A Pragmatic Investigation of Ease of Use of Business Software

Jennifer D. E. Thomas

A Thesis

in

the Faculty

of

Commerce and Administration

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

March 1994

ISBN 0-315-90887-4

Dedicated to:

My Heavenly Father,
who makes all things possible,

and

My Grandmother,
who gave me the gift of knowing the love of my Heavenly Father

Abstract

A Pragmatic Investigation of Ease of Use of Business Software

Jennifer D. E. Thomas, Ph.D. Administration
Concordia University, 1994

Ease of use is an important concept in software development, selection and application. The literature suggests that ease of use comprises several learning components including: user performance speed, memory, effort, and psychological comfort. Therefore, the extent to which a package design supports these learning dimensions can be expected to impact ease of use. Two exploratory studies examined several factors believed to influence ease of use. Study 1 sought to ascertain expert consensus on the importance of the software features and the learning dimensions that affect ease of use. Study 2 examined users and their actual use of packages to determine the effect package differences, experience differences, and other user characteristics have on measures relating to ease of use.

The results indicate that experts consider command structures and manuals to be the most important features of ease of use and that they consider the learning dimensions to be of equal importance. In addition, the findings show that package characteristics, user experience and other user characteristics--such as perceived quantitative competence, computer anxiety and gender--while having a statistically significant impact on ease of use, were low in terms of predictive power. From a perceptual point of view, the greater the number of package attributes perceived by users to be supporting speed, memory, effort and comfort, the better were the users' performance and perceived comfort with the packages. On the other hand, in actual use situations, the package with the best performance time scores (speed) did not have the best error and help call scores (memory, effort) or perceived comfort ratings (comfort) compared to the other treatments. Over time, however, the results tended to converge, with the best package having the best scores on all measures.

Several implications are suggested by the findings. First, it may be necessary to consider objective and subjective measures of ease of use separately. This may help in gaining a better understanding of how the learning dimensions--speed, memory, effort and comfort--relate to each other and to the construct, ease of use. A second research implication is that ease of use may be unrelated to the learning dimensions which were identified. Differences among users, other than those analysed (e.g. cognitive style), might be a third possibility and may, in reality, account for a substantial portion of the variance. This would suggest that package design should give greater consideration to these other differences than is currently the case. Future research should address these possibilities.

ACKNOWLEDGEMENTS

To Dr. Peter Falcon Wade, I would like to express my gratitude and appreciation for having been my mentor over these many years, beginning with the supervision of my M.B.A. research paper. It was due to his faith and confidence in me, that I had the nerve to consider the possibilites of my doing a Ph.D. I have a great deal of respect for the quantity and quality of ideas which he has shared generously and unreservedly with me and which have challenged me to expand my own thoughts and ideas. His years of experience and his practical approach to research and academia has been a great asset. I have particularily appreciated his integrity towards me

To Dr. Dennis Kira, I would like to say thanks for having so readily agreed to serve on my committee and fill the space created, after one of the original members left the university. I appreciated the quiet support he offered, allowing me to bounce ideas off in a non-threatening, friendly, atmosphere. He also kindly offered financial assistance needed for the research assistants who monitored the computer labs during the running of the experiments.

To Dr. Gary Boyd, I would like to say thanks, for also agreeing to serve on my committee, busy as he is, after another member of my committee, this time the Minor area member, left the university. He supervised my second Theory Paper and in the process of this work, I developed a great admiration for his eclectic mind, and I appreciated the intellectual stimulation our encounters afforded.

To Dr. Jesus Vasquez-Abad, I express my appreciation for serving as the Internal-External member of my committee. Although we met infrequently, and mostly by phone, his knowledge of the area and his suggestions were invaluable. His friendliness and good-natured support were felt and welcomed.

To Dr. Thomas Jerome Tomberlin, I express my eternal gratitude for all he has been instrumental in teaching me, academically and otherwise. Though not a member of my committee, he devoted many hours of his precious time and patience to assisting me with the statistical analyses used in the thesis. He also knew and offered a great deal of friendship and encouragement whenever I was in need of it. Thank you.

To Drs. Dennis Kira, Danielle Morin, Fassil Nebebe, and Steve Kepecki and Johanne Thiffault, I am very grateful, for their cooperation in agreeing to compensate students in their classes, who volunteered for my experiments, with an assignment remission.

To all those those who did take part in the experiments, the survey and the evaluations of the packages, I say thank you.

To Iona Farrell, the Computing Services Department, and the Audio-Visual Department, my heartfelt thanks. Without them, I would not have been able to run the many experiments I did, even videotaping some of the sessions. Iona was very accommodating and good-natured about my disrupting her schedules and making demands on her and her personnel.

To all of the members of the Decision Sciences and MIS Department, Faculty and Staff, I wish to say thank you for the support and comraderie throughout this arduous process. It made it that much easier to bear.

I would also like to thank Dr. de Brentani for her assistance in getting me over the finish line and for the helpful suggestions she made.

I would like to thank my sister Andrea Thomas, Jia-Lin Xie and Tammy Booth for acting as research assistants and assisting in monitoring the computer labs during the conduct of the experiments. Each willingly volunteered without knowing if there was any remuneration for the task. Fortunately, a token gesture for their efforts was kindly provided by Dr. Dennis Kira and by funds from the office of the Director of the Ph.D. and Research Program. I am also grateful to Francois Reindeau for his painstaking data entry.

To all my various and sundry friends too numerous to mention, but who know who they are, I say thank you for all the prayers, support and patience through all this. I could not have done it without you. Particular mention is due to Lilyclaire Bellamy, who as a testiment to her friendship, was willing to forego precious sleep during her vacation in Canada to assist in the arduous task of entering countless references. Jai-Lin Xie must also be mentioned, having shared my office for 5 years and all the pain, laughter, friendship and warmth that comes from a shared experience such as this.

To my immediate and extended family, last but not least, I give my undying gratitude. My parents, Yvonne and Geoffrey Thomas have sacrificed all their lives so that I could attain to the highest levels I was capable of reaching. Their love and respect for each other, and for all their family and friends have been a model for me. My sister Andrea, spent her precious time helping with the boring task of putting together experiments, way into the wee hours of the morning, and helped with manning the computer labs for the experiments. She also generously and effectively acted as editor of the thesis. Her love and laughter kept life in perspective.

To my Grandmother, Kathleen Barnett, words are inadequate to express the love and admiration I have for her. Had she lived in another time she would have done her own Ph.D. She has been a source of great inspiration and teaching. She has overcome so much in her life, polio being just one of the many obstacles, without any bitterness or regret, only a continual deepening of love, service and honour of God and His creation. She is a model for all humanity.

To all my extended family, I give thanks and prayers. I have truly been graced by having been born into such a warm, loving, and supportive family.

Preface

"...for the time being we should be delighted to have some methods that, even at a rather approximate level of precision, and even if they require a component of human judgment, are capable of helping us to evaluate one design versus another."
(Landauer,p.14, in Interfacing Thought, Carroll, 1989)

The evolution in computer technology has permitted a movement from information systems developed on mainframe computers by technical users to information systems developed on micro-computers by non-technical users. This shift has been facilitated by the ever-increasingly sophisticated software emerging on the market. These software address a number of business needs, permitting the manipulation of numbers, text, graphics, et cetera, and purport to be easy to use and learn in relatively short time frames. In essence, these software fall into two broad categories:

1) **specific-purpose software** - packages designed to perform a specific function. These include off-the-shelf packages and custom-built packages for specific applications, such as payroll processing, et cetera;

2) **general-purpose software** - packages designed to perform general purpose functions. These include spreadsheets, database packages, et cetera. and special subject packages for Statistics, Computer-Assisted Design, Wordprocessing, et cetera.

In a competitive market, the aim of product research and development is the quest for, and production of, a 'better' package. Manufacturers vie to gain and maintain some competitive edge in a highly competitive and ever-evolving industry. This concept of what defines a package as 'better' than another is by no means clear-cut. The complexity of the interplay between and among design and assistance characteristics a: d user characteristics and needs, leads to less than obvious choices. It would seem that where the marketing of business software packages is concerned, the often used quote, 'build a better mouse-trap and the world will beat a path to your door', does not necessarily seem to hold. There appears to be a marked resistance to change, as witnessed by the entrenchment of the Lotus spreadsheet for many years, even with the availabilty of other superior products. But are they superior? On what basis can this be asserted? Is it strictly based on user preference, or do certain combinations of factors dictate superiority? Is there enough differentiation across packages for users to appreciate a real or perceived difference? Or, are packages so different that transfer of learning from one package to another is minimal, thereby creating a disincentive to switch? Are packages not as easy to learn and use as claimed by the developers, so that once one learning hurdle is scaled, users are unwilling to face another? All of these questions are as yet unanswered and need to be, both for business and academic purposes. Seymour (1991,p.87) has this to say on the subject,

> "Even if the new product is demonstrably better than the old one, lots of users just won't change...Those of us who find PC's innately interesting and challenging too often forget that most PC users fall into a different category. They don't want a graphing program; they want graphs. Give them a way to get graphs - let them work their way up the learning curve; let them build up a file of old graphs they can keep reusing and also adapt as templates for new work; let them get comfortable with a graphing package - and you're going to have a hell of a time getting them to switch...dark clouds of an economic slowdown are on the horizon, corporations are thinking twice about moving their users to new software packages. The old 'good enough' rule is at work here...Is the gain in terms of net contribution to the bottom line so great as to justify moving to a package...?"

