designer (one of cognitive, associative or autonomous) and the variable <u>RLpass</u> will indicate if the benchmark was indeed re-exhibited (exhibited fewer errors to completion than the benchmark threshold set by the designer).

Consistent with the Initial Learning Domain, we define the variable <u>RLiterations</u> as the total number of task completion iterations a user requires before the OLpass threshold is met.

The RL Variable Set then consists of the variables RLtime, RLerrors, RLpass, RLiterations and the thresholds RLlevel and RLinterval. The RLtime variable provides an indication of the total amount of time to re-reach or re-exhibit a previously achieved task benchmark (threshold RLlevel), which will be useful in making relative judgments on the time efficiencies of interface prototypes with respect to task re-learning. The RLpass variable has a binary result; either the subjects completed the task with fewer errors (RLerrors) as defined in the task threshold RLlevel, or they failed. The variables RLtime, RLpass, and RLiterations are intended to be automated systemmeasured dependent variables to be used for interface comparison purposes.

4.2.3 Learning Effectiveness Evaluation

The learning effectiveness evaluation function is an attempt to provide the designer with metrics for use in comparing different task prototype implementations or individual task implementations as part of an overall implementation. Learning Effectiveness scores are manipulations of the dependent variables defined above (measured during original learning and recall testing), and are formulated in an attempt to provide easily relatable summary metrics consistent with currently recommended usability engineering practice, for example usability attribute specifications (Whiteside et al, 1988). It should be pointed out that the metrics set proposed is far from exhaustive, but would appear to meet a number of the needs of the practitioner community as enumerated in chapter 3.

<u>OLeffectiveness Score (Original Learning Effectiveness)</u>

Given the OL variable set from a new interface treatment under test, it can be compared to an initial or reference implementation. It is based on the amount of time required as follows, expressed in a form which results in a ratio normalized by the reference standard:

$$\text{OLeffectiveness} = \frac{\text{OLtime(ref)} - [\text{OLtime(test)} - \text{OLtime(ref)}]}{\text{OLtime(ref)}}$$

$$= \frac{2*\text{OLtime(ref)} - \text{OLtime(test)}}{\text{OLtime(ref)}}$$

The score provides a central value of 1 if the two interface treatments are equivalent. For example, suppose an initial task implementation prototype test resulted in a mean OLtime = 1000 seconds, with OLlevel = autonomous. This is interpreted as that on average it took users a total elapsed time of 1000 seconds before a task proficiency level of autonomous was demonstrated. If a subsequent prototype implementation results in an OLtime = 850 seconds, OLlevel =

autonomous, then the effectiveness of the new implementation compared to the reference implementation would be (for an equivalent task level and task complexity held constant):

OLeffectiveness Score =

$$\frac{2*OLtime(ref) - OLtime(test)}{OLtime(ref)} = \frac{2*(1000\ sec.)-850\ sec.}{1000\ sec.} = 1.15$$

Note: A score of less than 1 indicates lower effectiveness than the reference and greater than 1 indicates a more effective implementation, with respect to time. Also note that a further use of the score could be to a design reference or objective. That is, the reference does not have to be an actual implementation of the interface, but rather a design objective or target such as "80% of users are to learn the task within 30 minutes", etc.

RLeffectiveness Scores (Retained Learning Effectiveness)

Similarly, given the RL variable set from a new interface treatment under test, it can be compared to an initial or reference implementation. It is also based on the amount of time as follows, expressed in a form which results in a ratio normalized by the reference standard:

$$RLeffectiveness = \frac{2*RLtime(ref) - RLtime(test)}{RLtime(ref)}$$

We add an additional measure of relative recall error performance effectiveness consistent with the above as follows:

$$\text{RLeffectiveness (errors)} = \frac{2 * \text{RLerrors(ref)} - \text{RLerrors(test)}}{\text{RLerrors(ref)}}$$

RLeffectiveness scores are calculated in a similar fashion to OLeffectiveness scores described above. A score of less than 1 indicates lower effectiveness than the reference and greater than 1 indicates a more effective implementation, with respect to either time or iterations. Again, the reference does not have to be an actual implementation of the interface, but rather a design objective or target, such as "80% of users are to re-learn the task within 30 minutes", etc.

<u>CLeffectiveness Score (Combined Learning Effectiveness)</u>

A combined comparator has the utility of evaluating the learning time required in both the original learning and the recall test. This score will enable the evaluator to make overall comparisons of interfaces that may require some additional time to learn than a reference treatment, but exhibit sufficiently improved recall performance that implementation may be justified. We define the CLeffectiveness score as simply the mean of the OL and RL effectiveness scores, with equal weight to OL and RL effectiveness.

$$\text{CLeffectiveness} = \frac{\text{OLeffectiveness} + \text{RLeffectiveness}}{2}$$

4.3 Summary of Framework

The framework proposed is consistent with current usability design practice, and utilizes similar measures of learning speed and error performance that have been previously proposed. However, it also introduces the concept of a Learning Retrieval Domain to explore learnability evaluation utilizing retrieval dimensions.

The framework has proposed the introduction of a number of straightforward metric scores for use in interpreting test results obtained when applying the methodology. The scores are intended to assist designers in making interface design comparisons and facilitating analysis of potential implementation trade-offs.

The framework also suggests a step to enlarge a designer's scope of application by suggesting potentially useful learning strategies in applications where effective learning and recall are deemed to be of importance within a total usability design specification.

CHAPTER 5

Procedure for Framework Implementation

5.1 Implementation Steps

A Tool for Measuring Computer Interface Learning (ToMCIL) has been implemented to facilitate using the proposed framework in a development environment. It provides the system interface designer with a mechanism for implementing an automated software testing procedure that will measure the variables and manipulate the functions described in the previous framework and methodology.

ToMCIL was designed to provide a designer with a software tool to implement automated testing of the interface prototype both at the initial learning and subsequent retesting stages. The tool will provide the designer with the ability to define and codify error events and set the framework thresholds OLtypes and task proficiency levels. The tool will output the quantitative metrics as described in the framework to allow determination of the relevant scores, in order that a meaningful comparison of task implementations can be made.

This chapter will enumerate and describe the steps and procedure for implementing the proposed framework in a design and evaluation situation.

5.2 Step 1: Learning Strategy Selection

In the first step, a designer will be attempting to identify an appropriate learning strategy or strategies for the task to be implemented on a computer interface. A designer will already have completed a requirements analysis of the particular tasks to be completed using techniques outside the domain of this framework. The description of the task or subtask that results from this analysis is then manually matched to the Jonassen taxonomy in order to codify the task type (see figure 9 and table 4 for a listing of strategies available for information processing tasks). It should be noted that this will not involve a simple or prescriptive procedure, but rather it is intended to guide and enlarge the designer's creative horizons. The learning strategy or strategies selected form part of the Task Characteristics Descriptor.

In addition, the designer utilizes one of the available and appropriate methodologies for identifying task complexity, and attempts to determine the complexity level of the task. For initial purposes of demonstration, we are using Paynes' Task Action Grammars (Payne, 1984). This task complexity result in combination with the above Task Strategy forms the complete Task Characteristics Descriptor.

As noted earlier, currently available methodologies to measure task complexity are difficult to use in a practical design environment. However, at this point we propose to include use of such a methodology but with the limitation of applying the framework initially to only simple, structured tasks. The intent is not to use the task complexity measure with any rigour, but rather to use the technique as a means to keep complexity relatively constant across any interface treatments being compared later.

This taxonomy and task complexity measurement technique is not original to this proposed framework. We are simply suggesting that the determination of task type and complexity and selection of a learning strategy be "institutionalized" in the design process when learnability is deemed to be an important result.

5.3 Step 2: Interface Design

The intent of the methodology is that once a task strategy is recommended, a prototype design of the interface for that task would be implemented. The methodology makes no prescription of how to do the implementation or to complete step 2, but recommends it be based on existing state-of-the-art implementation methodologies (such as Shneiderman, 1986; Norman et al 1986; Whiteside et al 1988 etc.) consistent with designing usable interfaces. This typically involves principles of rapid prototyping and iterative design (Gould & Lewis, 1985).

5.4 Step 3: Application of ToMCIL

This step involves application of the Tool for Measuring Computer Interface Learning (ToMCIL). For demonstration purposes, ToMCIL was implemented using IBM Linkway software running on an IBM PC to provide an automated testing tool. Linkway was selected for demonstration purposes because it is flexible, inexpensive, and allows rapid prototyping of direct manipulation, icon-based user interfaces. As such, a designer can utilize IBM Linkway (or other prototyping or UIMS software) to implement a prototype after completing information and task analysis and selecting an appropriate learning strategy. ToMCIL procedures written in Linkway (or subsequent similar procedures written in the appropriate prototyping language used) provide a designer with additional capabilities that will allow him or her to do the following activities subsequent to the completion of the first iteration of the prototype design completed in step 2:

1) Define Error Events - The designer can define the error events to be detected by the system (variables OLerrors & RLerrors). The designer can then use ToMCIL procedures to overlay on the prototype the various error events to be recorded, by assigning an error code to each possible error category.

2) Define Success Events - The designer can also define the event that determines if the user has achieved successful task (or subtask) completion, or any intermediate benchmark event. The designer then sets the OLlevel and RLlevel by determining the task proficiency level to be exhibited (cognitive, associative or autonomous) and the acceptable number of errors exhibited prior to task completion. This is used by the system to recognize when the user has completed the task (or subtask), and is used to determine the pass or fail condition of the variables OLpass and RLpass.

Once the designer has completed defining the error events and success events, and has assigned each of the possible user responses or inputs to the interface by overlaying the appropriate test element, ToMCIL is then ready to automatically record the variables of interest while the user is learning to use the interface, as well as when the user is attempting to retrieve the knowledge or skill previously learned. In all respects, the interface should behave to the user in a manner identical to an interface prototype running independently of ToMCIL.

ToMCIL provides a journal file which records each individual user event (keystroke, icon activation, mouse manipulation, error code, and other input/outputs) which can be used for later troubleshooting and problem analysis. A second journal file is produced in a similar manner during the test of the user learning retrieval session.

Before automated testing begins, the individual subjects are also administered a series of short questionnaires to determine their Individual Difference Variable Set (Age, Sex, Experience, Attitude, LSI type).

Testing is repeated at a subsequent time (an application appropriate time interval is used) to measure retention performance (RL variables). At the conclusion of step 3 testing of the subject population, the designer has a complete set of OL and RL variables available for use in evaluating the interface treatment(s) as well as the Individual Difference variables available for use as concomitant variables during the analysis.

Development of the ToMCIL test module is complete for use within an IBM Linkway software environment.This approach was used successfully for the experimental study described in chapter 6.

5.5 Step 4: Evaluation of Interface

Once the subject sample has completed interface testing using the above test module, the data are computed to provide the designer with the resulting scores of Original Learning, Retained Learning and Combined Learning Effectiveness as described in the framework.

The ToMCIL output data can then be analyzed using a commercially available statistical package. In particular, the analysis of variance across the interfaces tested can be performed to determine the significance of any differences observed.

Based on the results, the design team can make evaluations of the interface effectiveness. In addition, the journal of events can be analyzed step-by-step to identify any particular problem or interpretation occurring on a regular basis to enable refinement of the actual prototype implementation details.

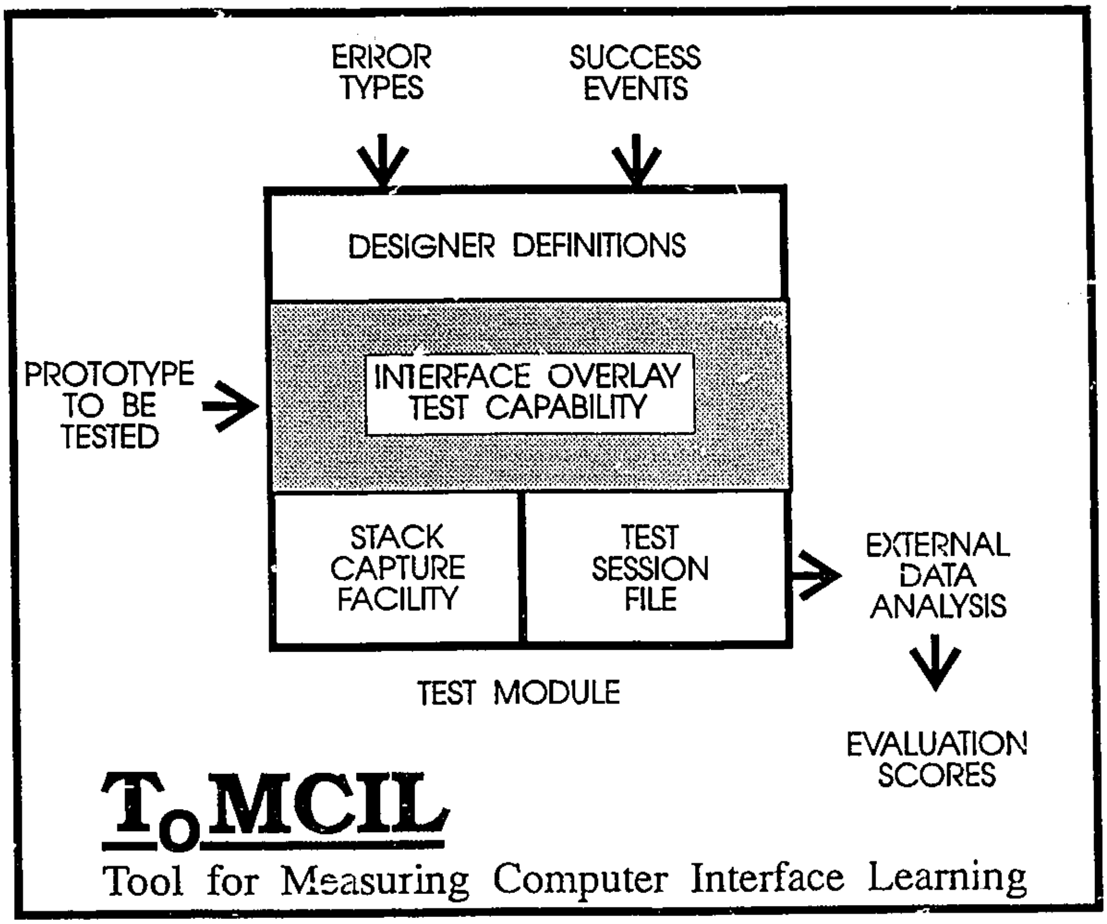


Figure 15
<u>ToMCIL MODULES</u>

5.6 Step 5: Re-iterations of Steps 2 to 4

Based on the results obtained from the above scores as well as other measures deemed important and consistent with the design implementation technique being utilized (e.g. Whiteside et al, 1988, -- user satisfaction, perceived user utility. etc.) the procedure is iterated until the design meets the objectives set.