The importance of having a basis on which to make that initial choice, then, becomes very critical in the light of these obstacles to change, and in light of the fact that, in many cases, the person using the software is not the one assessing and choosing the package. A method by which this can be achieved would certainly be appreciated by those who must make these choices and purchase packages, whether for personal use or for use by others.

Aspects of the design of a system can be said to address two requirements - system functionality, and user performance. Those features designed to address user performance must be shown to have positive impacts on factors which contribute to ease of use and learning, as well as user satisfaction, in order to be considered 'good' features. As proposed in this thesis, the learning factors that these features should support are performance speed, memory, mental effort, and psychological comfort. A number of specific design and assistance characteristics, many of which accommodate user characteristics, are generally considered to be important ingredients of a package, engendering it with these qualities of ease of use and learning. The inclusion or exclusion of a particular design characteristic has tended to evolve over time, through trial and error, rather than based on a theoretical foundation. The assumption has been that this is an objective assessment, amenable to the establishment of design standards. The purpose of this research project is to test this assumption and the impact the results have for developing tools and techniques for evaluation. It will do this by suggesting:

1) that the merits of the inclusion or exclusion of a particular design or assistance feature may be agreed on by experts, but, in fact, the degree to which features are deemed 'good' in a particular package may differ for different classes of users, thereby producing different performance and perception results;

2) that assessment of design and assistance features included in a package should be based in learning theory, that is, on the extent of their contribution to the major learning factors of speed, memory, effort, comfort. On this basis, packages which include more features supporting these learning factors may be said to be 'better' than those offering less support;

3) that expert designers/users and novice users of a package will favour packages which support different dimensions of learning, that is, their criteria for ease of use and learning will fall under different dimensions. In particular, novices will favour memory and comfort support because of unfamiliarity with, and anxiety resulting from, use of the system, while expert designers and users will favour speed and effort support. Expert designers and users already familiar with a package, or a like package, are interested in getting the job done as quickly and effortlessly as possible.

An evaluation which matches design and assistance features with their effect on the learning dimensions, speed, memory, effort and comfort, may be feasible. It may be possible to assess 'quality' by the extent to which features included in a general-purpose package support these dimensions, and whether this can be asserted objectively or subjectively. Further, the accordance these results have with objective performance and subjective preference measures can, then, be assessed. As Carroll and Thomas, cited in Davis (1989,p.323) state,

> "Although objective ease of use is clearly relevant to user performance given the system is used, subjective ease of use is more relevant to the users' decision whether or not to use the system and may not agree with the objective measures."

In this thesis, we look at some of these issues, taking a perspective which suggests that **ease of use** may be comprised of the learning dimensions, **speed, memory, effort** and **comfort**. The **comfort** dimension, we relate to the perceptions of, or liking for, the package as opposed to subjective, or perceived **ease of use**, of the package.

Table of Contents

Chapter 5 - RESULTS, ANALYSES AND FINDINGS

List of Figures

List of Tables

Appendices

Chapter 1 - INTRODUCTION

The business place abounds with commercial software touted as being able to provide the solution to all types of problems likely to be encountered in the ongoing process of doing business. The available selection includes spreadsheets for manipulating numerical data; database packages for storage, retrieval and manipulation of banks of data; wordprocessors for storage, retrieval and manipulation of text; statistical packages for more complex analyses of data and; numerous others. These packages can generally be separated into two classes:

1) **Specific-purpose software**. These packages are designed to perform a specific function, and include off-the-shelf and custom-built packages for specific applications, such as for payroll processing, inventory management, et cetera. These are sometimes referred to as "canned" packages;

2) **General-purpose software**. The interest of this thesis lies in this class of package, packages which are designed to perform general purpose functions, and include spreadsheets, database packages, and special subject packages, such as Statistics, Computer-Assisted Design, Wordprocessing.

This second class of packages purports to offer the user decision support facilities which maintain respect for the user's expertise and autonomy, by allowing the user to retain control of the applications developed and the analyses performed. In other words, the user is not offered a solution, as is done by an expert system, other than that offered by templates or command files, nor restricted to specific input as is usual in "canned" packages, but instead, generates his own solution(s) by making use of these tools to create his own environment in which to explore particular decision making activities. All this is expected to be possible, without the highly sophisticated programming and systems expertise and skills usually associated with development of information system aids. This has been facilitated by the use of fourth generation languages, known as 4GL, which permit users to interact with the system using English-like terminology and minimal programming skills. This has lead to the concept of End User Computing (EUC) or User Developed Applications (UDA), which is defined as,

> "Individuals with little or no formal Data Processing training...developing and using their own computer-based applications." (Rivard and Huff, 1985, p.89).

While the term End User Computing has taken on a number of meanings, from simple application development to complex, full-fledged system design by users, (Rockart and Flannery, 1983), the term is used here in a more limited context (Doll and Torkzadeh, 1988). The user is neither passive, such as a data input clerk or an executive perusing output to take the pulse on company happenings, nor active, in the sense of designing systems, such as a systems analyst. The user is, instead, interacting with software which facilitates data input, manipulation and interpretation of output, to assist in a decision making context, much as a manager, for example, might do. A likely scenario would see a manager access data and information from a database and/or knowledge base, perform analyses and projections on this data using a spreadsheet and statistical package or other modelling tools, and then use a wordprocessor to write the final report of recommendations suggested by this analysis.

The utopia described for this second class of package has not been fully realized. To make appropriate use of any tool requires, first of all, that the tool be needed, secondly, that the tool be appropriate and adequate for the task to be performed, and thirdly, that adequate training be provided. It is important to remember that software cannot, of itself, replace the user's required knowledge of the job function, nor his understanding of the role the tool might play with respect to the task to be accomplished. It is simply a tool. As yet, it does not compensate for inadequacies in the user and, in fact, may magnify them. (Ramaprasad, 1987). The user's facility with translating his expertise through this medium will determine the extent to which successful use is made of this tool. This implies training, not only in this tool but also in other tools needed to operate it, in this case, the computer and its particular operating system. The difficulties in mastering these tools can be attested to by the myriad of users, including this author, who have experienced innumerable frustrations attempting to learn to use computers and accompanying software packages. (Carroll and Mack, 1984).

Aspects of the design of a package can be said to address two requirements - system functionality, and user performance. We suggest that features designed to facilitate user performance must be shown to have positive impacts on factors which contribute to ease of use and learning, as well as user satisfaction, in order to be considered 'good' features. A number of specific design characteristics, many of which accommodate user characteristics, are generally considered to be important ingredients of a package, imbuing it with qualities of ease of use and learning. Most important among them, are the assistance and help functions provided to promote learning of the package. There is little consensus in the research literature, however, regarding the effect that the different design and assistance features, and combinations thereof, have on user performance and preference. There is some indication that different access methods, command versus menu, for instance, influence user performance. (Relles and Price, 1981). Also, the form and content of assistance and help functions is deemed important, but the variables involved are not clearly understood. (Shneiderman, 1987; Houghton, Jr., 1984).

Experience-level differences also seem pertinent. Shneiderman (1987), for example, distinguishes three classes of software users, novice, intermittent and experienced, each of whom requires correspondingly different support, and which are not static over time. (This is discussed more fully in Chapter 2.D.). Each of these three classes of users may also have novice, intermediate, and experienced, knowledge of the software package subject matter, as well as of their particular job function. (Carey, T., 1982; Martin, 1989, 1986). For instance, giving a manager a statistical package and training her in the mechanical use of the package, with the expectation of improving the quality of the work produced, is pointless if she does not understand the underlying principles of statistics and how it can be applied in his particular situation to assist his work. This aspect has, hitherto, not been addressed as part of business software design, though it may be found to be significant in the appropriate application of the software to successful decision making. (Thomas,J.D.E., 1989)

Users also come with their own individual cognitive traits which may affect their learning and use of software, as Ambardar (1984) and others have suggested. Sein,et.al. (1987), for instance, contend that if initial training and continued use of information systems is to be effective, the user's individual characteristics, such as learning style, motivation for using the package, should be determined prior to training, and the training method selected in congruence with these.

A survey conducted by Nelson and Cheney (1987a) indicates that self-training was the predominant medium used to acquire software familiarity, and that managers generally rate this training as being only slight to moderate in value. The average time spent, over the user's employment period (average period of employment not given), on interactive training manuals was 7.8 hours, compared to the average use of the resident expert which was 104.8 hours, while a large group (74 out of 100 managers) had no experience with any help component. A study by Carroll and Mack (1984) found that, indeed, self-study learning using online assistance was not effective in the learning of wordprocessing. After twelve hours of self-study learning, of the ten subjects who took part in their experiment, none were able to transfer what was learnt to a new task without making serious errors. One may ask how this might be rectified and whether improvements in the current design of online assistance has done so. From this author's own casual observations of users, it still seems that users tend to be exploratory, active learners of software packages, who resist preliminary assistance from manuals or tutorials. Serious problems are posed for software designers if this is, indeed, the case.