5.7 Summary of Implementation

This chapter has proposed a number of steps for implementing the framework in a real design situation. Step 1 suggests a procedure for capturing the essentials of the task to be implemented and matching a descriptor to a taxonomy of potentially useful learning strategies and the appropriate cognitive processes and activities that may improve interface learning and recall. Step 2 suggests prototype implementation within a rapid prototyping/ iterative design methodology as proposed by others. Step 3 outlines a test methodology for obtaining appropriate learning variables, with step 4 evaluating the results of the test and guiding re-iterations of the process. The following chapters provide a detailed description of an application of the framework and additional detail on data analysis and interpretation of framework test results.

CHAPTER 6

An Application of the Framework

6.1 Objectives of the Application

The objective from section 1.1 addressed by this chapter is as follows:

Objective 2: To apply the framework and demonstrate its usefulness. In order to do this, a new interface design utilizing multiple alternative techniques will be implemented, based on principles relevant to improving human learning and memory. Actual working prototypes of the interfaces will be experimentally tested on human subjects using the tool in order to evaluate its effectiveness when used for comparing interface designs.

This chapter describes an experiment designed to demonstrate the application of the framework and test tool in a realistic situation. Previous chapters have deduced various measures and techniques that will provide an ability for designers to establish how much two or more interfaces differ, hence allowing potential cost benefit or judgmental decisions regarding implementation to be made. However, experimentation and statistical analysis is required to determine if the various variables measured differ in a significant way.

This experiment will therefore demonstrate one application of the proposed framework. If the interface design treatments have a systematic effect on the experimental dependent variables, then it is argued that this framework and test methodology has demonstrated relative usefulness and utility.

While this experiment is expected to add confidence to the use of the framework for design comparison of learning speed and retention, it should be noted that this is <u>an application of the framework on but a single subset of the framework</u> in a simulated design task. It is expected, however, that this exploratory demonstration of utility will allow subsequent trial and procedural refinement as a result of practitioner use and feedback.

6.2 Overview of Study

As we have discussed earlier, the framework and methodology allow the selection and testing of an appropriate learning strategy. The methodology does not, however, prescribe the best or any exact implementation. The designer must use other available design methodologies and his or her experience to build the interface based on the underlying strategy principles (see Shneiderman, 1986; Norman et al, 1986; Card, et al, 1983; Whiteside, et al, 1988 etc.). In order to demonstrate the methodology, a number of different interface prototype treatments will be designed and implemented. The framework and tool will then be used to test the various implementations, and evaluate the success of each of the implementations with respect to the appropriate learning scores as defined in the framework. The different interface designs are used to approximate the real world of interface design. As our earlier discussion of cognitive psychology outlined, there is not always a clear agreement on which of many possible approaches will provide the "best" results. In addition, we have mentioned that often the design performance will be heavily affected by not only the approach taken, (e.g. commands vs. menus) but also on the implementation details (Whiteside, et al, 1985). The designer gets an indication of a promising direction, or perhaps even conflicting direction from the theory. It is up to the designer to try various implementation approaches. The proposed framework is an attempt to help the designer make evaluative decisions which involve trade-offs of both theory and implementation.

The framework incorporates a number of measures that are equivalent or similar to many used in current usability practice, including learning speed and error performance. However, the application of the framework described in this chapter <u>will concentrate on demonstrating the variables used to compare retention performance</u> of the various interface designs, as it is in this area that the framework has the greatest potential of contributing to usability evaluation.

6.3 Determining the Design Treatment Alternatives - Step 1

In order to test the validity and practicality of the framework, an example interface was designed for an electronic office mail system. The task to be developed was the use of the mail system to receive, file and retrieve mail. Other tasks are present in a typical mail system, such as creating mail, editing mail etc. However, the proposed design concentrates on one major task manipulation. Other criteria were as follows:

1) The task was to be implemented on an icon-based system.
2) The target group were discretionary user-managers.
3) The target group were likely to range from very experienced to no experience.
4) Based on the management task style of the target group (Mintzberg, 1973) we assumed that learning execution was likely to be very fragmented (in the order of days to weeks between sessions).

From an analysis of the task and criteria matching the Jonassen Taxonomy, a potential learning strategy appeared to be an elaboration strategy built around imaging and analogy; imaging because the style was to be iconic, and analogy because we wanted to incorporate both inexperienced and experienced user preconceptions of the existing manual tasks to be automated on the system.

Difficulty was experienced in measuring the task complexity when attempting to use Task Action Grammars and alternatively Command Language Grammars, primarily due to attempting to use them for iconic-based systems (for which to be fair they were not designed). As an alternative, it was decided that the task could be held constant across different implementations if the following criteria were met (Payne, 1982):

1) The number of sub-tasks (or sub-goals) would remain constant.
2) The number of alternatives available in each sub-task choice set would remain relatively constant.
3) The number of dimensions of information used to define an alternative choice would remain relatively constant,
4) The amount of time to make decisions would be open-ended, and thus each implementation made no time pressure demands on subjects. The interface would simply wait as long as the subject required without using time-out prompts or queries.

At this time we are satisfied that this procedure will assist designers in a production environment as long as they are not attempting to compare radically different task types with large differences in complexity.

From a matching of the Task Characteristics Descriptor, the Jonassen taxonomy lists the cognitive processes involved as dual coding and encoding variability which maps to the following appropriate learning/ instructional activities (see section 3.3.1 and table 4):

1) Mental Images/ create drawings describing content.

2) Complete analogies or create own.

6.4 Designing and Implementing Interface Treatments - Step 2

The designer is now faced with the task of enumerating the various design alternatives available. Based on a review of elaborative techniques discussed earlier in chapter 2 , it is not clear which type of elaboration is most appropriate to this task, as the cognitive theories discuss the "generation effect", depth of processing, and imaginal-verbal dual processing (see sections 2.2.4 and 2.2.5). Applied to an icon-based, direct manipulation interface, there are several techniques available which may provide this elaboration. A typical implementation of an iconic interface provides a system-defined graphic icon for each of the high level functions available to the interface user. However, some research theorists suggest that increased learning may occur if the user chooses his/her own graphic for the function by selecting one from a predetermined list of graphics available from the system (Lansdale, 1988, also Lansdale, 1987). Indeed, this approach was implemented and measured by Lansdale, but not analyzed within the context of the "generation effect". Other studies indicate that some advantages may occur if users have the ability to "generate" verbal labels of the function that is meaningful to them, and have this label appear visually with the graphic (Madigan, 1983) (see section 2.2.4). However, the Madigan study was a cognitive psychology experiment and was not implemented as part of a computer interface design.

So far in step 1, we as designers have determined that elaboration is an appropriate learning strategy. As well, we have done further specific research (reported previously in chapter 2) and found that a number of techniques have been explored that may show promise in improving interface learning, namely:

1) Selection by the users of their own graphic icons.
2) Generation of a text label by each user, to appear in addition to the graphic icon.

As designers, we have so far identified two promising directions for implementing our new interface design. Existing widely adopted interface systems (such as Apple Macintosh or Microsoft Windows environments) typically provide the user with system defined graphical icons for functions either alone, or with an accompanying system defined text label (see Figure 16).

We can treat these as existing and available interface treatments. We are therefore interested in examining new interface treatments incorporating the two newly identified

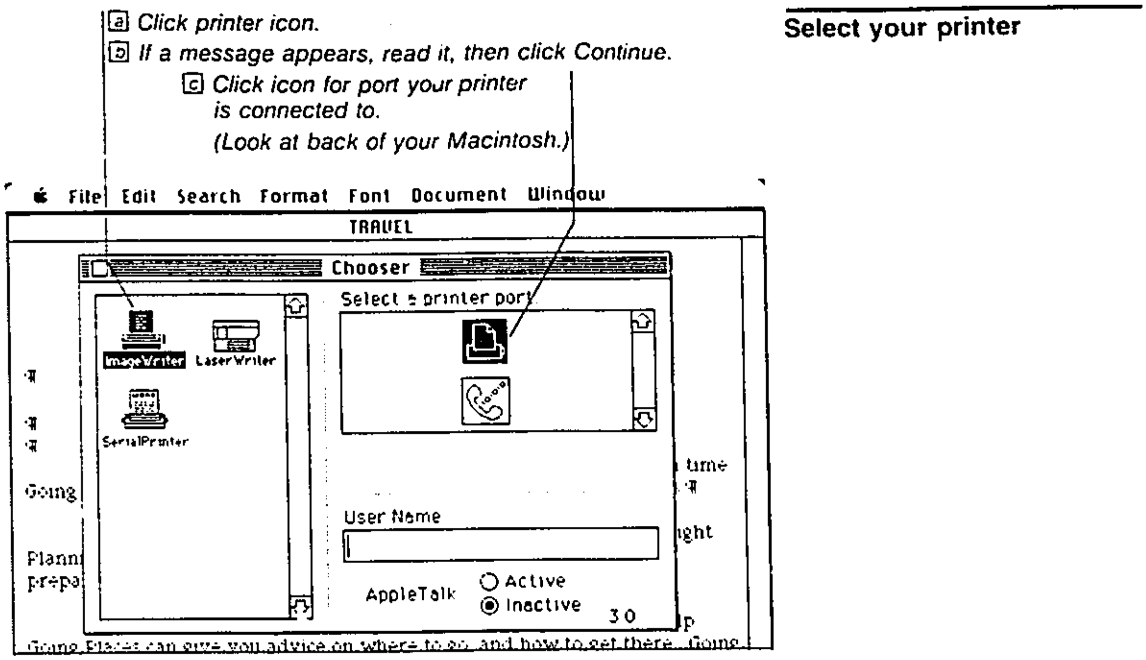


Figure 16
Typical Macintosh Icon Procedures

techniques. While the possible implementation alternatives are limited only by the design team's imagination, and the dissertation framework makes no prescription on how to create alternatives, we have identified five possible interface implementations that would appear to allow comparison of existing and new treatments as follows:

<u>Existing treatments</u>

T1) System-provided icons only.

T2) System-provided icons <u>plus</u> system-provided text labels.

<u>New treatments</u>

T3) User-selected icons only.

T4) User-selected icons <u>plus</u> user-generated text labels.

These are all possible and reasonable implementations of a system. However, a system designer is presented with a number of possible tradeoffs. If a user selects his/ her own graphic for an icon function, a universality constraint may be broken. In other words, if users learn to use a system with their own graphic set, they may not be able to use the interface at another location or terminal unless the system can dynamically instantiate their personal "selection set" at any time of use (e.g. dynamically adaptive interface- Benyon & Murray, 1988; Browne et al 1990). This may be a very complex and costly capability to provide. The designer may wish to determine the gain in learnability scores between using system- or user-defined graphics for the functions in order to make a meaningful decision on final design. As a result, this analysis results in the desirability of testing an additional "hybrid" of old and new, namely:

Hybrid treatment

T5) System-provided icons plus user-generated text labels.

This interface would provide the practicality of a universal graphical icon for a function which remains constant across all installations of the interface software. The interface treatments are summarized in Figure 17.

Other tradeoffs are of interest. If users define their own graphics, or text labels, these activities may add additional time to the initial learning session. Will this result in a system which is easier to learn, or could the same

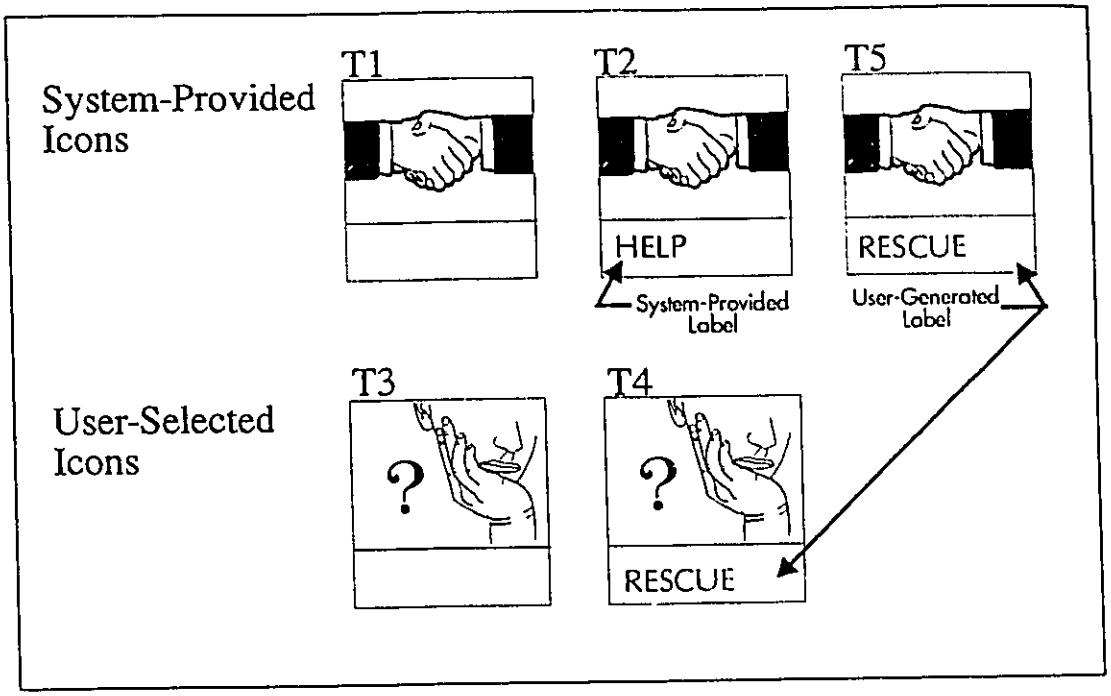


Figure 17
Interface Treatments

amount of time be spent on simple practice iterations to achieve the same level of original learning by simple rote techniques? Also, what is the size of improvement relative to any additional investment of time expended during learning (Landauer, 1987). In other words, there may be significant differences in interface performance, but are the differences substantial and of practical value (milliseconds relative to minutes). It is expected that these tradeoffs can be analyzed using the learning effectiveness scores from the proposed framework.

6.5 Pre-Test Expectations

Based on the cognitive theories discussed in the literature review sections 2.3 and 2.4, we can make the following proposition:

The treatments consistent with a true generation effect--Treatment T4 (user-selected icon plus user-generated text label) and T5 (system-provided icon plus user-generated text label) should provide better retention than the two existing treatments -- Treatment T1 (system-provided icon only) and Treatment T2 (system-provided icon plus system-provided text label).

We reach this proposition based on the premise that T1 and T2 should provide the poorest performance on learning retention of all of the treatments, because no elaboration or user generation is occurring. Also based on the literature, the generation effect results in better recall performance, so we expect that T4 and T5 should be better.

Therefore, using retention error performance as our framework measure of recall precision, the proposition can be stated as follows:

P_1: The RLerror performance of treatments which incorporate user-generation will be better than treatments which do not.

However, based on the literature we are not able to predict the performance of T3, or the relative performance of T3, T4 and T5. This is because the author has not found any evidence supporting or disproving whether or not the process of both generation and selection results in any additive effects on performance. In addition, implementation effectiveness may determine the degree to which the icon representations provided to the user by the system or available to the user for selection accurately represent the users' internalization or understanding of the function. If the icon brings with it a meaning to a user that is inconsistent with the actual activity (recall discussion of Bott, 1979) some negative interference may result.