Evidently, training costs will escalate when user resistance, created by improper or inappropriate use, is generated by software purchased to improve productivity. Gewirtz (1988) gives rates of between $12 and $25 U.S., per hour per person, charged by training consultants, and between $75 and $100, per person, for full-day community college courses. These are high costs to bear for training which ends up being unused or under-utilized. One of the major reasons for this may be the due to package designs which make them difficult to learn and use, one cause being lack of consistency within and between package designs.

To date, little consistency exists in command naming and operating conventions across and within commercial software packages (Relles, et. al., 1981; Sein, et.al., 1987), except in-so-far-as ESC to undo previous commands, and the use of Ctrl Home and Ctrl End in some editors to manoeuvre between the top and bottom of documents, seem to be emerging as standards. Smith and Mosier (1984) established a list of 679 guidelines for designing the user interface. However, in a survey conducted by them, respondents reported that although they found the guidelines useful and did use them to varying degrees, they found them difficult to apply in practice. In fact, fifty percent of the managers and software designers only skimmed the guidelines, and software designers found the guidelines less useful than other respondents. One of the needs identified by the respondents, but not addressed by the guidelines, was guidelines for "text editing, interactive graphs, data bases, multi-window displays, and mental models." (Mosier and Smith, 1986, p.44). These guidelines, then, have not found wide acceptance and application in the commercial development of general purpose packages. The lack of consistency which still exists would imply that retraining will be necessary if a switch is made to another package. Does this mean that users must remain locked into their original purchases to avoid this?

The current practice in industry seems to be to adopt a standard, (for example, LOTUS 1-2-3,), and stick with it regardless of the features of comparable, competitive products, (for example, Quatro, Pro, which in Gomes, 1992, was reported as being rated superior to LOTUS 1-2-3, which, in addition to the attributes evaluated, scored poorly in ease of use). The question is, why is there such a resistance to new products which perform the same function? One reason, retraining, has already been mentioned, but is this the whole picture? Other determining factors may be at play. Is the choice of a package, and the decision to remain with it, based on the superiority in quality of this package over another? Is it simply a case of following the leader? Is it to do with advertising, the support offered by vendors, fellow colleagues, available books and magazines, available courses? Is it that commitment has been made to other add ons, making conversion and integration with other packages difficult? Is it the initial cost of the package or that of retraining? Is it that packages are so dissimilar that transfer of learning is too horrendous to consider? Or, are they not differentiated enough to convey any appreciable difference warranting switching?

This thesis concerns itself with the fundamental questions of: what makes a package 'better', that is, 'easier to use and learn', than another? What features or combinations of features predicate this assertion? Is this a subjective or objective call? Do users agree on this concept? Or, is it the case that the other factors override any consideration of the merits of the features in the package?

The purpose of this thesis, then, is to conduct an exploratory and holistic study of this concept called **ease of use**, which is understood differently by different researchers and which has hitherto been examined from an experimental/micro perspective. We view **ease of use** as a global concept of user performance with a package and **ease of learning** as a subset of it. Learning, in this case, is not to be understood in the deeper terms used in education. **Ease of use** refers to package usage generally, regardless of level of expertise, while **ease of learning** refers to the transition stages climbed to achieve mastery of a particular operation or set of operations. This is discussed more fully in Chapters 2.A. and 3.

It is also necessary to distinguish between **ease of use/learning** and 'usability'. This latter concept often includes **ease of use/learning** as one of the factors leading to it and not as the outcome being measured. Features beyond those of the package are also often noted as contributing to it, such as Reliability, Installation, Maintenance, Serviceability, Support of Memory and Effort, et cetera. (Holcomb and Tharp, 1991; Shackel, 1990; Nielson, 1992; Karat, et.al., 1992). We limit ourselves to a consideration of the impact of package-specific features on **ease of use** and how this might be tempered by different experience levels and other user characteristics.

We suggest that **ease of use/learning** of business software is determined by the **speed** with which a user is able to perform a task and the **memory**, mental, and psychological **effort** and **comfort** required to perform it, as well as the degree to which learning is retained over time. In our study, we measure **ease of use** in terms of performance time (**speed**) which included time spent making and correcting errors and reading help messages (**memory, effort**). Performance time is also likely to be influenced, to some extent, by subjective perceptions (**comfort**) of the package. These measures of errors and help calls made and users' perceived comfort with the package, therefore, provided insights which helped to explain the performance time results obtained. The methods used to measure these are discussed in Chapter 4.B.8.

Presumably, features included in the design of a software package are there because of their contribution to these **ease of use and learning** outcomes outlined above; however, the relative quality of these may not be the same. To gain some insights into the means by which the assessment of this relative quality between packages could be ascertained, this research had two main aims:

a. determine expert opinion of what makes a package '**easy to use**' and/or preferred over another;

b. test the expert opinion by comparing performance and preference of subjects with differing experience levels, using one of two packages with differing design and assistance features to perform a similar task.

With these in mind, we posed four main questions:

Research Question 1 - **Expert Consensus**

1.1 Is there agreement, among experts in the field, on: (1) the importance of certain Design and Assistance Features in determining **ease of use**, (2) the importance of the Learning Dimensions identified as influencing **ease of use**, (3) the links between features and the learning dimensions they support, and, (4) whether these match expectations which were derived from a review of the literature?

Research Question 2 - **Package Differences**

2.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, as a consequence of differing Design and Assistance Features?

Research Question 3 - **Experience Level Differences**

3.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to differences in experience levels?

Research Question 4 - **Other Factors**

4.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to other factors, such as computer anxiety, gender, and quantitative competence?

We propose our study as a starting point for discussion and for directing research along another perspective, as opposed to providing a definitive answer to the question of **ease of use** of business software. Gaining a deeper understanding of **ease of use** and having some structure by which it may be evaluated is becoming more and more crucial, not only for designers but for practitioners as well. Other authors concur. Adams, et.al. (1992) state,

> "...differences in interfaces tend to be large, and differences in terms of function tend to be small. In such cases ease of use might be more important because it is a primary feature on which the packages can be differentiated. By contrast, it might be argued that the general interface for spreadsheet packages is similar, and therefore spreadsheet packages are not differentiable in terms of ease of use."

We feel our study has contributed towards these aims on a number of fronts:

1) by integrating for the first time two major research bodies, the Learning and Human Factors literatures;

2) by summarizing the many and complex factors impacting **ease of use** into a comprehensive framework;

3) by suggesting a multi-dimensional view of **ease of use** which has the components, **speed, memory, effort** and **comfort**;

4) by validating previous research efforts done at a micro level with the more realistic, generalizable holistic approach;

5) by conducting a field survey of experts which asks for a multi-dimensional assessment of **ease of use** rather than the uni-dimensional approach previously employed in research;

6) by collecting and examining data on the perceived support offered by package features to the learning dimensions, **speed, memory, effort** and **comfort** and its impact on measures relating to **ease of use**;

7) by suggesting a delineation of package features based on those which are integral to package design (Design features) and those which can be 'added-on' externally or internally (Assistance features). In current research, this distinction is not formally specified and online assistance is usually viewed as one feature;

8) by examining the effect of experience with packages of similar and dissimilar function and the extent of this experience on measures relating to **ease of use**;

9) by suggesting an approach to measuring performance time, errors and help collected in an holistic setting which makes exact measurement more difficult than experimental/micro research;

10) by comparing a command structured package with a directed menu structure as opposed to the undirected menu structures usually compared in research;

11) by adding to the store of knowledge on the factors impacting **ease of use**, the results of which provided some support for the validity of our research approach and measures, supported findings of previous experimental research, suggested new areas for research, and illuminated a number of challenges and recommendations for package design.

Chapter 2 which follows, investigates the Human Factors literature and the Learning literature to determine the important factors affecting ease of use and learning of business software. The empirical findings related to these theoretical foundations are also presented; Chapter 3 discusses the scope of the study; Chapter 4 details the research design employed; Chapter 5 presents the results, analyses, and findings of the research; Chapter 6 summarizes the findings in the context of their implications and importance for business and academia, the limitations of the study being also presented, and directions for future research are discussed; Chapter 7 gives the concluding arguments of the thesis.

Chapter 2 - BUSINESS SOFTWARE DESIGN, EVALUATION AND EMPIRICAL FINDINGS

This chapter examines the Human Factors literature and the Learning literature in order to gain an understanding of the concept of ease of use and learning of business software and those factors which contribute to and detract from this. This is done in several parts:

2.A. Definitions of Ease of Use and Ease of Learning of Business Software
2.B. Impediments and Remedies for Ease of Use/Learning of Software
2.C. Framework of Software Ease of Use
2.D. Factors Affecting Software Ease of Use and Empirical Findings
2.E. Methodologies to Study Ease of Use
2.F. Evaluation and Measurement of Software Ease of Use
2.G. Summary of Existing Research Findings
2.H. Limitations of Existing Research

2.A. Definitions of Ease of Use and Ease of Learning of Business Software

It is assumed that various features of a package design are included in the design because of their contribution to necessary goals of functionality and completeness, and crucially, ease of use and learning, which impact user performance with the package and is the concern of this thesis. These terms **ease of use** and **ease of learning** have been defined differently by different researchers.

There is a belief that **easy to use** and **easy to learn** may be opposing concepts. That is, a package which is easy to use may not be easy to learn, and vice versa. Khalifa (1990) suggests that this may be a function of user expertise. He suggests that if a package is easy to learn but not easy to use, experts will not like it, and conversely, if it is easy to use but not easy to learn, novices will not like it. These assertions are based on his definitions of ease of use and ease of learning. **Ease of Use**, in this case, refers to performance at expert levels and is a function of recall, recognition, and cognitive complexity required by the particular package design. **Ease of Learning**, on the other hand, refers to performance at initial use, and is a function of the user's computer experience and the complexity of the skill being acquired. Scriven (1990) also espouses this definition. He classifies tutorial material as part of **Ease of Learning**, and reference material as part of **Ease of Use**.

In contrast, researchers Roberts and Moran (1983) define **Ease of Use** as performance across all levels of expertise, while **Ease of Learning** is a measure of the time taken to progress from novice to expert performance level. Davis (1989, p.325) views **Ease of Learning** as "one substratum of ease of use construct, as opposed to a distinct construct." He found in his research that user perception of ease of use hinged on physical effort, mental effort, and perceptions of how easy a system is to learn, that is, in remembering and supporting memory. Other research also supports this view, demonstrating strong correlations between measures of ease of use and learning, and their interconnectedness for users. (Roberts and Moran, 1983; Whiteside, et. al., 1985; Carroll and Mack, 1984).

This latter definition is the one espoused in this thesis. **Ease of Use** is used to refer to performance at all levels of expertise, and is said to be a function of impact on performance speed, memory load, mental effort, and psychological comfort, and **Ease of Learning** is considered a subset of **Ease of Use** as defined by Davis (1989) above. **Ease of Learning** is not used in the same context understood in education, which is more concerned with the user's grasp of underlying concepts of what is being learnt.

Credence for relating the dimensions of performance speed, memory load, mental effort, and psychological comfort with ease of use is based on the definitions given above and is supported by other researchers as suggested by comments made by Bennett (1983, p. 51),

> "If the equipment gives the highly motivated user a function that is unobtainable in any other way and is perceived to be important, the user may cope with a poorly designed computer interface. Nonetheless, the impact of defects in the design will appear in an analysis of usability as low efficiency, fatigue, and dislike of using the terminal."

Likewise, Scrivens (1990, p. 34) gives Speed, Ease of Use and Support as examples of package 'Dimensions of Merit', as he terms them. **Speed**, in this case, refers to the speed of the program, rather than the speed of the user's ability in accomplishing a task. This latter aspect of speed he includes in **Ease of Use**, namely,

> "the extent to which routine tasks are automated, the extent to which command key combinations are easily reached by hands of all sizes, the avoidance of having to go six menus deep to get to a style parameter, etc., along with the part of the documentation that you turn to after you've learnt the program once, but still need to refresh your memory occasionally. We also include one other matter - the enjoyability of using the program, principally a function of elegance of design.... Pleasure in use is largely a matter of good design."

Support includes vendor support, warranties, upgrades, user groups, newsletters, etc.

As we have seen, a package which allows a task to be accomplished more speedily than another is often equated with being easier to use. In most research studies, performance time is the measure collected and analyzed as a measure of **ease of use**. This is discussed further in Sections D and E of this chapter. A number of factors may, however, affect performance time. It has been well recognized that humans have limited short term memory capacities. The well-known studies of Miller (1956) indicate that humans are unable to process more than between five and nine items concurrently. A package which minimizes this requirement, or alleviates it completely, is likely, then, to be easier to use than one which imposes greater load on this already taxed capacity. Features provided in software which are designed to alleviate this human shortcoming include databases, macros, menu structures, templates. Waern (1985) found that users have a tendency not to read, preferring to rely on memory unless the opportunity for reading and the material to be read requires little effort.

The mental effort or cognitive strain involved in accomplishing a task is likewise expected to contribute to ease of use. Given a choice between equal outcomes but obtainable through unequal effort, most users will opt for the one which requires the least effort if it is known to them. Confirmation for this hypothesis was reported in a recent study by Todd and Benbasat (1991), which examined decision strategies used in light of differing decision aids provided and the number of alternatives to be assessed. It was found that strategies requiring the least effort were the ones adopted in most instances and that these were adapted to the format of the decision tool provided. Users were asked to choose an apartment based on preferences for a number of attributes. It should be noted that memory requirements are also related to mental strain, and that in some instances, greater memory and mental loads will be tolerated if the outcome is sufficiently desired to compensate for the additional burden required to obtain it. This can be noted in users who tolerate less than optimal package designs in order to have their information needs met, or because of pressure from superiors to make use of the product. Psychological comfort is also related to memory and mental strain but also includes the aesthetic appeal and cognitive fit of the package to the user. Ease of use is, therefore, also likely to be affected by this concept. Section D in this chapter discusses some aspects of this concept which have been researched to date.

A number of factors may militate against ease of use and learning of a package. These are discussed in the section which follows.

2.B. Impediments and Remedies for Software Ease of Use/Learning

Learning a software package involves many levels of syntactic and semantic learning: that of the software, the computer and its operating system; that of the package subject matter; and, application of these to the particular problem. There are a number of ways in which learning can be impeded, thereby preventing movement from novice to expert levels of performance. It is also difficult, where complex learning is involved, which characterizes software learning, to accurately measure learning, since it is less amenable to quantitative - recall/recognition measurement than is simple learning.

Mayer (1981,p.121) attributes meaningful learning to:

a) Reception - being attentive to incoming information;
b) Availability - having the necessary prerequisite knowledge in long-term memory to be able to hook this new incoming information;
c) Activation - actively using the prerequisites to hook the new material.

In business, attention is usually divided (Bailey, 1989), and although the user may have some of the prerequisites for learning software, such as command names, he may have fewer of these when it comes to applying the software to accomplishing the tasks of the job function. Current software provides for the activation and availability of meaningful learning by using metaphors in the interface design, such as a desk-drawer filing system to explain databases; icons, such as a trash container to simulate disposal of unwanted files in the Xerox/MacIntosh environment; and, the choice of command names, mnemonics, and images. Examples are also another vehicle but these are usually very simplistic and elementary, making it difficult to see the relevance and applicability to the user's particular domain.

Failure to use software packages, in particular failure to use them effectively, can be attributed to trade-offs made to learning identified by Bigge (1982,p.236) as being blocks to learning in general (see also Guillemette, 1989). These include:

a) **Informational situation** - this depends on the learner's perception of how much more information should be gathered in relation to the strain and the risk involved, and the ability of the software package to assist this information gathering needed for the task solution. This perception may or may not be accurate;

b) **Cognitive Strain** - this refers to the time taken and the quantity of frustrating attempts that are required to arrive at understanding, and how confident the person feels about the solution attained. The major complaint of beginner software users is frustration with not knowing where to start, how to proceed, what it all means. The feeling is one of operating in a vacuum, without knowing where it all leads, and how to get there, nor knowing how long it may take. Trial and error are the usual mode of learning, always unsure of what is going to happen, how to recover from errors or wrong paths, whether certain actions will lead to the desired effects or to undesired erasure, blown disks, jamming, et cetera. (Carroll and Mack, 1984). Work familiarly done by hand is more predictable in terms of what is involved and what can be expected to occur and in what time frame. There is no added dimension of an unknown, unpredictable entity, the learning of which is often perceived as a waste of valuable time when it is already being successfully done by hand (Nickerson, 1981);

c) **Risk** - this refers to the extent to which consequences of actions taken and errors made are disastrous. The more so, the more willing will the individual be to spend more time and effort on data acquisition and analysis, though to some extent individual differences come into play. Some individuals are more risk-taking than others. In business, however, the company's bottom-line, and the individual's continued employment is contingent on good decisions. This may be undermined if a system with which the individual is unfamiliar and uncomfortable is being used, differing from the usual way of 'doing'. However, if the perceived benefits of learning the system outweigh the disadvantages, the likelihood of learning the system improves.

Not only are these factors deterrents to learning, but individual differences in expertise may also impose different requirements for software learning. The user may be expert in some features and functions of the software and entirely novice in others. There may be simultaneous, varying degrees of expertise in the syntactic, semantic and application knowledge of the software, and package subject matter itself. A user may be very versed in syntax but not in the semantics of the software, or vice versa; and likewise, versed or not, in its application to the task at hand; and likewise, in applying the package subject matter, the relative importance of which is not equally constraining. The transition between novice and expert at these various levels needs to be understood.

> "Understanding occurs when we come to see how to use productively, in ways that we care about, a pattern of generalized ideas and supporting facts." (Bigge, 1982, p. 296).

In software use, the problem is one of what is meant by and how to measure understanding and productivity. One can conceivably arrive at a correct answer, using inefficient means, or even incorrect means, never being any the wiser. (Borgman, 1986). It is extremely difficult to assess and prevent negative transfer of learning in business in general, and is especially so in the use of software packages for decision analyses.

The section which follows categorizes the various deterrents to learning found in the learning literature and applies them to the learning of software.