It is also not at all clear which of the new or hybrid treatments would provide the best learning effectiveness and retention scores, because it is not predictable with any precision from the theory examined that user-generated verbal labels added to system-provided picture graphics provides better learning and memory than user-generated (selected) picture graphics alone, or vice versa. This approximates a real situation where a designer can use differing techniques for implementation which may not be supported yet by scientific theory or research.

In section 6.1 above, we stated our intention to determine whether or not the interface design treatments have a systematic effect on the experimental dependent variables in order to demonstrate relative usefulness and utility for the framework. We therefore propose to statistically test for rejection of the null hypothesis:

$$H_0: \quad M_{T1} = M_{T2} = M_{T3} = M_{T4} = M_{T5}$$

where M_{Ti} is the mean RLerror performance of the treatment. If the null hypothesis is rejected, we will use additional post hoc statistical analysis to determine the treatment performance ranking.

It is expected that the tool will provide an indication of performance for testing H_0 <u>over the subject population only, given the effectiveness of the actual interface implementation details.</u> No attempt will be made to generalize the results to prove or disprove the cognitive theories. However, in a practical application, a designer is seeking the best implementation for the subject population for which the system is being designed. The study is expected to provide a relative ranking of the interface implementations for this subject population, and to demonstrate the framework, methodology and tool being proposed.

In addition to H_0, it is not clear which of the treatments will provide the best learning effectiveness. As discussed earlier, a more elaborative learning strategy may require more elapsed time during original learning, which may or may not be more time effective than a simple interface with more iterations (power law of practice). The study results of the learning effectiveness scores are expected to provide the designer with information useful in evaluating the various tradeoffs available.

6.6 Experimental Design and use of ToMCIL Test Tool - Step 3

The five interface alternatives were implemented as prototype interface designs using IBM Linkway software. The ToMCIL tool was then used to define the appropriate error events and success events to be recorded for each interface treatment as provided in the ToMCIL test module (as defined in section 5.2.2 above).

The interface alternatives were then tested on human subjects using a sample of 80 subjects with equal numbers of subjects assigned to each interface treatment (16 per alternative).

In order to eliminate the introduction of any systematic bias due to individual differences (i.e. general ability, visual ability etc.), ideally each subject would be presented with all five treatments of the interface, so that individual difference effects would be eliminated. This would be a so called "within subjects" experimental design. However, as in most educational experiments, this would result in a practice effect (residual learning) from one interface implementation to the next. Due to the power law of practice theory (refer to section 2.1), this practice effect may be large enough to swamp the comparison of each implementation. As a result, the experimental design is a "between subjects" design, with each subject receiving only one of the treatments. In order to mitigate for other individual differences, we measured individual subject's LSI scores, attitude, experience, age, and sex. These were treated as concomitant, or moderating variables measured only to control for individual differences and analysis of covariance techniques were used. By ensuring that users exhibit a common level of task/skill proficiency (see chapter 3.2) we minimized the effects of other individual ability differences. In addition, analysis of covariance techniques were used in order to eliminate some of the variability effects of individual differences.

The experiment, therefore, is a random, balanced between-subjects design as follows:

Experiment

Treatment: T1, T2, T3, T4, T5

Independent Variable:

1) Treatment (T1, T2, T3, T4, or T5)

Dependent Variables [1] :

1) Retained Learning # errors (framework variable RLerrors)
2) Retained Learning success (framework variable RLpass)

Concomitant Variables [2] :

1) LSI (framework variable ID-Learning Style)
2) Attitude (framework variable ID-Attitude)
3) OLdifmeasure (derived from framework variable OLtime)

[1] Note that the complete set of framework variables proposed in chapter 4 and listed in chapter 6 will be measured in the experiment. However, the hypothesis test is based on RLerror performance and thus only recall error performance variables form the dependent variables in the experiment.

[2] As a result of a pilot study using 17 subjects, of the individual difference variables proposed by the framework, only LSI and attitude appeared to have any covariate explanatory power, and these were used as concomitant variables in the analysis of the experiment. In addition, the pilot study identified a potentially useful use for OLtime as an individual difference variable. In the pilot study OLtime was highly correlated to the dependent variable OLerrors and offered the promise of being a useful covariate. The interpretation is that the amount of time a subject takes to reach proficiency was acting as a surrogate measure for individual ability in manipulating an interface. However, not suprisingly OLtime was also highly correlated to the treatment (treatments differing in mean learning time performance). This makes interpretation of the ANCOVA result difficult and suspect, although the usual difficulty is that high correlation to the independent variable may account for so much of the variability that the effect on the dependent variable may be masked (see Neter, et al, 1985 p. 847, also Wesolowsky 1976 p. 216). In order to remove the effect of the treatment on the OLtime, a new variable was derived. We defined and calculated an OLdifmeasure which is the individual OLtime less the mean OLtime measured for the individual treatment assigned. That is, the new variable measures the difference (positive or negative) in learning time for each individual from the group mean learning time for the assigned treatment. Thus, the variable represents the relative time performance of the individual but normalized to the group performance. The new

variable is still strongly correlated to the dependent variable RLerrors, but is no longer impacted by the treatment, making ANCOVA interpretation more straightforward. OLdifmeasure is similar to pretest scores commonly used in learning experiments where general IQ, reading ability etc. is used to remove from the experiment the test performance variance due to individual ability and allow treatment results to show through (Kerlinger, 1973 p. 370).

6.6.1 Experimental Procedure

The study implemented a prototype electronic mail system interface usin; IBM Linkway software running on an IBM PS/2 model 70 computer equipped with a mouse. A total of 85 subjects participated in the study on a volunteer basis. Subjects were undergraduate business students at the University of Toronto, MBA students at McMaster University and a variety of managerial and specialist employees of a not-for-profit institute in suburban Hamilton, Ontario. A total of 4 subjects were unable to complete the second phase of the study, and 1 subject data set was corrupted due to a system malfunction (power failure). In the laboratory, subjects were randomly assigned to one of the following treatments:

T1) System-provided icons only.
T2) System-provided icons plus system-provided text labels.
T3) User-selected icons only.
T4) User-selected icons plus user-generated text labels.
T5) System-provided icons plus user-generated text labels.

Subjects were administered preliminary questionnaires to determine their previous computer experience, age, sex, as well as their attitude towards computers (see appendix 1). In addition, subjects were administered the Learning Style Inventory (LSI).

The subjects then learned to use the computer interface by following instructions provided step by step on the screen. Once the users completed a system-provided tutorial of how the mail system worked, they were then asked to perform the following task:

1) Turn on the mail system.
2) Retrieve a letter previously received called Salestax.
3) Read the letter and find the sales tax rate listed within.
4) Re-file the letter.
5) Exit the mail system.
6) Type the sales tax rate into the computer when asked.

The task was deemed to be successfully completed (OLpass = yes) at a benchmark level once the subject was able to type the sales tax rate into the computer. This was defined as reaching a <u>cognitive level of acquisition</u> of the task skill (OLlevel set to = cognitive). Because of the nature of the task the number of iterations of task performance never exceeded one (OLiterations = 1). Appendix 4 is a reproduction of the Treatment T5 computer interface screens presented to the subject in the order they would be presented if no errors were made.

Based on the target group characteristics described above in step 1, 3 weeks was deemed an appropriate retest interval

(RLthreshold = 3 weeks) to reflect typical user delay in subsequent attempts at such a system in the early stages of learning and adoption. Each subject returned to the same lab to be re-tested exactly three weeks later, at the same time of day as the original session in an effort to minimize external contextual effects. The subject was then asked to perform a task similar to the original task, but without system prompting of the correct procedure to follow. The system only provided simple (non-contextual) feedback if an error occurred, and asked the subject to please try again. The task was deemed to have been successfully completed once the subject was able to type the new sales tax rate into the computer. This was defined as re-exhibiting a <u>cognitive level of skill retrieval</u> (RLlevel = cognitive). As in the initial learning session, subjects were only allowed one iteration to complete the task (RLiterations = 1). The remaining learning retrieval domain variables were measured and recorded (RLtime, RLerrors and RLpass).

The framework variables were measured using the ToMCIL test tool, and analyzed statistically using Minitab and SPSS statistical software.

Because of the nature of the testing of individual learning and memory, it is difficult to use standard methods of test-retest to provide measures of reliability of the test procedure (Christensen 1988). A preliminary pilot study was completed using protocol analysis procedures to provide a means of determining reliability as well as validity of the proposed test methods and validating our attempt to hold the task complexity constant. This preliminary study was also used to help estimate the sample size requirements of the main study in order to reach conclusive results. The computer also recorded on disk all keystroke & mouse events of the user session with time stamps for later matching with the protocol analysis audio tape. This formed the total protocol that was used for later analysis. The results of this analysis provided clues to the mental processes being used, and aided in refining the experimental tasks to eliminate common problems that interfered with the measurement of the variables of interest.

As an additional indicator of test reliability, the pilot study was compared to an equivalent subset of data on new subjects drawn from the main experiment. The comparison provided the consistent result that the treatment effect was significant. The analysis of the experimental test and comparison of interface treatments (Step 4) is detailed in the next chapter.

CHAPTER 7

Results & Analysis

7.0 Data Analysis of Main Study

Using the ToMCIL tool, the data collected consisted of the complete keystroke record and framework variables including the number of errors made by the subjects during the task learning session, the number of errors made during the retrieval session three weeks later, and the time taken during both events (see Appendix 2 for an example of a sample user session file). During the retrieval session, we defined the task to be successfully completed (variable RLpass = yes) if the subject completed the task with 2 or fewer errors (OLerrors = 2 or fewer and OLlevel = cognitive if task completed). Above this number, it was judged to be possible to complete the task simply by guessing and randomly trying anything until the task was completed. Because the task was procedural, it was felt that two or fewer errors would indicate that the task procedure was indeed recalled. However, as this reduction rule was based on subjective designer observation, the data were analyzed both ways, first with the raw number of errors (RLerrors), and using the two or fewer error reduction rule (RLpass) to investigate the usefulness of the rule.

7.1 General Data Overview

Table 7 is a review of the framework variables and their use in the experiment and table 8 is a summary of the measured variables. Note that throughout the analysis a significance level of $p < .05$ is used.

Measures of original learning are similar to a number of existing measures of learnability. However, a simple analysis consisting only of comparing OLtime and OLerrors results in conflicting and inconclusive results (Table 8). There is no significant difference between T1, T2 or T3 with respect to OLtime. However, both T4 and T5 require more time to learn than T1, and the difference is significant. In addition, there is no significant difference in OLerror performance. This may lead us to the conclusion that T1 is "better" than T4 and T5. We now turn to the experimental results using the framework, and more specifically recall measures, to provide a richer evaluation of the interface treatments.

Table 7
Use of Framework Variables

INITIAL LEARNING DOMAIN VARIABLES

Individual Differences Variable Set (ID Variables)

- **Attitude- used as covariate in experiment**
- Experience- measured but not used
- Age- measured but not used
- Sex- measured but not used
- **Learning Style (LSI)- used as covariate in exp.**

Task Characteristics Descriptor (TC Descriptor)

- Task Type- held constant across treatments
- Task Complexity- held constant across treatments

Learning Style Variable Set (LS Variables)

- Appropriate Learning Style(s)- influenced design

Original Learning Variable Set (OL Variables)

- OLtime- measured
- **OLerrors- used to derive OLdifmeasure covariate**
- OLlevel- set = cognitive
- OLpass = yes if task completed
- OLiterations = 1 for all subjects due to task type

LEARNING RETRIEVAL DOMAIN VARIABLES

Retained Learning Variable Set (RL Variables)

- RLinterval:set = 3 weeks
- RLtime: measured
- **RLerrors: used as dependent variable**
- RLlevel: set = cognitive
- **RLpass: used as dependent variable**
- RLiterations: =1 for all subjects due to task type

Table 8
Summary of Experimental Data Raw Means
(No Covariates Included)

	REFERENCE TREATMENTS		NEW TREATMENTS		
VARIABLE	T1	T2	T3	T4	T5
OLtime (sec)	316.6	317.1	321.8	462.1**	401.3*
OLerrors	1.25	2.19	2.06	1.75	1.25
RLtime (sec)	68.3	88.9	62.4	80.1	56.6
RLerrors	2.63	3.88	2.75	2.63	1.56***
RLpass(1=pass)	0.375	0.312	0.375	0.563	0.688

Note: Using Tukey Method of multiple comparisons of the Means:

** = significantly different than the reference T1 @ p < .05

* = marginally different than the reference T1 @ p < .10

*** = significantly different than the reference T2 @ p < .05
and marginally different than the ref. T1 @ p < .10

7.2 Experimental Results

The experiment was defined as shown in figure 18.

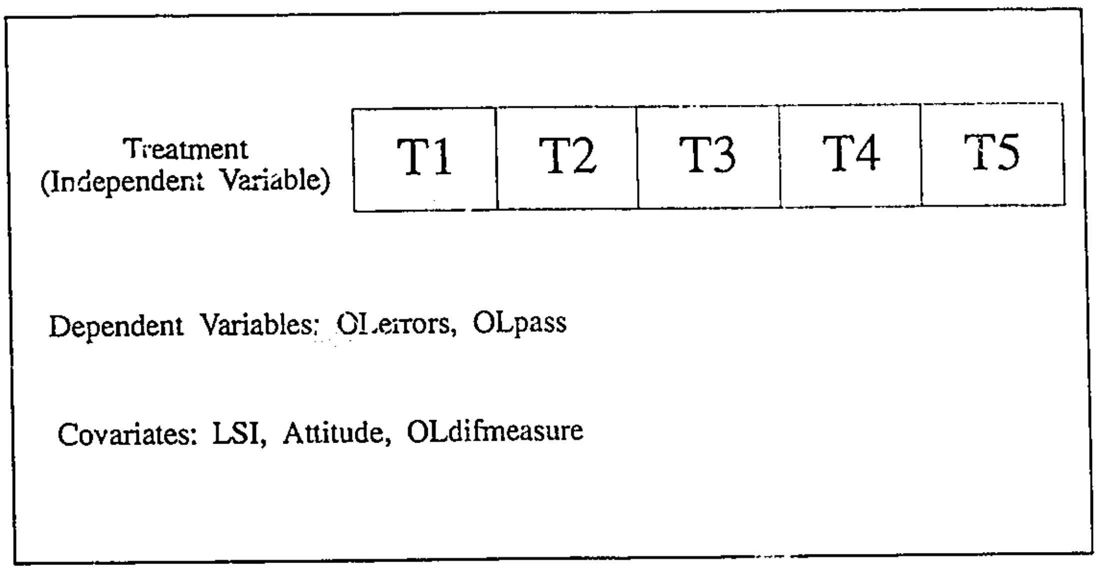


Figure 18
Experimental Design

We now present the results of the experiment using analysis of covariance (ANCOVA) as follows (dependent variable = RLerrors, LSI, Attitude and OLdifmeasure as covariates):

Factor	Levels	Values				
Treatmnt	5	1	2	3	4	5

Analysis of Covariance for RLerrors

Source	DF	ADJ SS	MS	F	P
Covariates	3	67.635	22.545	7.57	0.000
Treatmnt	4	43.973	10.993	3.69	0.009
Error	72	214.552	2.980		
Total	79	325.188			

ADJUSTED MEANS

Treatmnt	N	RLerrors
1	16	2.7495
2	16	3.8310
3	16	2.8235
4	16	2.5335
5	16	1.4999

As can be seen, the results allow us to reject H_0 ($F(4,72) = 3.69$, $p < .05$).