2.B.1. <u>Human Information Processing (HIP) Impediments and Remedies</u>

In the literature on general learning, Rigney (1980, p. 337) identifies a number of impediments and remedies to cognitive processing in humans on conscious and unconscious levels. We apply these to the learning of business software. Other authors also corroborate and enlarge this list. (Bailey, 1989; Guillemette, 1989; Trumbly, 1989; Young, 1989; Van-Lehn, 1988; Jonassen and Hannum, 1987; Harmon, 1987; Borgman, 1986; Weiner, 1986; Carroll and Mack, 1984; Bigge, 1982; Bethke, et. al., 1981; Rigney, 1980; Travers, 1975). These deterrents to learning are described in Tables 1a and 1b, below, and explained in depth in this section. The bracketed notations beside the respective impediments indicate the instructional support implied by them and the learning factors supported as a consequence of providing this support, as envisioned by this author. These impediments touch on a number of factors affecting software ease of use and learning: package design and assistance features, instructional strategy and material, learning environment, user characteristics, such as motivation.

Table 1a

Human Information Processing (HIP) Deterrents and Remedies

a. CONSCIOUS LEVEL

Condition (currently seen in software usage)	Remedy (currently applied in software design)
1. Limited Short Term Memory -inability to manipulate more than 7 $\pm$ 2 items at once (using a software package requires a preponderance of commands and sequences of operations, too numerous to commit to memory)	-live with - rehearsal; serialization of subject matter and processing operations -get around - hierarchical organization of subject matter - chunking, mnemonics, icons (hierarchical menu stuctures; database and knowledge base structures, mnemonics, images, icons)
2. Limited Self-Program Skills -inability to organize appropriate sequence of operations (so many steps involved in performance of one operation, difficult to know/remember what they are)	-teach effective sequencing of processing operations - use heuristics and algorithms, explore alternatives, try different sequences of operations (tutorial exercises and examples)
***3. Limited Self-Monitoring Skills** (meta cognition) -inability to keep one's place in a long sequence of operations, to know when subgoals have been attained, to detect and correct errors and recover from errors by returning to last correct operation or by making quick fixes (many routes to perform one operation, unsure which to take or which took to get there, and how to recover if not the correct one)	-checklists -looking-ahead - prescriptive avoidance of error where most likely to occur -looking-back - postscriptive analysis of errors already committed, maintaining history of processing to current place in sequence (check of illegal entries, valid operations· undo command)
4. Distractibility of Attention incapable of prolonged concentration on any one processing task (unscheduled formal training time, intrusion of regular work activities)	-orienting techniques - points of reference, trace, motivation; simplification; modularization (scheduled training time; save work done to point in time)
5. Motivation (Extrinsic) -learning limited by factors beyond control of learner (voluntary vs involuntary use, need vs convenience, short terms vs. long term use)	-punishment/reward systems (easy to use designs, demonstrable relevance to work, compulsory use)

(* Indicates greatest deterrents to software learning and use, expected by this author)

2.B.1.a. CONSCIOUS HIP Impediments and Remedies

1. Limited Short Term Memory (STM) (structure and depth, command naming,text wording/grouping) (memory)

Research conducted by Miller (1956) indicated that humans have a limited capacity to process more than between five and nine items simultaneously. Further research indicated that this limitation could be alleviated somewhat by grouping items in some recognizable fashion, thereby allowing easy discrimination between items. Rigney (1980) refers to this as "getting-around". As opposed to the traditional approach to learning, which requires rote memorization through drill and practice and serialized learning of subject matter and operations, this approach offers associations or "hooks" on which to attach new material being learned, thereby facilitating future recall.

This approach to a solution has found its way into software design in the form of hierarchical menu structures, database and knowledge base structures, use of mnemonics, images and icons for easy command recognition. (Harmon,1987). There is still more room for additional structure to reduce the number of possible commands, sequences and applications which have to be remembered by the user, and to ensure consistency. This need for consistency in commands and structure within and across software packages is supported by Bigge (1982, p. 244) who states,

> "The more generalized a coding system is, the more useful it is to a learner in that it relieves him of any need to learn and try to remember a great mass of isolated facts."

Travers (1975, p. 143) expresses it more colloquially,

> "If each person formed his own classification system, it would be as confusing as if each were to use his own language."

Yet this is precisely the norm in software design, where each vendor, using menus, groups commands according to his own predilections, and why learning proves to be so frustrating for users. An example of this is the Lotus, menu configuration wherein to change the column-width of one column, the user is required to access the Worksheet menu rather than the Range menu, which would be more conceptually logical and consistent. The other side of the argument, however, is that individual differences may need to be accommodated in customized design.

2. Limited Self-Program Skills (tutorials,examples,traces,subject matter assistance,step-by-step guidance) (effort)

This condition refers to a learner's inability to appropriately organize the sequence of operations required for resolution of the problem. The same task can be formulated differently by different people, with varying degrees of correctness. (Ramaprasad, 1987). The answe suggested is to teach and allow practice of the correct sequences required, making use of heuristics, algorithms, exploration of alternatives and different sequences of operations, thereby learning to differentiate correct from erroneous paths. Using simplified structures and examples, giving correct answers, using counter examples, hypothetical cases and allowing traces are further suggestions found in Collins and Stevens (1982), and Travers (1977). These techniques will assist the mental efforts required by the user in performing his task.

Current software provides tutorial exercises to show examples of limited cases. Users are often not sure of the exact steps used, or to be used, to arrive at the various junctures in these exercises. This makes duplication difficult, especially when the transfer to different tasks is called for. The remedy for this condition may be to provide step-by-step guidance through exercises, showing the most 'likely' or 'best' way to accomplish the task, showing all possible ways, showing counter examples, especially in the case of commands with synonymous names, but which are used differently in the particular package. (Harmon, 1987; Trumbly, 1989). As Carroll and Mack (1984, p. 38) point out, "Learners expect functional consistency from operations with similar names." This is often absent in software packages, compounded by the fact that the same word may be ascribed different meaning by different people depending on the context. (Rubinstein and Hersh, 1984). Anecdotal evidence of this is seen in the frustration of users who see the **PgUp** and **PgDn** keys on the computer keyboard and surprisingly discover that they do not perform these promised functions in the particular package they are using. Some aspects of self-program skill deficiencies may further be alleviated by teaching within the context of the subject matter, such as statistical principles, database management principles, et cetera, thereby providing a base of relevancy, that is, 'hooks' (Young, 1989; Bethke, et. al., 1981).

3. Limited Self-Monitoring Skills (checklists,traces,maps,trees, memory aids,error checking) (memory)

Tied to the human's limited short term memory problems, is that of having to keep track of where one is in a long sequence of operations, to assess the relative correctness or incorrectness of actions taken so far, and to take remedial or preventive action if necessary. Ways of doing this include keeping checklists of usual actions and consequences, and whether these have been executed; 'looking-ahead' to avoid likely errors, and as a reference point, when errors do occur, for where next to proceed and the consequences; 'looking-back' as a postmortem of actions taken, erroneous and otherwise, and their resulting outcomes. This is also useful in diagnosing areas of difficulty and mastery, and where remedial assistance or advanced topics may be required. Travers (1975) further suggests the need for adequate viewing time after feedback in order to absorb the relative correctness of actions compared to a standard. Maps or trees are useful means of monitoring movement. Other beneficial pointers and memory aids include answers to - "Where am I; What can I do here; How did I get here; Where can I go, and how do I get there?" (Nievergelt and Weydert, 1987, p. 438).

As Carroll and Mack (1984) noted in their experiments with users of wordprocessors, users often are unaware of where they are going, how they got there, where next to proceed, or how to get there. So many operations are required to accomplish the task, that it is very easy to get lost in the mayhem and uncertainty. Sterman (1989, p. 321) found his subjects,"insensitive to the feedback from their decisions to the environment", because of the difficulty of attributing cause, given the frequency of occurrence of multiple feedbacks. Currently, the monitoring that is provided to users is in terms of illegal entries in the form of command names, alpha/numeric entries, validity of selected operation permitted at particular junctures. The onus for 'correct' operation and application of the software is entirely in the hands of the user. Some systems allow for error correction and recovery via an 'undo' command or session print out. However, the notion of incorporating checklists, and looking-back traces offers tremendous potential for alleviating this problem for software users. This would provide the positive reinforcement required in learning and so lacking in current packages. These traces could also provide useful insight for researchers into the debate of individualization versus generalization of software design, (Martin and Fuerst, 1987), since individuals' paths could be tracked and analyzed, vis-a-vis novice/expert, psychostructure, and demographic differences. 'Directed' menu structures, that is, those that guide the user through the usual steps required for a particular function, are also another way of ensuring that the user accessess only valid options.

4. Distractibility of Attention (modules,traces,macros,learning times and locales) (memory,comfort)

Humans have a limited capacity for maintaining prolonged, focused attention on any one task. Distractions, boredom, emotional states, all work to dissipate attention. Learning self-discipline is one of the caveats of human development. As Travers (1975, p. 40) puts it,

> "The acquisition of control requires, as most courses on studying habits indicate, living for a time a well-planned life in which certain activities take place on schedule and in an appropriate place."

Motivational factors, such as promotion of interest in the subject matter, the structuring of the subject matter into small, simplified modules not requiring extended periods of learning, are approaches used to return the learner to the place at which attention wandered or the work was abandoned. Other ways to assure attentiveness include private learning rooms or carrells in which learners can work without being disturbed or distracted. There is psychological comfort in knowing that reserved times and locations are designated for uninterrupted learning and that features exist that offer support of memory.

The research into learning curves and forgetting point to the detrimental effects of interrupted learning. Argote, et.al., (1990, p.141), refer to various works in the psychology literature that suggest,

> "if the practising of a task by an individual is interrupted, forgetting occurs (Ebbinghaus, 1885). While interference from other tasks causes forgetting, forgetting occurs when performance is delayed even if there is no interference (Anderson, 1985; Wickelgeen, 1976). When performance is resumed, it is typically inferior to when it was interrupted but superior to when it began initially. (e.g. see Kolers, 1976)."

Evidently, in the business environment work is not likely to proceed in highly conducive, ideal circumstances, nor is learning. Activities get "fitted-in" as time permits. Shuell (1980, p. 297) points out that,

> "There may be times when the desired objective is antagonistic to the learners preferred or optimal style of learning."

This is likely to be prevalent in business because of time constraints, resource constraints - personnel and equipment; working constraints; badly designed software help features - online and offline manuals and tutorial material; with which users must constantly contend (Guillemette, 1989). A study reported in Borgman (1986) indicated that while successful learning of a software package requires devoting longer than thirty minutes, the majority of users are unwilling to give even that amount of time. Nowhere is it more crucial than in the business environment, to be able to leave off work or training and resume at will. Use of features such as looking-back traces described previously, would greatly aid this problem in the business world. Most systems already allow users to resume at the point terminated, but few, if any, offer the facility for reviewing the steps taken to a point in time, except by way of setting the printer in lock step using Ctrl PrtSc. The tradeoff in this case is between excessive paper usage versus disk space usage. The facility for macro creation in some software is one way around this. The 'Remember' command in the $SMART_t$ software facilitates the storage of the sequence of operations performed during the activation of this command. The command is most often used at the upper levels of expertise for macro building, more so than at the novice levels for performance assessment, or for comparison to a standard, expert sequence. Special training rooms and times for learning, as well as off-site training settings are other means conducive to undisturbed learning in the work place.

5. Motivation - Extrinsic (feedback,relevance) (comfort)

Weiner (1986) distinguishes between intrinsic and extrinsic motivation which he asserts affect learning and performance in different ways. Extrinsic motivation includes those factors outside the learner's control which impact learning positively or negatively. This is achieved by way of punishment/reward systems which can either promote or detract from psychological comfort.

In business, motivation for learning a software package is based on a number of factors: whether its use is voluntary or imposed, whether it is needed to perform the job function or simply convenient for doing it, whether its use will be short-term or long-term. Progress is mainly dependent on self-motivation, though corporate culture does have influence on the desire to use and actual use of software (Sein, et. al., 1987). In the absence of human reinforcers to promote this motivation, there is a need to have built into the system reassurances on how one is progressing, and if not, where erring or weak. Rigney (1980, p. 325) poses these questions:

> "1. What is it?
> 2. What should I do about it?
> 3. How do I do it?
> 4. Can I do it?
> 5. How am I doing?
> 6. Am I through?"

This need for a sense of security, knowing how you measure up, what criteria are being used to assess this, and where you are headed, is only weakly provided for in present software design. There is no frame of reference on which to measure or base performance and progress.

Table 1b

Human Information Processing (HIP) Deterrents and Remedies

b. UNCONSCIOUS LEVEL	
Condition (currently seen in software usage)	Remedy (currently applied in software design)
1. Inadequate Basic Processing Skills	
-ineffective processing strategies	-Drill & Practice in effective strategies
(attributing correct semantic meanings to command names or sequences of operations is sometimes difficult)	(trial and error)
2. Inadequate Knowledge Bases	
-deficient accumulated knowledge	-Drill & Practice - analogy, paraphrasing, mental imagery, etc
(contingent on expertise of developer of knowledge base and availability of information)	(taken as given, accessed by trial and error)
***3. Processing Interference**	
-negative transfer of learning	-strengthen desired processes, weaken undesired ones
(unsure of what actions led to what outcomes)	(trial and error, examples)
4. Long Term Memory Retrieval Failures	
-forgetfulness	-frequent and prolonged use
(recall of commands and operations after extended non-use)	(templates, online help)
5. Motivation (Intrinsic)	
-learning arrested by attribution of lack of ability based on prior experience and performance	-change perception to lack of effort or prerequisite knowledge or exposure
(users attribute poor performance to their inability and not to bad design resulting in limited or no use)	(attempts at easy to use designs)
(* Indicates greatest deterrents to software learning and use, expected by this author)	

2.B.1.b. UNCONSCIOUS HIP Impediments and Remedies

1. Inadequate Basic Processing Skills (training,examples,practice) (effort)

As was already stated, the same problem can be formulated differently by different people, with varying degrees of correctness and efficiency, and requiring differing amounts of mental effort. Different people may arrive at the same answer using different means, and one route may be more efficient than the other, though both are effective. Repeated practice over extended periods, in the effective strategies, eventually leads to automatization of the processing operations. The use of examples demonstrating the relative benefits of one route over the other, is also likely to be helpful.

This problem is very acute in using business software. (Carroll and Mack, 1984). Because of the very hit-or-miss ways of learning software, assuming that an association can and has been made between certain actions and certain outcomes (Wittrock, 1974), these are likely to be repeated, even in the light of new evidence which indicates a more optimal route. (Nisbett and Ross, 1980). The tendency is to cling to the tried and true, until such time as the user gains confidence with the package,

> "...people are often, correctly or incorrectly, influenced [in their judgment] by the relative availability of the objects or events, that is, their accessibility in the proces of perception, memory, or construction from imagination." (Ramaprasad, 1987, p. 142).

The antidote would seem to be to prevent these associations from being formed initially, by teaching the "most likely" scenario and sequence of operations for achieving particular tasks. Less efficient routes could also be demonstrated, indicating its deviation from optimality. (VanLehn, 1988; Collins and Stevens, 1982). An example of this is the formatting of a spreadsheet, which can be done either globally, in blocks of rows or columns, or by individual cell. Depending on the extent of the formatting required, one of these options may be more efficient than the other, that is, require fewer key strokes, yet the outcomes may be exact, that is, effective.

2. Inadequate Knowledge Bases (storage,hooks,practice,structure, grouping) (effort,memory)

Differences in learners' backgrounds and experience imply that differences in learners' accumulated stores of knowledge will exist, some more adequate than others. Tools used for assisting the building of these knowledge bases, or the augmenting of existing ones, require drill and practice in developing and making use of meaningful analogies, paraphrasing, mental imagery, et cetera, in order to create 'hooks' on which to hang incoming information. These will enhance memory and retrieval of this store of information. (Norman, 1980). By arranging "concepts to be learned in an order consistent with their structure" as Travers (1975, p. 138) suggests, knowledge bases are more likely to be successfully constructed as we generally learn ideas "in context of other more general ideas".

In business, database and knowledge base facilities can augment this deficiency. However, the relative efficiency and effectiveness with which these are used is suspect since this is very difficult to test and assess. If these tools are used at all, efficient and effective use are assumed. The difficulties come in the assigning of command names themselves, since, as was noted earlier, different words can have the same semantic meaning, and the same word can have different semantic meaning for different people. (Ramaprasad, 1987). This is pertinent whether the user is using an existent, created by another, knowledge base or database, or attempting to create his own. It is also pertinent in the formation of the person's own mental store of knowledge of the workings of the software being acquired, this being dependent on his ability to understand and make the necessary associations.

3. Processing Interference (consistency,feedback,storage,examples, structure,simplify) (memory,effort,speed)

Negative transfer refers to the interference or weakening of correct associations of responses to stimuli in one learning experience as a consequence of a prior learning experience. There are four possible outcomes of transfer of learning between situations (Bigge, 1982, p. 252), as depicted below in Figure 1.

Figure 1 - Transfer of Learning

Performance	**Reinforcement**	**Learning Outcome**
+	+	+ learning
+	-	- does not recognize learning
-	+	- misinterprets learning
-	-	- no learning

For transfer to occur (Bigge, 1982, p. 275),

> "the individual must generalize, i.e. perceive common factors in different situations; they must comprehend the factors as applicable and appropriate to both situations and thereby understand how a generalization can be used; and they must desire to benefit by the sensed commonality."