As we have discussed throughout the dissertation, individual differences in ability or other factors have an effect on subject performance. In the pilot study we found that sex, age and experience on other systems did not appear to have significant explanatory power on performance. This is consistent with the literature. However, attitude and LSI (Learning Style) appeared to be covariates, although the size of sample was small and unbalanced with respect to LSI and Attitude-- that is we had very few subjects with the LSI type "diverger" or "accomodator", and no subjects with a strongly negative attitude towards computers. It was speculated that these results were perhaps artifacts due to the high degree of unbalance. The main study had a much better balance of LSI types, but there were still no subjects with strongly negative attitudes towards computers (perhaps due to the population drawn from business employees and business university students). The main study show the covariate set was significant, but LSI did not provide much covariate or predictive power and was individually not significant. Attitude is also not significant in this analysis. However, OLdifmeasure was significant at the $p < .01$ level. This has provided further evidence that the variable OLdifmeasure, which is a normalized measure of initial learning time (normalized OLtime), functions as a pretest of individual ability in manipulating the computer interface.

The treatment effect is also strong <u>even without inclusion of the covariates</u>. Using simple analysis of variance (ANOVA) techniques (independent variable = treatment, dependent variable = # of errors, <u>no covariates considered</u>), the results show the variable treatment to be significant ($F= 2.86$, $p = .029$). These results are reproduced in Appendix 3.

An analysis of the RLpass variable (pass/ fail 2 errors or less rule including covariates) provided similar results, but the results are not as conclusive as the measure is only marginally significant ($F=2.19$, $p=0.078$). Indeed, this result was consistent throughout the analysis and the conclusion can be drawn that the RLpass variable is not as consistent an indicator of recall performance as OLerrors.

7.3 Summary of Experimental Data Analysis

Consistent with the experiment, the only treatments that appear to improve error performance substantially have the common thread that users generate their own text label.

From the analysis including covariates we found that T4 and T5 provided mean error performance of 2.5 and 1.5 errors respectively, compared to T1, T2 and T3 results of 2.75, 3.83 and 2.82 errors respectively. Performing post hoc contrasts of the treatment means using the Tukey test, we find that at the $P < .05$ level, three contrasts-- T5 compared to T1, T5 compared to T2, and T4 compared to T2 -- can be said to exhibit a significant difference. The other contrasts with the reference treatment T1 and T2 are not significant.

In summary, the experiment has provided consistent results which support the proposition that treatments using user-generation of text labels (T4 and T5) provides better recall error performance than existing treatments that use only system provided icons or system provided icons plus system provided labels (T1 and T2). In particular, the practical hybrid treatment T5 performance was significantly better than T1 (the best performing reference). In addition, the experiment has shown that the treatment that utilized user selection of an icon alone (T3) did not result in a significant improvement when compared to system provided icons alone (T1). Finally, user-selection of an icon only (T3) is not significantly better than if the system provides the icon (T2).

7.4 Framework Interpretation and Interface Design Implications

Having used the experiments to show that the treatments exhibit significant performance differences with respect to recall error performance, this section will now interpret the results using the framework metrics.

We will base all comparisons on T1 (system-provided icon only) which as we stated previously is an existing treatment widely used in many human-computer interfaces. We will eliminate the second existing interface treatment T2 from further consideration and analysis, as it provided the same or worse recall performance compared to T1. As well, T3 (user-selected icons only) and T4 (user-selected icon plus user-generated label) show little improvement potential. The analysis from this point will concentrate on comparisons of T5 (system-provided icon plus user-generated label) with the existing T1 treatment, which we will denote as our reference treatment.

The first metrics of interest are <u>OLtime</u>, that is the amount of time required for subjects to reach a common or benchmark level of mastery, and RLtime, the time required at the retrieval session to complete the task. Table 9 summarizes the mean times to complete the original learning and retrieval tasks, as well as the calculated effectiveness scores. Neither OL or RL time were significant, although OL time is marginally so at the $p < .10$ level.

Table 9 Effectiveness Comparisons of the Treatments

Treatment:	T1	T5
OL Time (sec)	316.56	401.31**
RL Time (sec)	68.3	56.6
OLeffectiveness	1.0	0.73
RLeffectiveness	1.0	1.17
CLeffectiveness	1.0	0.95

** Marginally significant using Tukey test @ $p < .10$

As discussed in the framework, tradeoffs with respect to time requirements of the interfaces are of interest to designers.

From the Original Learning Effectiveness Scores, T5 requires more learning time and hence has a lower original learning effectiveness score than the reference T1. This is not surprising and is expected, as T5 requires additional time for users to determine (generate) an appropriate text label for the functions they are learning.

However, T5 exhibits significantly better recall performance. Also, based on RLeffectiveness scores, Interface T5 requires less time than the reference interface (but not a significant difference) to recall and complete the previously learned task. Based on the fact that T5 is also performed with roughly half of the recall errors of T1, then T5 appears to be the treatment of choice. The tradeoff is that it requires about 26% more time than T1 to initially learn the task, or about an additional 1.25 minutes. However, it then exhibits 50% fewer errors and requires 17% less time, or about 0.2 minutes less than T1 to recall and complete the task. Based on the CLeffectiveness score of .95, T5 is comparable to T1 when trading off additional learning effort with reduced recall time. It should be noted that CLeffectiveness places equal weight on both initial learning and recall performance. In a typical application, a user only performs <u>one initial learning session but many recall sessions</u>. Based on the intended application, a designer may decide to weight RLeffectiveness more highly than OLeffectiveness. These data are now available to the design team to use in conjunction with other data on usability such as user preference or likeability, ease of use, usefulness etc. available from other existing methods to decide on overall design direction.

Recall as well that interface treatment T5 is a hybrid design which eliminated the requirement for users to select from a set of icons. This treatment would thus allow users to use the software at any location, avoiding the need for a universal and standardized icon set. It would appear that from a learnability perspective there is little reason to impose the T4 technique on the interface design; indeed the system-provided icons plus user-generated text label solution (T5) provides performance equal or better to T4 along all variables tested.

In conclusion, from a recall perspective, T5 appears to be a useful interface design to pursue. At this point in a development environment, a design team would begin reiteration of steps 2 to 4 with the intention of refining the T5 treatment based on this evaluation and on additional usability specification performance measures not covered here such as likability, perceived utility etc. A possible learnability objective may be to improve CLeffectiveness to 1.0, so that there is less differential learning time.

7.5 Comparison of Results Using Other Evaluation Methods

It was proposed in chapter 3 that inclusion of recall performance in learnability evaluation would provide a more complete evaluator of total learnability. As mentioned previously, typical learnability evaluators include initial learning time, error performance, percent task completion etc, but usually within an equivalent to the Original Learning Domain. From Table 10, an analysis of original learning variables OLtime and OLerrors only would indicate that T1 is the best interface.

Table 10

Summary of Original Learning Domain Evaluators

	T1	T4	T5
OLerrors	1.25	1.75	1.25
OLtime	316.56	462.13*	401.31**
OLtime stdev.	89.732	104.87	121.26
NPR	3.53	4.40	3.31

* Significantly different than T1 @ $p < .05$

** Marginally different than T1 @ $p < .10$

In conditions of measuring the effectiveness of initial performance, T1 may well be the most time efficient. However, we have argued in section 3.1.5 that this may be reflecting how efficiently we are aiding the user to reduce cognitive load, aid in repetition techniques to hold the task procedures in short term memory, or any other cognitive operator of importance in improving initial performance. These measures fail us in evaluating how well we have transferred the skill to long term memory for subsequent retrieval, which may be of equal or greater importance, depending on the intended use of the interface and the target user group. As well, we have mentioned that the original learning task is typically performed once, but the recall task is performed many times.

In addition, a calculation of Normalized Performance Ratios (NPR, section 3.4) identifies T4 as providing the interface which "best satisfies its operational objectives" (Moffat, 1990). To be fair, the NPR may not be applicable in these test circumstances, as its evaluation may require much larger sample sizes to evaluate than we have provided, or that may be practical in a design environment. Also, it is not clear that we have been able to meet the control requirements that NPR may require, as its operationalization in a test environment has not been reported by its author. As discussed in section 3.4, setting the operational objectives for the NPR metric may not be consistent with learning speed and retention, but more reflective of some measure of ease of interface manipulation.

7.6 Summary

The experiment has provided support for rejection of the null hypothesis H_0 that the treatments provide the same RLerror performance. In addition, post-hoc tests show T5 mean RL performance to be significantly better than both T1 and T2.

The framework variables relating to recall error performance have demonstrated that they can provide reliable and significant results when testing multiple interface treatments using the prescribed methodology. The contribution of new measures of recall performance provides additional dimensions of learnability which can be incorporated when appropriate by the design community. In addition, the variable OLdifmeasure has demonstrated to be an excellent covariate which appears to be related to individual ability and should be added to the framework variable set.

The concomitant set OLdifmeasure, Attitude and LSI have been shown to reduce the data variance and allow better interpretation of treatment effects. In addition, inclusion of the covariate OLdifmeasure has demonstrated its potential contribution of reducing the sample sizes required to evaluate interfaces with respect to recall. This is an important practical consideration in a development environment. However, attitude and LSI individually have not consistently been significant or reliable concomitant variables in this test. Other individual difference variables appear to have little impact on recall performance in this test.

The ToMCIL tool has demonstrated its practical utility in an interface test environment. Finally, the framework variables and efficiency scores allowed analysis of the interface treatments along dimensions of learning speed as well as retention performance. This has provided additional performance data available for subsequent design iterations. Also, a treatment which performs significantly better on recall scores and almost as well on learning speed has been highlighted by these tests. This would have been overlooked if the more traditional measures of learning speed and initial error performance alone were used as surrogates for learnability.

Finally, the framework has shown practical application in an actual design and test situation, requiring as few as 16 subjects per treatment. We note that the pilot study using 9 subjects per treatment also gave a reasonably clear indication of treatment differences.

CHAPTER 8

Summary & Conclusions

8.1 Discussion of Results

There has been significant progress in the last two years in usability design and specification methodology and its exploration and adoption by the design community. This framework is consistent with a usability approach so that the rich range of usability metrics can be augmented with measures of effective recall and learnability.

The framework should be used with the following qualifications:

1) Task Complexity

The interface tests have demonstrated use of the framework on procedural tasks with near-identical protocols so that task complexity has been held approximately constant. At this time we recommend use of the framework within this restriction until the measurement of task complexity and its impact on interface performance has been investigated and clarified.

The treatments tested should at this stage be of similar implementation. Radically different tasks cannot be compared with respect to learning and recall until much improved practical task analysis and categorization techniques are developed which are optimized for graphical interfaces.

2) Individual differences

The research has proposed a practical surrogate for individual ability. The variable OLdifmeasure is a normalized learning score variable which has been shown to be easy to measure and provides sufficient reduction in data variance that the introduction of IQ tests or other individual ability measures does not seem required for design testing applications. The remaining individual difference variables proposed have not been shown to be significant in this work. A substantial amount of work is presently being conducted elsewhere in the area of adaptive interface research, concentrating on individual differences (Browne, et al 1990), and the conclusions and techniques developed in that area may be applicable in future extensions of the framework. However, we recommend continued measurement of the variables in order to provide complete user profiles.

3) Iterations Measures

The framework has proposed the use of iterations measures as additional evaluators of task-skill acquisition benchmarks in addition to error performance. However, the nature of the tasks tested in the study were of a procedural nature requiring a single iteration, and did not demonstrate the potential multiple iteration comparison capability. It is expected that other task extensions and more complex task implementations may require this ability and so it remains a proposed but undemonstrated part of the framework for subsequent testing. In addition to the demonstrated framework scores, it may prove useful to provide an indication of the task efficiency, that is the relative amount of relearning required (number of iterations) to re-achieve the level of proficiency exhibited at the time of original learning. Such a score could take into consideration the fact that more complex tasks will require more iterations and time than less complex tasks before they are learned to the autonomous level. However, the most efficient implementation from the point of minimizing user recall frustration should result in requiring only a single iteration for success at the time of retrieval. Recognizing that this represents the most efficient implementation, we can define the learning efficiency score consistent with the concept of general input/output efficiency as follows:

CLefficiency Score =

$$\frac{1 + OL(\#\ iterations) - RL(\#\ of\ iterations)}{OL(\#\ iterations)}$$

Testing and demonstration of this score remains for future research.

4) Error Types

The concept of error type may provide some potential refinement of the framework in later applications so that errors of a non-impacting nature (eg. slips) could be distinguished from more severe and event-altering errors. This distinction could be similar to the error definitions apparently used for interface testing at IBM (ie Impact 1/ Impact 2 errors--see Lewis, et al 1990; also Rosson, 1987)

Definition of error categories could potentially allow higher resolution of measuring task proficiency. For example, in the early cognitive stages of learning a task, we may expect a user to make an error of a type that is fatal to the operation of the task (in the extreme, turning off the computer). This would be defined as an error of the cognitive stage type. However, once the autonomous skill level were achieved, we would expect the error types to be of a more benign nature (misspelling a command, or forgetting the name of a file). In addition, the error may or may not result in an erroneous result (recall Lewis & Norman, 1986). For example, if a user is attempting to use a mouse to place a cursor on and activate an area of the interface screen that is undefined or cannot respond, the system may inform the user of this fact. This would be defined as an error with no effect on the task result. In contrast, if the user successfully sends an electronic message to a different person than intended, then the error is of a type that affects the task result (wrong result).

The framework could require that the designer define each category of user error for the particular task as to its type and result. Thus each error category could be an instance of, say, a new variable ET (Error Type) defined as being an error displaying a task proficiency of the cognitive, associative or autonomous skill level. In addition, each error could also be classified as to its task result (no major result eg. syntax error or wrong result, eg. inputting an incorrect parameter). The ET (Error Type) variable could be measured while a subject is learning to perform a task on a computer interface. The type and number of errors could then be used to determine the task skill proficiency level exhibited by a subject. For example, if an autonomous skill level has been achieved, then the user should not have performed any errors of the cognitive or associative type during the most recent task iteration.

8.2 Future extensions and research

Future investigations of the following research areas would be valid extensions from the experience and results obtained in developing and testing this framework:

1) Trial and incorporation of other instruments to measure and incorporate individual difference variables such as attitude, experience (interface type) and cognitive learning (or other) style to investigate the existing demonstrated covariate ambiguity as well as trials of some of the untested scores in the proposed framework (CLefficiency, OLiterations, RLiterations).