For this to occur, that is, for there to be positive transfer of learning, Travers (1974) suggests that there must be mastery of what is learned, which is only possible by providing experience with a wide range of problems that differ somewhat from each other, and by emphasizing principles and their application as opposed to just facts, and allowing for practice. If provision for these is too short, then transfer will not occur, if too long, then diminishing returns ensue. Further, unless one can differentiate what the right and wrong actions are, performance will not be improved. (Travers, 1975, p. 74). He states that,

> "There is maximum negative transfer or interference when one learns a particular set of responses on one piece of equipment and then moves to another piece of equipment that requires the opposite set of responses". (Travers, 1977, p. 376)

In switching from one software package to another, the likelihood of the occurrence of maximum negative transfer is quite high. There is still little consistency across packages which sometimes sees words usually ascribed the same semantic meaning being attributed to very remotely related functions, for example, Save, File (Kogan, 1980). The possibilities for defective learning is further increased by the potential for negative reinforcement and for positive reinforcement of negative performance. (The reader is reminded that in the context used here, learning is being used to refer to the ability to obtain a desired outcome rather than to the deeper meaning usually implied in education). In current software design, input errors are screened, but no provision is made for processing or correction of errors. Errors can go undetected because of correct answers materializing despite wrong or weak sequences of operations. It is also possible that the correct answer may never be arrived at because the correct sequences necessary to get there are not known, and the user ends up using the wrong ones. The criticisms levelled at Behaviourists by Gestaltists are also pertinent to current software design. 'Learning' gets measured in terms of proximity to reproduction of predetermined answers, and not by the processes or understanding used in getting to the answers themselves. However, as the Gestaltists point out,

> "For learning to result, the doing must be accompanied by the doer's realization of the consequences of the act." (Bigge, 1982, p. 100).

In using software, there is a lot of room for the user to see desired results actualized from erroneous sequences of executions, and thereby experience negative transfer from confused reinforcement. The occasions for repetitive, reinforced performance are limited in business. Usage is often sporadic, as is the learning leading up to it. Negative reinforcement easily interferes with performance because of attributing erroneous actions and sequences to desired outcomes, which arises from forgetting the exact steps that led to the result. (Carroll and Mack, 1984; Travers, 1975). Carroll and Mack (1984, p. 16) state,

> "People in this situation see many things going on, but they do not know which of these is relevant to their current concerns. Indeed, they do not know if their current concerns are the appropriate concerns for them to have. The learner reads something in the manual, sees something on the display, and must try to connect the two, to integrate, to interpret. It would be unsurprising to find that people in such a situation suffer conceptual - or even physical - paralysis. They have so little basis on which to act."

The problem in learning software is that no match is usually provided between what was executed and how "correct" it is. Bigge (1982,p.248) points out,

> "For a learner to perform the rewarding function on himself, in place of any external reward, he must have a continuous available knowledge of results. 'Knowledge of results should come at that point in a problem solving episode when the person is comparing the results of his tryout with some criterion of what he seeks to achieve.'"

and as Travers (1975, p. 113) suggests,

> "Information given before responding can influence behaviour only when the information can be stored until needed. Information given after responding can influence behaviour only if a knowledge of the response can be stored long enough so that it can be evaluated in terms of information given."

The advantage of a computerized system is that these reference points can potentially be supported and stored, thereby allowing the learner to view and absorb the outcome of their actions as Travers (1975) advocates. Only by such means can memory, mental effort and psychological comfort be supported in the face of such high possibilities for negative transfer of performance. In the ideal situation, such as in the PALS system proposed by Boyd and Jaworski (1985), new software commands would be translated into the user's habitual commands, alleviating any transfer problems.

4. Long-Term Memory (LTM) Retrieval Failures (storage,templates, icons, mnemonics,manuals,macros,subject matter relevance) (memory)

It is a well known fact that humans have difficulty remembering what has been stored in memory. The exact causes are not understood, but there is indication that frequent use increases recall, and that the manner of encoding the information initially also aids in future recall of it. Travers (1977,p.402) suggests,

> "...transfer of information from short term memory systems to long term memory systems takes place with greatest efficiency if the information is organized...The function of outlining is that of drawing attention of the learner to the inherent structure of what is to be learned, around which the details can be clustered."

The use of "chunking" by way of images, mnemonics, categorizing of similar functions together, is deemed helpful here.

In business, databases, knowledge bases and other storage facilities, as well as support staff, are used to augment and assist memory on the job. However, long-term retention of software use know-how is jeopardized due to infrequent periods of training and subsequent use. (Guillemette, 1989). Templates, user- created command files and online help attempt to provide memory jogs. Demonstrating the relevancy and merits of this tool for assisting the job function can greatly increase motivation, and hence, use and memory of it. As well scheduling planned times for training and use, and the use of organizers can be beneficial (Harmon, 1987).

5. Motivation - Intrinsic (tone,feedback) (comfort)

Intrinsic motivation is deemed to play a role in learning, according to Weiner (1986), when learning is inhibited due to the learner attributing poor performance to lack of ability, which has been reinforced from prior experience on any number of tasks. This leads to self-defeating performance and inflated anxiety and discomfort. Such feelings, and its detrimental effects on performance, can only be overcome if the perception of poor performance can be shown to be a lack of effort or prerequisite knowledge or exposure to the subject material, and not to the user's innate inability.

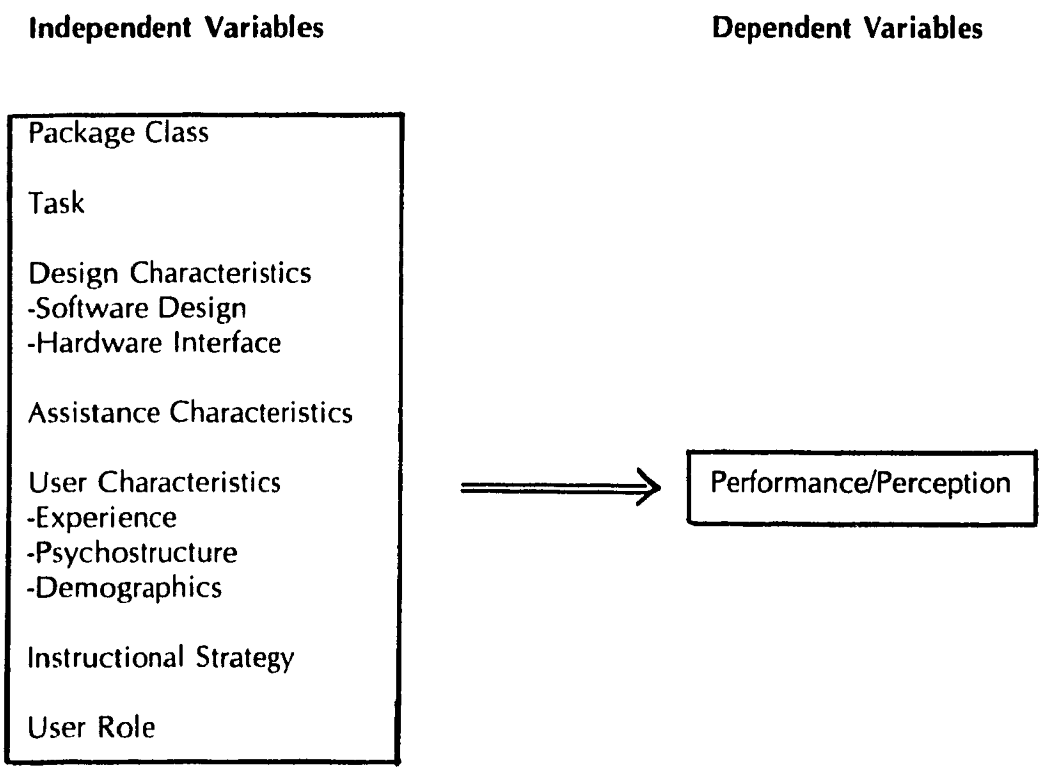
This phenomenon is seen often in the use of business software packages, where users may attribute their poor performance to inability rather than to bad software design. As a consequence, limited or no use is made of the package, reinforcing fears of further use. Carroll and Mack (1984) give anecdotal evidence of this in describing the reactions of Typist learners compared to a Ph.D. learner. The Typists assumed it was their deficiency which hampered performance, while the Ph.D. learner assumed it must be the manual which was inadequate. This was also evident in our research, in which one participant reported, "It may be my own fault. I'm not really an expert user". Either of these ways of reasoning may or may not have been true. The answer to this problem is to make designs so easy to use, that no one need feel inadequate learning and using the package. The tone of 'system' and 'error' messages is also important. Shneiderman (1982) conducted experiments in which it was shown than threatening versus non-threatening wording of messages influenced how comfortable users felt interacting with the system.

The deterrents and remedies discussed above identify a number of features which pertain to a variety of factors affecting software ease of use, such as learning environment; software design; instructional design; training; user characteristics; et cetera. Means examined to overcome the various deterrents to learning include attention to: **screen design - wording, grouping, command naming, tone, consistency; structure - depth, logic, consistency; relevancy to task; subject matter assistance; feedback; simplification; manuals; tutorials; training - practice, examples, step-by-step instruction; storage facilities; points of reference - traces, checklists, memory aids, navigational aids; error trapping - correction, and recovery; macros; templates**. All of these features are expected to offer support to one or more important factors in learning - **speed of performance, memory, mental effort, and psychological comfort**. The section which follows summarizes these elements into a framework in which to examine the Human Factors literature.

2.C. Framework of Software Ease of Use

To structure the factors spoken about above, which are deemed to contribute to ease of use and learning of software, a framework is offered which identifies seven important factors. These are shown in Figure 2 below as - the class of package, package design characteristics, package assistance characteristics, instructional strategy employed, user characteristics, nature of the task, role adopted by the user. In this thesis, these factors are said to impact users' objective performance with, and subjective perceptions of, the package.