2) Use of the framework in design and production shops and incorporating changes as required to make the framework itself usable, and incorporating the framework within current usability engineering practice. This would require development of a fully productized ToMCIL implemented in the UIMS or CASE environment used in typical design environments.

3) The Jonassen taxonomy is a recent snapshot of an available set of learning strategies. There is an ongoing requirement to update this taxonomy to keep current with modern cognitive theories and recommendations.

4) The practical result of the framework test is a rather unique interface style modification (T5). Exploration of the new interface style in an actual production system with further extensions and testing would appear to be justified.

5) The framework can be extended to apply to more complex cognitive tasks with the inclusion of better task complexity measures if and when available.

6) The framework can be used to create and explore other potential interface styles which may improve recall and learning performance.

8.3 Conclusions and Contribution

The research has outlined the typical conditions present for managers and other discretionary users at the time of initial computer learning and has presented the case for the importance of recall as it applies to computer learnability. As the evolving methodologies of usability specification and engineering have placed primary emphasis on initial learning speed and error performance, the research has proposed a modified framework consistent with usability practice, but has contributed additional measures and techniques for determining long-term retention and recall of a computer interface design in a practical test environment.

The framework has also proposed a technique for enriching the designer's set of implementation possibilities by introducing the concept of learning strategy into the iterative design and development process.

The framework proposed has been demonstrated to provide a methodology and measurement criteria that were useful in evaluating a set of five user interface prototype designs. In addition, an automated tool as proposed in the framework has been created which fulfils its objectives. The relative usefulness of the framework for adding to a designer's toolkit new and improved user interface treatments based on appropriate learning strategies has been demonstrated.

Finally, the framework evaluation results have shown support for the proposal to include recall measures as an additional learnability metric to augment existing measures when appropriate, as their inclusion has resulted in different treatment ranking than if the existing techniques based on original learning time were used.

Bibliography

Akin, O., & Rao, D. R.,[1984] "Efficient computer-user interface in electronic mail systems", International Journal of Man-Machine Studies, 22, 589-611.

Allwood, C. M. [1986] "Novices on the computer: a review of the literature" Int. Journal of Man-Machine Studies, 25, 633-658.

Anderson, J. R., [1981] Cognitive Skills and their Acquisition, Hillsdale, N.J.: Lawrence Erlbaum.

Anderson, J. R., [1985] Cognitive Psychology & its Implications, New York: Freeman & Co.

Antin, J. F., [1988], "An empirical comparison of menu selection, command entry and combined modes of computer control", Behaviour and Information Technology, 7,#2,173-182.

Baecker, R. M., & Buxton, W. A. S., [1987] Human-Computer Interaction, A Multidisciplinary Approach, Los Altos, Calif.: Morgan Kaufmann.

Bennett, J. L. [1984] "Managing to meet usability requirements: Establishing and meeting software development goals" In J. L. Bennett, D. Case, J. Sandelin, & M. Smith (Eds.), Video Display Terminals: usability issues & health concerns. Englewood Cliffs, N. J.: Prentice Hall.

Benyon, D. & Murray, D. [1988], "Experience with Adaptive Interfaces", The Computer Journal, 31,#5,465-473.

Black, J. B., & Sebrechts, M. M., [1981] "Facilitating human-computer communication." Applied Psycholinguistics, 2, 149-177.

Bly, S. A. & Rosenberg, J. K., [1986] "A Comparison of Tiled and Overlapping Windows", CHI'86 Proceedings, 101-106

Booth, P. A. [1989] An Introduction to human-computer interaction, Hillsdale, N. J. : Lawrence Erlbaum.

Bower, G. H., Karlin, M.B., & Dueck, A. [1975] "Comprehension and memory for pictures." Memory & Cognition, 3, 216-220.

Bayman, P., & Mayer, R. E., [1982] "Instructional Manipulation of User's Mental Models for Electronic Calculators", International Journal of Man-Machine Studies, 20, 189-199.

Bjorn-Anderson, N., [1988], "Are 'Human factors' Human?", The Computer Journal, 31, #5, 386-390.

Black, J. B. & Moran, T. P. [1982] "Learning & Remembering Command Names", Proceedings of the Conference on Human Factors in Computing Systems.

Bott, R. A. [1979] "A Study of computer learning:Theory & Methodologies.", Centre for Human Information Processing Technical Report #82, San Diego: University of California (reported in Farr, 1989).

Bower, G. H. & Winenz, P. [1970] "Comparison of associative learning strategies" Psychonomic Science, 20, 119-120.

Brooke, J., Bevan, N., Brigham, F., Harker, S., & Youmans, D. "Usability Statements and Standardisation - Work in Progress in ISO", Human-Computer Interaction - INTERACT '90, Amsterdam: North Holland.

Browne, D., Totterdell, P., & Norman, M. [1990] Adaptive User Interfaces. London: Academic Press.

Bury, K. F. [1984] "The interactive development of usable computer interfaces." In B. Shackel (Ed.), Human-computer interaction- Interact'84: Proceedings of the first IFIP conference on human-computer interaction, London. Amsterdam: North Holland.

Card, S. K., Moran, T., & Newell, A., [1983] The Psychology of Human-Computer Interaction, Hillsdale, N.J.: Lawrence Erlbaum.

Card, S. K., Pavel, M, & Farrell, J.E., [1984] "Window Based Computer Dialogues", Human-Computer Interaction- Interact'84, Amsterdam: North-Holland.

Carroll, J. M., & Mack, R. L. [1984] "Learning to use a word processor: By Doing, By Thinking and By Knowing" in Baecker op cit, pp279-297.

Carroll, J. M., [1989] "Evaluation, Description and Invention: Paradigms for Human-Computer Interaction" in 1989 Edition of Advances in Computers.

Carroll, J. M. & Rosson, M. B. [1985] "Usability specifications as a tool in interactive development." In H R Hartson Ed. Advances in human-computer interaction. Norwood, N. J.: Ablex

Carroll, J. M. & Rosson, M.B., [1984] Usability Specifications as a tool in iterative development, Research Report # 10437, Yorktown Heights, N. Y.: IBM Watson Research Center. (as reported in Frese, op cit)

Christensen, [1988], Experimental Methodology, 4th edition, Boston: Allyn & Bacon.

Chubb, G. P., Laughery, K., & Pritsker, A. B. [1987] "Simulating manned systems" in Handbook of Human Factors, Salvendy, G. ed. New York:Wiley.

Craik, F. I. M., & Lockhart, R. S. [1972] "Levels of processing: A framework for memory research." Journal of Verbal Learning & Verbal Behaviour, 11, 671-684.

Collins, A. M., & Quillian, M. R., [1969] "Retrieval time from semantic Memory", Journal of Verbal and Learning Behaviour, 8, 240-247.

Dayton, C. M. [1970], The design of educational experiments, New York: McGraw Hill.

Dansereau, D. F. [1978] "The development of a learning strategies curriculum", in Learning Strategies, New York: Academic Press.

Donaldson, W., & Bass, M. [1980] "Relational information and memory for problem solutions", Journal of Verbal Learning and Verbal Behaviour, 19, 26-35.

Eason, K. D., & Damodaran, L. [1981] "The needs of the commercial user" in Coombs, M. & Alty, J. (Eds.) Computing Skills and the User Interface, New York: Academic Press.

Ekstrom, R.B., French, J.W., & Harman. H.H. [1976] Manual for Kit of Factor-Referenced Cognitive Tests, Princeton, N.J.:ETS.

Ericsson, K. A. & Simon, H. A. [1980] "Verbal reports as data", Psychological Review, 87, 215-251.

Erlich, R. W., & Williges, R. C.(Editors), [1986] Human-Computer Dialogue Design, Amsterdam: Elsevier.

Egan, D. E., Remde, J. R., Gomez, L. M., Landauer, T. K., Eberhardt, J. & Lochbaum, C. C.,[1989] "Formative Design Evaluation of SuperBook", ACM Transactions on Information Systems, 7, #1, 30-57.

Evans, J. St. B. T., [1988], "The knowledge elicitation problem: a psychological perspective", Behaviour and Information Technology, 7, #2, 111-131.

Farooq, M. U. & Dominick, W. D. [1988] "A survey of formal tools and models for developing user interfaces" Int. Journal of Man-Machine Studies, 29, 479-497.

Farr, M. J., [1987] The Long Term Retention Of Knowledge & Skills: A Cognitive and Instructional Perspective, New York: Springer-Verlag.

Fitts, P. M., [1954], "The information capacity of the human motor system in controlling amplitude of movement", Journal of Experimental Psychology, 47, 381-391.

Fitts, P. M. & Posner, M. I. [1966] Human Performance, Belmont. California: Brooks, Cole.

Foss, D. J. & DeRidder, M. [1987] "Technology Transfer: On Learning a New Computer Based System" In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction. Cambridge Mass.: MIT Press.

Frese, M., [1989] "Human Computer Interaction within an Industrial Psychology Framework", Applied Psychology: An International Review, 38, #1, 29-44.

Gabel, P. C., [1987] An Investigation of the Relation Between Remembering and Learning, Ph.D Thesis, Hamilton: McMaster University.

Gardiner, M. M. & Christie, B. (Eds.) [1987] Applying Cognitive Psychology to User-Interface Design. New York: Wiley.

Gawron, V. J., Drury, C. G., Czaja, S. J., Wilkins, D. M. [1989] "A Taxonomy of independent variables affecting human performance", International Journal of Man-Machine Studies, 31, 643-672.

Gist, M. E., Schwoerer, C., & Rosen, B. [1989] "Effects of Alternative Training Methods on Self-Efficacy and Performance in Computer Software Training", Journal of Applied Psychology, 74, #6, 884-891.

Good, M. D., Whiteside, J. A., Wixon, D. R., & Jones, S. J. [1983] "Building a User-Derived Interface", Communications of the ACM, 27, #10, 1032-1043.

Graf, P., [1982] "The Generation Effect and a Related Memory Phenomenon: An Interpretive Framework", Journal of Verbal Learning & Verbal Behaviour.

Graf, P., [1980] "Two Consequences of Generating: Increasing Inter- and Intraword Organization of Sentences", Journal of Verbal Learning and Verbal Behaviour, 19, 316-327.

Graf, P., [1981] "Reading & Generating Normal & Transformed Sentences", Canadian Journal of Psychology, 35, #4, 293-308.

Greenwald, A. G. & Johnson, M. M. S. [1989] "The Generation effect extended: Memory enhancement for generation cues", Memory and Cognition, 17, #6, 673-681.

Greif, S. & Gediga, G. [1987] "Problems and Perspectives of GOMS and Keystroke-level models", in Frese, M., Ulich, E., & Dzida, W. (Eds.) Psychological issues of human-computer interaction in the workplace, 357-377. Amsterdam: North Holland.

Grudin, J. & Barnhart, P. [1984] "The role of prior experience in command name abbreviation", in Human -Computer Interaction-Interact'84 conference proceedings, Amsterdam:North Holland.

Hammond, N., Morton, J., MacLean, A., & Barnard, P. [1983] "Fragments and Signposts: Users' models of the system", Proceedings of the 10th Symposium on Human Factors in Telecommunications.

Hanley, J. R., [1984] "Dual Processes in Recognition and in Recognition Failure", Memory & Cognition, 12, #6, 575-580.

Hayes, P. J. & Reddy, D. [1981] "Steps towards Graceful Interaction in Spoken and Written Man-Machine Communication", International Journal of Man-Machine Studies, 19, 231-284.

Hemenway, K., [1982] "Psychological Issues in the use of Icons in Command Menus", Proceedings of the Conference on Human Factors in Computing Systems, Gaithersberg, Maryland.

Hiltz, S. R. & Turoff, M., [1978] The Network Nation: Human Communication via Computer, Reading, Mass,: Addison-Wesley.

Hintzman, D. L. [1989], "Human Learning and Memory: Connections and Dissociations", University of Oregon Institute of Cognitive & Decision Sciences Technical Report #89-8.

Hirschheim, R., & Newman, M. [1988], "Information Systems and User Resistance: Theory and Practice", The Computer Journal, 31, #5, 398-408.

Hurlock & Montague [1982] Skill retention and its implications for navy tasks: An Analytical Review , NPRDC Report SR 82-21, San Diego, CA: Navy Personnal Research & Development Center (as reported in Farr, 1987).

Jakobson, G., Lafond, C, Nyberg, E., & Piatetsky-Shapiro, G. [1986] "An Intelligent Database Assistant", IEEE Expert, Summer 1986, 65-78.

Jones, W. P. & Dumais, S. T., [1986], "Experimental Tests of Reference by Location versus Name", ACM Transactions on Office Information Systems, 4, #1, 42-63.

Jonassen, D. H., 1988 "Integrating Learning Strategies into Courseware to Facilitate Deeper Processing", in Jonassen, D. ed. Instructional Designs for Microcomputer Courseware, New Jersey: Erlbaum.

Johnson-Laird, P. N.,[1983] Mental Models:Towards a Cognitive Science of Language, Inference, and Consciousness, Cambridge: Cambridge Press.

Johnson-Laird, P. N., [1988] The Computer and the Mind: An Introduction to Cognitive Science, Cambridge, Mass.: Harvard Univ. Press.

Karat, C., [1990] "Cost - Benefit Analysis of Iterative Usability Testing", Human-Computer Interaction - INTERACT '90, Amsterdam: North Holland.

Kerlinger, F. N. [1973] Foundations of Behavioral Research, second edition, New York: Holt, Rinehart & Winston.

Kiras, D. & Polson, P. G. [1985] "An approach to the formal analysis of user complexity." Int. Journal of Man-Machine Studies, 22, 365-394.

Kitajima, M. [1989] "A formal representation system for the human-computer interaction process." Int. Journal of Man-Machine Studies, 30, 669-697.

Klix, F. & Wandke, H.-editors, [1986] Man-Computer Interaction Research--MACINTER I, Amsterdam: North Holland.

Kolb, D. A., [1984] Experiential Learning: Experience as the Source of Learning and Development. New Jersey: Prentice Hall.

Kolers, P. A., [1979] "A pattern analyzing basis of recognition" In Cermak & Craik (Eds.) Levels of Processing in Human Memory, Hillsdale, N.J.: Lawrence Erlbaum.

Hiltz, S. R. & Turoff, M., [1978] The Network Nation: Human Communication via Computer, Reading, Mass,: Addison-Wesley.

Lansdale, M.W. Simpson, M., & Stroud, T.R.M. [1987] "Comparing Words & Icons as Cue Enrichers in an Information Retrieval Task", Human-Computer Interaction-INTERACT '87, Amsterdam: North Holland.

Landauer, [1987] "Relations between cognitive psychology and computer system design", in Carroll, J. M. ed., Interfacing Thought: Cognitive Aspects of Human-Computer Interaction, pp. 1-25, Cambridge, Mass.: Bradford/MIT Press.

Lansdale, M. W.[1988] "On the memorability of icons in an information retrieval task.", Behaviour and Information Technology, 7, #2, 131-152.

Ledgard, H. Singer, A. & Whiteside, J. [1981] Directions in Human Factors for Interactive Systems. Berlin: Springer Verlag.

Lewis, C. & Norman, D.A. [1986] "Designing for Error." In Norman, D. A. & Draper, S. W. (eds.) [1986] User Centred System Design, Hillsdale, N.J.: Lawrence Erlbaum.

Lewis, J. R., Henry, S. C., & Mack, R. L. "Integrated Office Software Benchmarks: A Case Study", Human-Computer Interaction - INTERACT '90, Amsterdam: North Holland.

Lindgaard, G. & Perry, L., [1988a] "Toward a Solution to Vocabulary Problems in Computing: a Measure of Goodness of Fit", Ergonomics, 31, #5, 785-801.

Lindgaard, G. & Perry, L.,[1988b] "Making Life Easier for Computer Novices: Some Factors Determining Initial Performance", Ergonomics, 31, #5, 803-816.

Madigan, S. [1983] "Picture Memory" in Yuille, J. C. ed. Imagery, Memory and Cognition:Essays in honor of Allan Paivio, Hillsdale: Lawrence Erlbaum.

Mayer, R. E. [1981] The Promise of Cognitive Psychology, San Fransisco: Freeman.

McDaniel, M. A., Waddill, P.J., & Einstein, G. O. [1988], "A contextual account of the generation effect: A three-factor theory" Journal of Memory & Language, 27, 521-536.

Meyers-Briggs, I., [1962] The Meyers-Briggs Type Indicator Manual, Princeton: Educational Testing Service.

Miller G. A., [1956] "The Magical Number Seven, plus or minus two: Some Limits on our Capacity for Information Processing", Psychological Review, 63, 81-97.

Mintzberg, H., [1973] The Nature of Managerial Work, New York: Harper & Row.

Moffat, B. [1990], " Normalized Performance Ratio- a measure of the degree to which a man-machine interface accomplishes its operational objective." Int-Journal of Man-Machine Studies, 32, 21-108.

Moran, T. P., [1981] "The command language grammar: a representation for the user interface of interactive computer systems." Int- Journal of Man-Machine Studies, 15, 3-50.

Moran, T. P. [1986] "Analytical performance models: A contribution to a panel discussion." In M. Mantei & P. Orbeton (Eds.), Human-factors in computer systems-III: Proceedings of the CHI'86 Conference, Boston. Amsterdam: Elsevier.

Morrison, P. R., & Nobel, G. [1987] "Individual Differences and ergonomic factors in performance on a videotex-type task." Behaviour and Information Technology, 1, #1 69-88.

Murdock B. B. Jr., [1961] "The retention of individual items" Journal of Experimental Psychology, 62, 618-625

Murphy, E. D. & Mitchell, C. M., [1986], "Cognitive attributes: implications for display design in supervisory control systems", Int. Journal of Man-Machine Studies, 25, 411-438.

Murray, D. [1988] A Survey of User Cognitive Modelling, National Physical Laboratory Report DITC 92/87, Teddington, U.K.:NPL.

Myers, A. [1987] Experimental Psychology 2nd Edition, Monterey: Brooks/Cole.

Myers, B.A. [1989], "User-Interface Tools: Introduction and Survey.", IEEE Software, Jan. 89, 15-23.

Myers, J.L. [1979], Fundamentals of Experimental Design, 3rd edition, Boston: Allyn and Bacon.

Nelson, D. L., [1979] "Remembering pictures & words: Appearance, significance, and name." In Cermak & Craik (Eds.) Levels of Processing in Human Memory, Hillsdale, N.J.: Lawrence Erlbaum.

Neter, J., Wasserman, W., & Kutner, M. H. [1985] Applied Linear Statistical Models, Second Edition, Homewood, Il: Irwin.

Newell, A. & Card, S. K. [1985] "The prospects for psychological science in human-computer interaction", Human Computer Interaction, 1, 209-242.

Newell, A. & Simon H. A. [1972] Human Problem Solving, Englewood Cliffs. N.J.: Prentice Hall.

Norman, D.A.[1986] "Cognitive Engineering" in Norman & Draper [1986] op. cit.

Norman, D. A. & Draper, S. W. (eds.) [1986] User Centred System Design, Hillsdale, N.J.: Lawrence Erlbaum.

Olson, J. R. [1987] "Cognitive Analysis of People's use of Software" In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction. Cambridge Mass.: MIT Press.

Paivio, A. [1971] Imagery and Verbal Processes, New York: Holt Rinehart & Winston.

Paivio, A. & Lambert, W. [1981], "Dual Coding and Bilingual Memory", Journal of Verbal Learning and Behaviour", 20, 532-539.

Parasuraman, S. & Igbaria, M [1990] " An examination of gender differences in the determinants of computer anxiety & attitudes toward microcomputers among managers." Int. Journal of Man-Machine Studies, 25, 327-340.

Payne, J.W. [1982], "Contingent Decision Behavior", Psychological Bulletin, 92, #2, 382-402.

Payne, S. J. [1984] "Task Action Grammars", In Shackle, B. (ed.) INTERACT '84. 139-144. Amsterdam: North Holland.

Paxton, A. L. & Turner, E. J., [1984] "The Application of Human Factors to the Needs of the Novice Computer User", Int. Journal of Man-Machine Studies, 20, 137-156.

Perlman, G. [1985] "Making the right choice with menus", Human- Computer Interaction-Interact'84, Amsterdam: North Holland.

Polson, P. G. [1987] "A Quantitative theory of Human-Computer Interaction." In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction. Cambridge Mass.: MIT Press.

Reisner, P. [1981] "Formal Grammar and Human Factors Design of an Interactive Graphical System", IEEE Transactions on Software Engineering, 7 #2 229-240.

Reisner, P. [1987] "Discussion: HCI, What it is and What Research is Needed?" In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction. Cambridge Mass.: MIT Press.

Rigney, J. [1978] "Learning Strategies, A theoretical perspective", in Learning Strategies, New York: Academic Press.

Roberts, T. L. & Moran, T. P. [1983] "The Evaluation of Text Editors: Methodology and Empirical Results." Communications of the ACM, 26, #4, 265-.

Rockart, J. F., & Crescenzi, A. D., [1984] "Engaging Top Management in Information Technology", Sloan Management Review, 3, pp-pp.

Rosson, M. B. & Grischowsky, N. L. [1987] "Transfer of Learning in the Real World", in Human Computer Interaction-INTERACT'87, Amsterdam: North Holland.

Rumelhart, D. E. & Ortony, A. [1977] "The Representation of knowledge in memory" in Anderson, ed. Schooling and the acquisition of knowledge, Hillsdale, N. J.: Lawrence Erlbaum.

Sein, M. K., Bostrom, R. P.& Olfman, L. [1987] "Conceptual Models in Training Novice Users" in Human-Computer Interaction-INTERACT'87, Amsterdam: North Holland.

Shackel, B. [1988] "Introduction to the design of end-user interfaces", in Designing End-User Interfaces, State of the Art Report 15:8. Oxford: Pergamon.

Shackel, B. [1986] "Ergonomics in design for usability." In M. D. Harrison & A. F. Monk (Eds.) People & Computers: Designing for usability. Proceedings of the BCS HCI specialist group. Cambridge: Cambridge University Press.

Shackel, B. [1981] "The concept of usability." In Proceedings of the IBM software & Information usability symposium. as described and cited in Booth, P. A. [1989] An Introduction to human-computer interaction, Hillsdale, N. J. : Lawrence Erlbaum.

Shneiderman, B. [1979] "Human Factors experiments in designing interactive Systems" IEEE Computer, 12, 9-19.

Shneiderman, B. [1984] "Review of Card, S.K., Moran, T.P., & Newell, A. The Psychology of Human-Computer Interaction", Datamation, 30, 236-240.

Shneiderman, B. [1983] "Direct Manipulation: A Step Beyond Programming Languages" IEEE Computer, Aug 83, 57-69.

Shneiderman, B. [1986] Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley.

Shepard, R. N. [1967] " Recognition memory for words, sentences, and pictures", Journal of Verbal Learning and Verbal Behaviour, 6, 156-163.

Simon D. P. & Simon, H. A. [1978] "Individual differences in solving physics problems", in R. S. Siegler (Ed.), Childerens Thinking: What develops?, Hillsdale, N. J.: Lawrence Erlbaum.

Simon, H. A. [1980] "Information Processing theory of human problem solving", in W. K. Estes (Ed.) Handbook of learning and cognitive processes: Human Information Processing, Hillsdale, N. J.:Lawrence Erlbaum.

Slamecka, N. J., & Graf, P. [1978] "The Generation effect: Delineation of a phenomenon", Journal of Experimental Psychology: Human Learning and Memory, 4, 592-604.

Slator, B. M., Anderson, M. P., & Conley, W., [1986] "Pygmalion at the Interface", Communications of the ACM, 29, #7, 593-604.

Smith, D.C., Irby, C., Kimball & Verplank, B. [1981] "Designing the Star user interface", Byte, 1981, August.

Smith, D. M., & Kolb, D. A., [1986] User's Guide for the Learning Style Inventory, Boston: McBer.

Thomas, J. C., & Kellogg, W. A. [1989], "Minimizing Ecological Gaps in Interface Design", IEEE Software, Jan 1989, 78-86.

Tullis, T. J. [1985] "Designing a menu-based interface to an operating system", in Proceedings of CHI'85 Human Factors in Computing Systems, pp. 79-84, New York:ACM.

Tulving, E. & Thompson D. M. [1973] "Encoding Specificity and Retrieval Processes in Episodic Memory", Psychological Review, 80, 352-373.

Umanath, N. and Scamell, R.W., [1988] "An Experimental Evaluation of the Impact of Data Display Format on Recall Performance", Communications of the ACM, 31, #4, 562-570.

Vainio-Larsson, A., "Evaluating the Usability of User Interfaces: Research in Practice", Human-Computer Interaction - INTERACT '90, Amsterdam: North Holland.

Vassiliou, Yannis-editor, [1984] Human Factors and Interactive Computer Systems, Norwood, N.J.:Ablex.

Vessey, I., [1988], "Expert-novice knowledge organization: an empirical investigation using computer program recall", Behaviour and Information Technology, 7, #2, 153-172.

Weinstein, C. E. & Mayer, R. E. [1985] "The teaching of learning strategies" in Handbook of research on Teaching (3rd Edition), New York: Macmillan.

Wesolowsky, G. O. [1976] Multiple Regression and Analysis of Variance, New York: Wiley.

Wessells, M. G., [1982] Cognitive Psychology, New York: Harper & Row.

Whiteside, J., Bennett, J., & Holtzblatt, K. [1988] "Usability Engineering: Our Experience and Evolution", Handbook of Human-Computer Interaction, Amsterdam: North Holland.

Whiteside, J. Jones, S., Levy, 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.

Whiteside, J. & Wixon, D. [1987] "Discussion: Improving human-computer interaction - a quest for cognitive science." In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction. Cambridge Mass.: MIT Press.

Whiteside, J., Wixon, D. & Jones, S. [1987] "User performance with command, menu, and Iconic interfaces" in Hartson, H. R. & Hix, D. ed. Advances in Human-Computer Interaction, Volume 2, Norwood, N.J.: Ablex.

Wickens, C. D., [1984] Engineering Psychology and Human Performance, Columbus Ohio: Merrill.

Williges, R. C., [1987] "The use of Models in Human-Computer Interface Design", Ergonomics, 30, #3, 491-502.

Wilton, J. A., & McLean, R.S., [1984] "Evaluation of a mouse as an educational pointing device", Computers & Education, 4, 455-461.

Wittrock, M. C. [1978] "The cognitive movement in instruction", Educational Psychologist, 15, 15-29.

Wolford, G. [1971] "Function of distinct associations for paired- associative performance", Psychological Review, 73, 303-313.

Yuille, J. C.[1983] ed. Imagery, Memory and Cognition:Essays in honor of Allan Paivio, Hillsdale: Lawrence Erlbaum.

Appendix 1 Preliminary Questionnaire

Computer Learning Project

The computer learning project is a study examining ways of improving how people learn to use computers. It is led by Michael Murphy, and supervised by Dr. N. Archer of the Faculty of Business at McMaster University, Hamilton Ontario.

Your participation in this research involves completing this questionnaire, as well as completion of the Learning Style Inventory. In the laboratory, you will be asked to complete a simple task using a computer. No previous computer experience is required. The study is examining how people learn to use computers. We wish to have both computer novices as well as experts involved in the study. Some individuals will also be asked to comment verbally while the experiment is underway. This will be recorded by tape-recorder.

You are assured total confidentiality. You may withdraw from the study at any time you desire.

Participants will be paid $15.00 for participation in the study. It will require completion of the Learning Styles Inventory (Approx. 20 min) as well as 2 sessions in the laboratory of approx. 30 min. each. These two sessions will be separated by a time delay of about 3 weeks. They will be scheduled at your convenience. You will be paid as soon as you complete the final part of the study.

If you wish to participate, please complete the following questionnaire and sign and date the attached consent form.

Thank you for your time and particpation.

Computer Learning Project

Confidential Preliminary Questionnaire

NAME: ______________________________

SEX: M ____ F ____ AGE: _____

CLASS: _________ UNDERGRADUATE DEGREE: ____________________

PROFESSION OR WORK EXPERIENCE (if appl.): ____________________

Please check off the following description most applicable to you:

____ a) No previous experience with computers whatsoever.

____ b) Some high school training in computers, but little practical hands-on experience since then.

____ c) A basic college or university course on computers or programming with little or no hands-on experience since then.

____ d) No formal courses or training in computers, but a limited amount of exposure in a course, or on the job.

____ e) Regular use of computers either in college or university studies, or on the job.

____ f) Very extensive use of computers either at school or on the job.

____ g) Other ______________________________

Do you have access to a computer at home or at work? (Y or N)_____
If yes, which type:____________________

In the following questions, please circle the number that most closely represents your agreement or disagreement:

	Strongly Agree 1	Agree 2	Not Sure 3	Dis-agree 4	Strongly Disagree 5
1) Computers are easy to use.	1	2	3	4	5
2) Computers are useful in many ways.	1	2	3	4	5
3) I enjoy using computers.	1	2	3	4	5
4) I have no concerns about pressures to learn how to use computers.	1	2	3	4	5
5) I think everyone should learn something about computers.	1	2	3	4	5
6) I expect to use computers frequently during my career.	1	2	3	4	5

Computer Learning Project

Consent Form

I, ______________________________ (print) agree to participate in the Computer Learning Project led by Michael Murphy and supervised by Dr. N. Archer, based on the following terms:

1) The Purpose of the Study
I understand that the purpose of the study is to measure individuals using different computer learning systems in order to determine if there are meaningful differences in the systems with respect to user learning speed and long term retention of the skills learned.

2) Completion of the Learning Style Inventory
I will complete a Learning Style Inventory questionnaire. This is a standardized instrument used to classify individuals with respect to their preferences in learning methods and procedures. This result is confidential, but my own analysis will be released to me at my request.

3) Participation in the laboratory
I agree to participate in performing a task on a computer system which will measure my results in performing the task. I understand that the results will be used to determine the effectiveness of different computer learning approaches on the subject population. All the results are to be confidential.

4) Confidentiality
I understand that my identity will not be revealed in connection with any data or research released as a result of this study.

5) Monetary reward
I will be paid $15.00 upon completion of the second laboratory session.

6) Right to withdraw
I understand that I may withdraw from this study at any time by informing Michael Murphy verbally or in writing. Payment for a session which is not completed will not be provided unless a written statement is made. In the latter instance, a pro-rated amount will be paid.

Signature ______________________________ Date ____________________

I can be reached to arrange an appointment by phone at ______________

Appendix 2 ToMCIL 1) sample error function code
2) sample event function code
3) sample user session file

1.

error6

```
set xtime=time;
object "error6  ";
write subjname, xpush,25,xtime;
set xpush = xpush + 25;
var xfold(9);
set xfold = folder;
write subjname,xpush,9,xfold;
set xpush = xpush + 9;
var xid(5);
set xid = id;
write subjname,xpush,5,xid;
set xpush = xpush + 5;
var xoname(41);
set xoname = oname;
write subjname,xpush,41,xoname;
set xpush = xpush + 41;
var message(30);
set message = "Please click it & try again.";
msg"The mouse is in the wrong area",message;
```

2.

continue

```
set xtime=time;
object "continue";
write subjname,xpush,25,xtime;
set xpush = xpush + 25;
var xfold(9);
set xfold = folder;
write subjname,xpush,9,xfold; set xpush = xpush + 9;
var xid(5);
set xid = id;
write subjname,xpush,5,xid;
set xpush = xpush + 5;
var xoname(41);
set xoname = oname;
write subjname,push,41,xoname;
set xpush = xpush + 41;
go 2;
```

3.

Subjectname

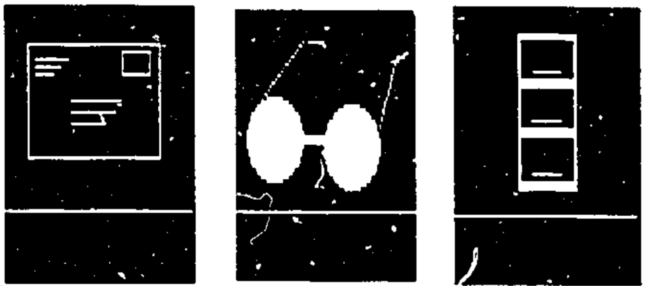
Tue	Jan	01	00:01:11	1980				
Tue	Jan	01	00:01:52	1980	base	1	continue	
Tue	Jan	01	00:01:55	1980	base	2	continue	
Tue	Jan	01	00:01:58	1980	base	3	mailicon	
Tue	Jan	01	00:02:07	1980	mail	1	Ready	read
Tue	Jan	01	00:02:12	1980	mail	1	continue	
Tue	Jan	01	00:02:17	1980	mail	3	continue	
Tue	Jan	01	00:02:25	1980	base	7	readicon	
Tue	Jan	01	00:02:33	1980	read	1	Ready	read
Tue	Jan	01	00:02:39	1980	read	1	continue	
Tue	Jan	01	00:02:43	1980	read	2	continue	
Tue	Jan	01	00:02:51	1980	base	8	error6	
Tue	Jan	01	00:02:54	1980	base	8	fileicon	
Tue	Jan	01	00:03:01	1980	file	2	Ready	file
Tue	Jan	01	00:03:05	1980	file	2	continue	
Tue	Jan	01	00:03:09	1980	file	4	continue	
Tue	Jan	01	00:03:19	1980	newbase	4	mailicon	
Tue	Jan	01	00:05:45	1980	on	10	fileicon	
Tue	Jan	01	00:07:41	1980	on	12	GoodWork	
Tue	Jan	01	00:09:36	1980	on	11	readicon	
Tue	Jan	01	00:12:02	1980	on	13	mailicon	
Tue	Jan	01	00:12:02	1980	6			
Tue	Jan	01	00:14:54	1980	newbase	11	error1	
Tue	Jan	01	00:16:52	1980	newbase	12	continue	
Tue	Jan	01	00:18:45	1980	newbase	10	mailicon	
Tue	Jan	01	00:20:39	1980	on	14	fileicon	
Tue	Jan	01	00:22:41	1980	on	15	SalesTax	
Tue	Jan	01	00:24:28	1980	on	16	readicon	
Tue	Jan	01	00:27:05	1980	on	17	mailicon	
Tue	Jan	01	00:27:05	1980	9			
Tue	Jan	01	00:29:33	1980	on	17	mailicon	

Appendix 3 Simple Anova Data Analysis

ANOVA of RLerrors to Treatment

```
MTB > anova c5 = c7;
SUBC> means c7.

Factor      Type  Levels Values
Treatmnt   fixed       5      1      2      3      4      5

Analysis of Variance for RLerrors

Source       DF          SS           MS         F        P
Treatmnt      4      43.000       10.750      2.86    0.029
Error        75     282.188        3.763
Total        79     325.188


    MEANS

 Treatmnt     N  RLerrors
        1    16    2.6250
        2    16    3.8750
        3    16    2.7500
        4    16    2.6250
        5    16    1.5625
```

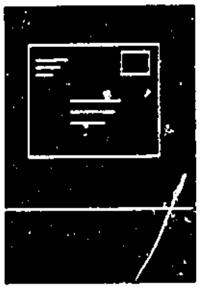
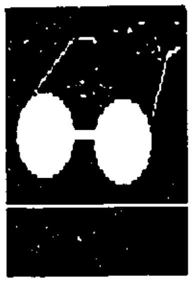
Appendix 4 Study treatment T5
Initial Learning Domain Screens



Super Electronic Mail System

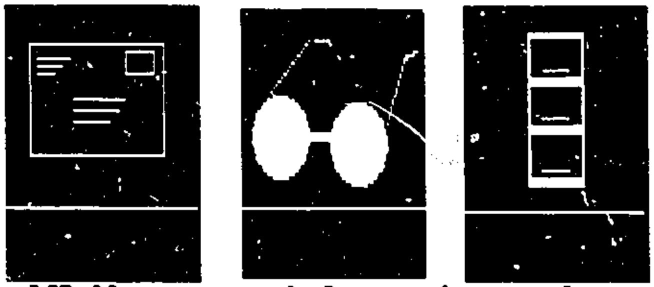
Welcome to the new Super Electronic Mail System.
We will now help you learn to use the system.
Each of the pictures above represents a function of the system. We will explain each function to you and then ask you to try to use the function. But first, a few basics. You activate a function by moving the arrow on the screen to the function to be activated. You move the arrow by moving the mouse device, which is beside the keyboard on the table. Once the arrow is in the position you want, then push the left button on the mouse to tell the computer to "ACTIVATE" the function. Lets practice by activating the "CONTINUE" box below.

continue

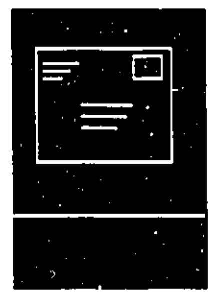
  

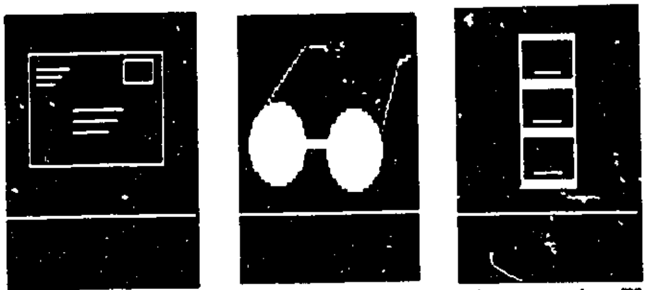
Excellent. Now one more time for good measure.

continue



OK. Now some information on how the Super Electronic Mail System works. We will describe each of the three functions shown above. Activate the first function picture... the one above that looks like this...





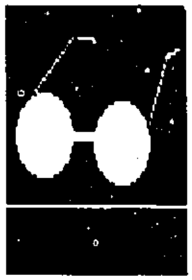
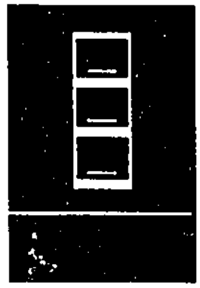
Correct. You have the mouse down pat. This function tells the computer that you would like to use the Super Electronic Mail System to do something with the mail. Perhaps you would like to read new mail, or old mail previously received. Perhaps you would like to send some mail. Whatever. This function simply gets the mail system going, and tells the computer that you wish to do "something" with the mail system.

At this time, we would like you to come up with your own name for this function. It should be only one word long. When you are ready, please activate the "ready" box below left with your mouse. You will be prompted to enter the name by using the keyboard. You can use the arrow keys to change your mind or make corrections. Activate the continue box below right with the mouse when you have finished this task.

my mail

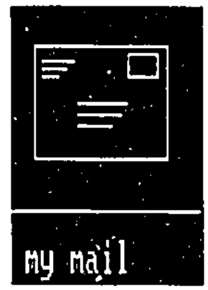
Ready

continue

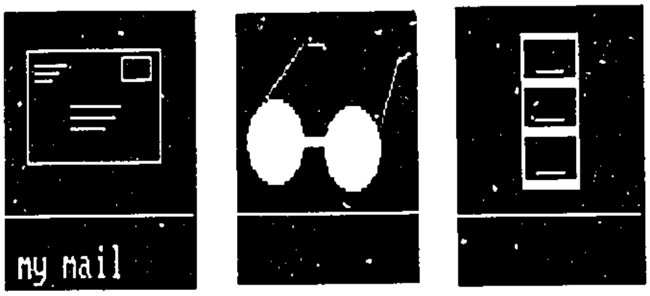
Right. From now on we will call the function... my mail

From now on, the picture of the function will appear with its name shown below:

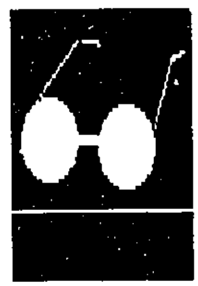


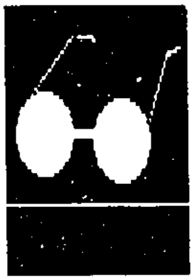
Please activate the continue box below.

continue



Now, please activate the second function picture
the one above that looks like this...





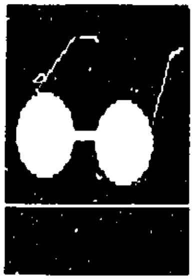
Correct. This function lets you view a piece of mail. The mail may be new mail, or mail previously received and stored.

At this time, we would like you to come up with your own name for this function. It should be only one word long. When you are ready, please activate the "ready" box below left with your mouse. You will be prompted to enter the name by using the keyboard. You can use the arrow keys to change your mind or make corrections. Activate the continue box below right with the mouse when you have finished this task.

view

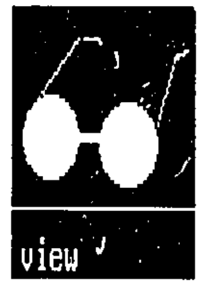
Ready

continue



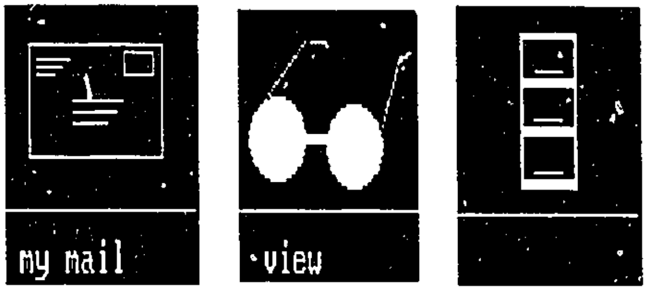
Right. From now on we will call the function... view

From now on, the picture of the function will appear with its name shown below:

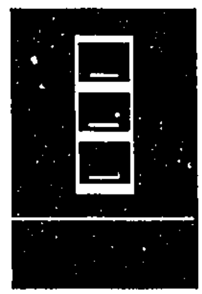


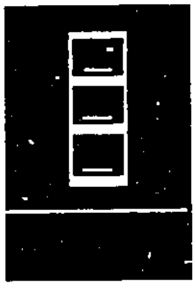
Please activate the continue box below.

continue



Now, please activate the third function picture...
the one above that looks like this...





Correct. This function allows you to store some new mail, or retrieve some old mail. You use this function whenever you wish to store new mail, or retrieve old mail for review.

At this time, we would like you to please come up with a name for this function. It should be only one word long. When you are ready, please activate the "ready" box below left with your mouse. You will be prompted to enter the name by using the keyboard. You can use the arrow keys to change your mind or make corrections. Activate the continue box below right with the mouse when you have finished this task.

cabinet

Ready

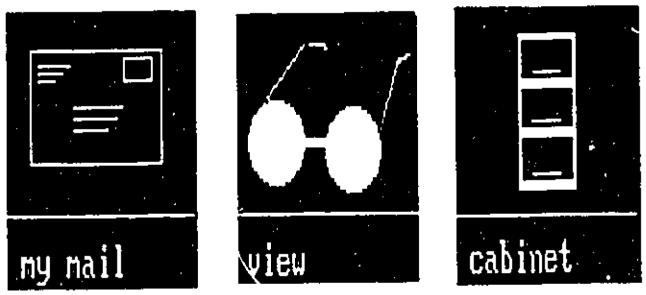
continue

Right. From now on we will call the function...cabinet

From now on, the picture of the function will appear with its name shown below:

Please activate the continue box below.

continue

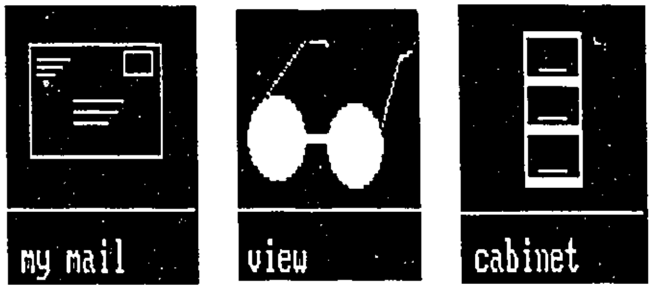


Now we will use the system to perform a task.

Your task is:

1) Activate The Super Electronic Mail System.
2) Retrieve the letter called GoodWork.
3) Read the letter and find the secret code number.
4) Deactivate (turn off) the Super Electronic Mail System.
5) Type the secret code number into the computer when asked.

We will do the task together. Do #1 above now by activating the function that turns on the Super Electronic Mail System by clicking it with the mouse.

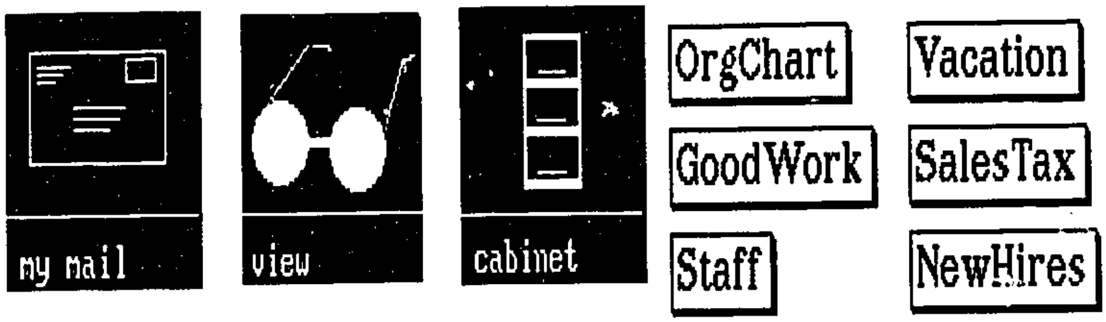


Good. You will notice that the screen colour has turned blue indicating that the Super Electronic Mail System is ready to work.

Recall your task is:

1) Activate The Super Electronic Mail System.
2) Retrieve the letter called GoodWork.
3) Read the letter and find the secret code number.
4) Deactivate (turn off) the Super Electronic Mail System.
5) Type the secret code number into the computer when asked.

Do #2 above now by activating the function used to store and retrieve letters previously received with the mouse.

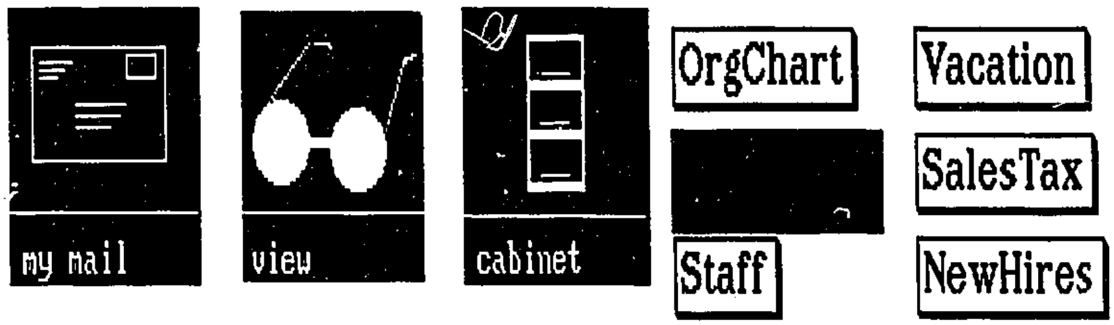


Good. You will notice that the names of the letters that can be retrieved now appear beside the icon. You can select one by clicking the mouse within the appropriate box. You are not quite finished task #2.

Recall your task is:

1) Activate The Super Electronic Mail System.
2) Retrieve the letter called GoodWork.
3) Read the letter and find the secret code number.
4) Deactivate (turn off) the Super Electronic Mail System.
5) Type the secret code number into the computer when asked.

Complete #2 above now by selecting the letter you want by clicking the name of the letter with the mouse.



Good. You will notice that the letter you selected is now highlighted in light blue. Now you can activate the function that lets you view the letter.

Recall your task is:

- [x] 1) Activate The Super Electronic Mail System.
- [x] 2) Retrieve the letter called GoodWork.
- 3) Read the letter and find the secret code number.
- 4) Deactivate (turn off) the Super Electronic Mail System.
- 5) Type the secret code number into the computer when asked.

Complete #3 above now by activating the function that lets you view the letter you have selected, and find the secret code.

File name: GoodWork F1=Help

From: Michael Murphy January 4, 1989
Computer Department

To : All Departments

Subject: New Secret Code

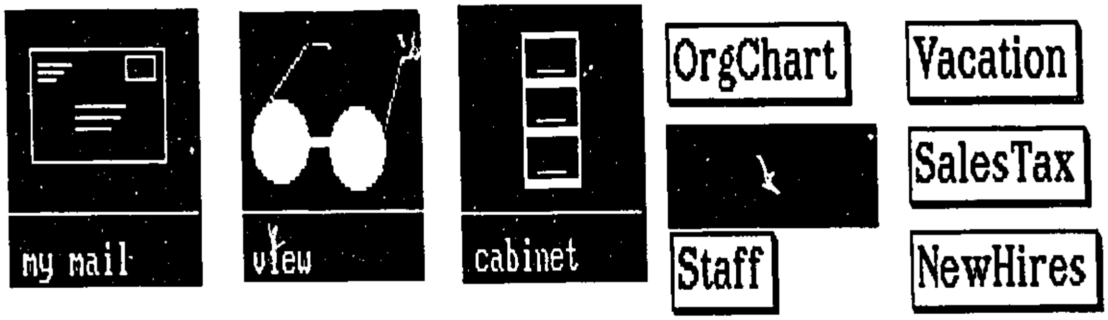
Please note that the new Secret Code Number is 6.

Try to remember this secret code because I will ask you to type it into the computer later.

Now that you know the Secret Code is 6 you can leave this letter.

Thank You

****
PRESS the key F3 to leave this Letter (Located at the top of the Keyboard)

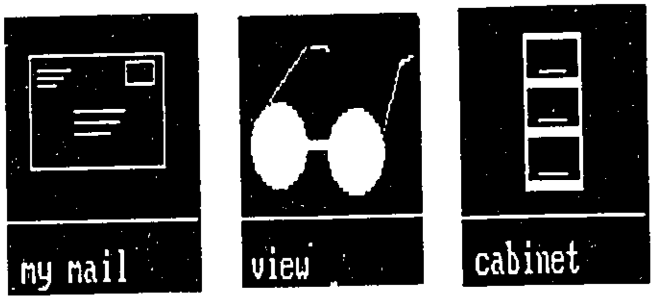


Excellent. Now turn off the Super Electronic Mail System by activating the same function you used to turn it on.

Recall your task is:

- [x] 1) Activate The Super Electronic Mail System.
- [x] 2) Retrieve the letter called GoodWork.
- [x] 3) Read the letter and find the secret code number.
- 4) Deactivate (turn off) the Super Electronic Mail System.
- 5) Type the secret code number into the computer when asked.

Complete #4 above now by activating the function that lets you turn on or turn off the Super Electronic Mail System.

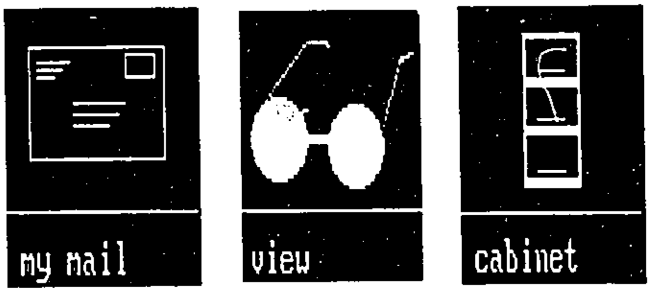


Very Good. Notice the screen is now black, indicating that the mail system is turned off. You are almost finished your task.

Recall your task is:

☑ 1) Activate The Super Electronic Mail System.
☑ 2) Retrieve the memorandum called GoodWork.
☑ 3) Read the memorandum and find the secret code number.
☑ 4) Deactivate (turn off) the Super Electronic Mail System.
5) Type the secret code number into the computer when asked.

Complete #5 above now by typing in the secret code number.
Click the mouse button when you are finished.

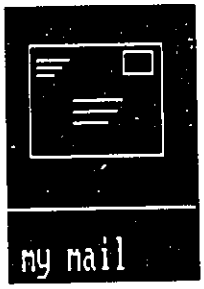


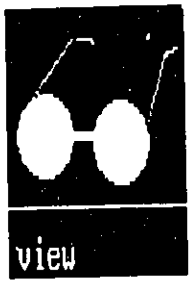
Excellent!

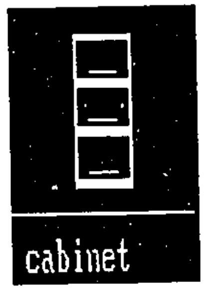
You have completed the first task.

Now, we would like you to complete a similar task, without as much help from the computer.

Please activate the continue box to the right with the mouse. continue





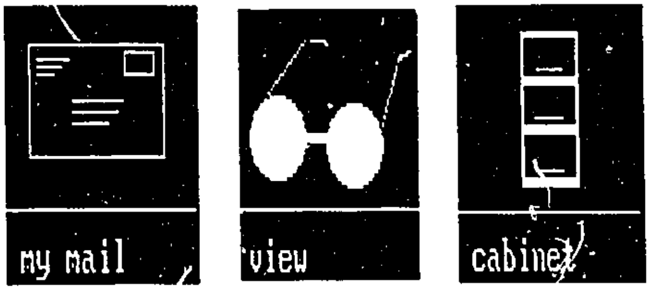


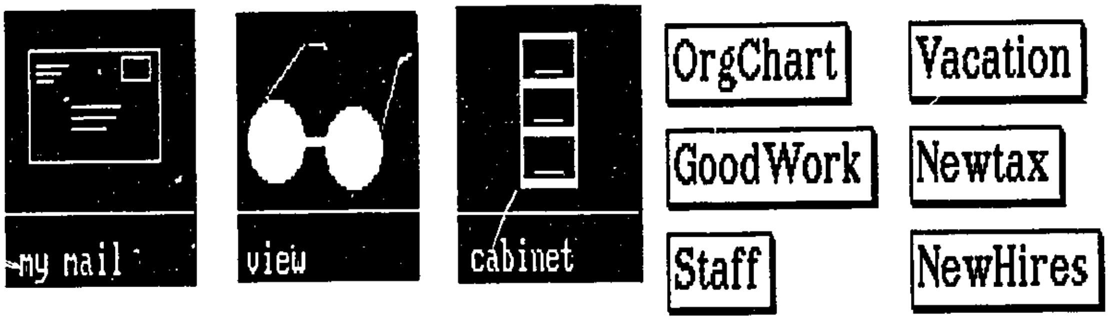
Now you will use the system to perform a task. You are a manager in a company in charge of contract negotiations. You were notified in a letter that the Federal Sales Tax Rate has again changed, but you do not remember what the rate is. You wish to retrieve this letter, which you named NewTax.
Your task is:

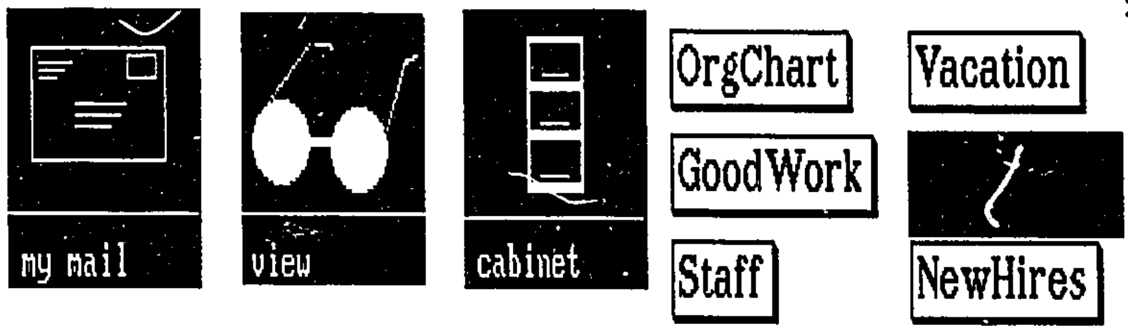
Find the new sales tax rate.

Type the sales tax rate into the computer when asked.

You will do the task on your own this time. You will not be prompted by the system. Please start now.


my mail
view
cabinet


my mail
view
cabinet
OrgChart
Vacation
GoodWork
Newtax
Staff
NewHires


my mail
view
cabinet
OrgChart
Vacation
GoodWork
Staff
NewHires

File name: SalesTax

F1=Help

From: Harry Bland
Accounting Department

January 4, 1989

To : All Departments

Subject: New Federal Sales Tax Rate

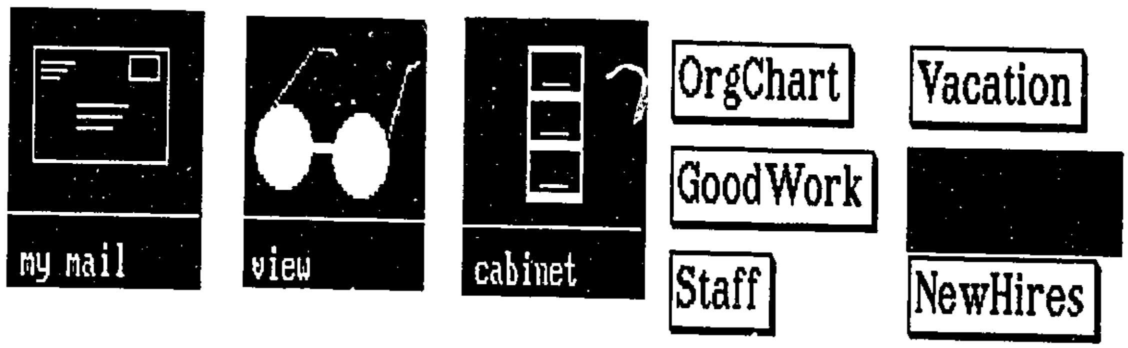
Please note that the proposed new Federal Sales Tax rate has changed and is now 7% effective January 1, 1991.

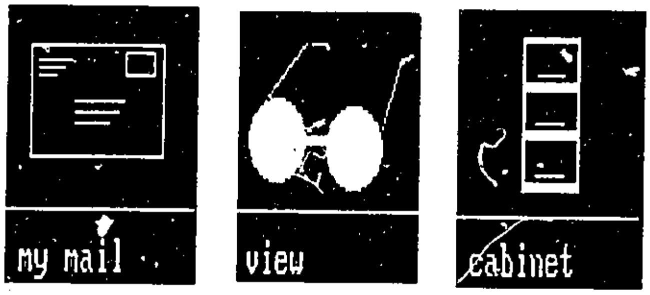
Please use this rate in all future contract negotiations.

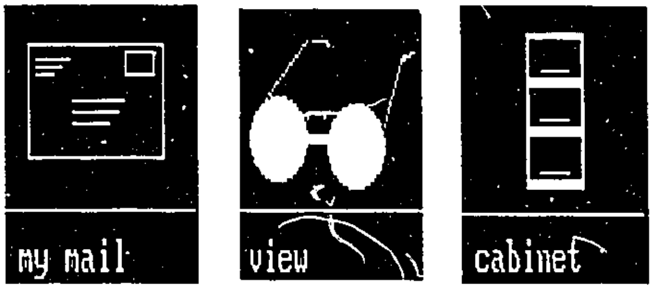
Additional information can be obtained from the accounting department.

Thank You

****
PRESS the key F3 to leave this Letter (Located at the top of the Keyboard)


my mail
view
cabinet
OrgChart
GoodWork
Staff
Vacation
NewHires


my mail
view
cabinet



Excellent!

You have completed the task.

The session is now finished.
Thank you very much for your help.
If you would like the results of the
learning styles test, please ask.