Figure 2 - <u>Variables Affecting Ease of Use of Business Software</u>



The nature of the relationship between these variables poses severe problems for software design research. Because of the interconnectedness of these variables, it is still not known which variable or variables exercise the greatest impact on user performance or perception. Likewise, it has not been established whether these variables have a mediating or a primary effect on the dependent variables. The underlying cause of this lack of a theoretical base is due largely to the fact that each of these dimensions is arrayed along a spectrum. The combinations of positions along these spectra defy human consideration and manipulation. It is no wonder that research has been hard-pressed to arrive at definitive conclusions and to establish a sound theoretical base.

The dimensions of these factors can be organized as shown in Figure 3, below.

Figure 3 - Spectra of Software Ease of Use Dimensions

Dimensions	⟵	Spectra	⟶
Class of pkg.	Canned ⟵	⟶	Decision Analysis
Pkg. Type	Payroll/ Inventory	Stats/ OR	WP/DB/SS/CAD
Pkg. Operation			
-Input			
-Process	Structured ⟵	⟶	Unstructured
-Output			
Task	Structured ⟵	⟶	Unstructured
Task Elements			
-Input	Predefined ⟵	Predefined ⟶	Varied
-Process	Predefined ⟵	Varied ⟶	Varied
-Output	Predefined ⟵	Varied ⟶	Varied
Design Chars.			
-memory			
-speed	High ⟵	⟶	Low
-effort			
-comfort			
Assistance			
-memory			
-speed	High ⟵	⟶	Low
-effort			
-comfort			
User Chars.			
-exp-pkg	Novice ⟵	⟶	Expert
-exp-subj.			
-psy-anxiety	High ⟵	⟶	Low
-psy-motive			
-psy-cog.stls*	Methodical ⟵	⟶	Free-spirited
-demogs.		varied	
Instructional Strategy	Scripted ⟵	⟶	Exploratory
User Role	Passive ⟵	⟶	Active

*The possible cognitive style dimensions are far more complex than this simplistic representation implies. See Chapter 2.D.5.ii.a. for a more rigorous treatment.

The two main classes of packages were identified earlier as **specific-purpose** packages and **general-purpose** packages. '**Specific-purpose** packages' refer to what are called 'canned' packages or packages designed to fulfil a specific function, such as accounting packages. These packages are usually used for tasks in which the inputs, processes and outputs are pre-defined or pre-specified, and for tasks common across industries, such as payroll and inventory. The user also has little control over the input, processing or output requirements of the package itself. At the other end are the **general-purpose** packages designed to support decision-making and analyses. Statistical packages, wordprocessors, spreadsheets, database packages, et cetera, fall into this category. These packages offer the greatest flexibility to users, in terms of what is input, processed, and output. It allows the creation of a workspace in which the user is free to manipulate input in order to assess the impact on output. These latter packages require more user know-how of the workings of the package than is required by canned packages when they are used in 'passive' mode. However, canned packages, which have stricter requirements for input, processing and output, may have even greater design requirements for ease of use.

The nature of the task to be accomplished can range from **structured** to **unstructured**. **Structured** tasks refer to tasks for which established, pre-defined procedures exist. These lend themselves easily to package designs which impose structured input, processing and output. **Unstructured** tasks, on the other hand, require problem solving on the part of the doer to determine the inputs and procedures leading to resolution of the task. This dictates flexible package requirements for input, processing and output. Statistical problems and Operations Research problems can be positioned somewhere between the two extremes, in that, the inputs for these problems are usually pre-defined and unchangeable, while the processes and outputs derivable are not.

Design characteristics and assistance characteristics are included in package design in order to support to some degree, **speed, memory, effort and comfort**. The extent to which these characteristics support these factors can range from **low to high support**. A package offering high support for these factors presumably will promote real and perceived performance gains over one offering low support.

The strategy employed in learning a software package can range from following a **scripted** or prescribed sequence of materials and methods to an entirely free-form, individualized, **exploratory** quest. Tutorials, examples, workshops, classroom lessons, et cetera, characterize the scripted form of instruction. Exploratory learning is a hands-on, trial and error, learning-by-doing and thinking approach.

User characteristics can be grouped into experience levels, psychostructure make-up, and demographics. Users' experience with computers and their associated disk operating systems, various packages, and the subject matter of the package and of the task, can range from **novice to expert levels**, in any one area or combination of areas. The psychological make-up of users implies that users have various cognitive styles or approaches to learning. While there are a number of classification schemes and instruments found in the Learning literature for assessing these, discussed later in this chapter, learners, and in our case learners of software, are described for convenience at the extremes, as either having a **methodical or a free-spirited** approach to learning. They also exhibit degrees of motivation and anxiety towards computers, from low to high. Demographically, computer users can be expected to have a range of educational backgrounds, age, work experience, et cetera, and to be for both sexes.

The user role in operating a package can fall anywhere on the spectrum from **passive** to **active** depending on the type of package. **Passive** users respond to system prompts, while **active** users take the initiative in prompting the system and creating the workspace.

Generally speaking, one would expect canned packages to be amenable to a scripted learning strategy, since there is relatively little flexibility with regard to what can be done with the package. Precise examples, therefore, should suffice to impart requirements of the package. These packages, also, imply the presence of a highly structured task, for which few exceptions exist. The user is likely to be at his most passive, responding to pre-defined requirements. One would expect novices to be more comfortable with this type of package than with more open-ended ones of the decision- support type, in part due to the high computer anxiety levels likely to be experienced. Novice users, in general, as well as methodical users are also likely to be more at ease in a canned environment and/or using a scripted strategy of learning, which offers the most guidance and least chance of error. The desired package design characteristics and assistance characteristics at this end of the spectrum dictate high support for memory, speed, effort and comfort. Novice users are likely to have a low tolerance for low support of these learning factors.

Packages for decision analysis require a highly active role on the part of the user, who is usually engaged in a mainly unstructured task. Free-spirited individuals, as well as experts, are likely to be at ease in this environment, as well as in using an exploratory instructional strategy. Expert users' computer anxiety should be lower than that of novices. Therefore, these individuals should have a higher tolerance for packages which have low support of memory, effort and comfort. Nonetheless, in all cases, high support will always be superior to low support.

The framework which has been discussed and presented above offers a rich source of research ideas. However, as noted earlier, the possible combinations which can be derived from this framework are beyond the scope of the human brain to contemplate. Research can address any one of the spectra identified, or any combination of spectra, investigating its impact on ease of use measures. Comparisons can be made within package class categories and across them. Various canned packages can be compared with each other, and with decision-support packages. Various design characteristics and assistance characteristics can be compared in terms of their support of memory, speed, effort and comfort, and the effect that these have on ease of use. Instructional strategy can be varied along the scripted-exploratory spectrum, examining its effect on ease of use, and whether this effect is different when applied to canned versus decision-support packages. These effects can also be analyzed in terms of the impact of user characteristics, the nature of the task, and the role to be adopted by the user. Which of these factors, or combinations of factors, contribute most to a package's **ease of use**? Or, is this dependent on location on the spectrum?

The following section examines the Human Factors literature in relation to the framework proposed above in Figure 3.

2.D. Factors in Software Ease of Use and Empirical Findings

The various features included in a package are presumably not there arbitrarily, without just reason. They should in some sense contribute to the **ease of use and learning** of the package. Package features can be classified into the categories of software design characteristics, hardware interface (not addressed in this thesis), help features and learning aids, accommodation of user characteristics in terms of level of expertise, psychostructure make-up and demographical differences. Other factors which contribute to a design's ease of use include the type of package being employed, the instructional strategy adopted in the teaching of the software, the nature of the task to be performed and the nature of the role the user is expected to play in the performance and use of the package. This section will examine these individual factors with reference to representative samples of the empirical evidence which has been found in the literature. Tables summarizing this research can be found in Appendix J.

2.D.1. Package Class

As noted previously, the type of package to be used will have an impact on the requirements for ease of use of tne package design. Specific-purpose packages may require greater attention to ease of use in their design than the general-purpose class, because of the restrictive functions permitted by this class of package and the limitations this places on the role of the user. Because of the structured nature of these packages, consistency becomes vital and should, therefore, make the package easier to use. Nonetheless, all classes of packages aim at being easy to use and learn. The research, which has been conducted along this dimension, has focused on comparing the relative ease of use of designs falling within the same package class and type. An example is Napier, et. al.'s study (1989) which compares the Lotus$_t$ 1-2-3 interface with that of the Lotus$_t$ HAL interface, in which the HAL interface was shown to out-perform and 'out-preference' the original Lotus$_t$ interface. (HAL was the name given to the overlay added to the original Lotus$_t$ interface and is not an acronym). A study by Green and Gilhooly (1990) examined the learning of the statistical package MINITAB$_t$ by novices, noting differences between fast and slow learners. Whiteside, et. al. (1985), on their part, examined seven different database management systems in order to compare different command structures, whereby they found that all users performed best with commands. Many of the studies concentrate on particular wordprocessing systems and, in fact, on the earlier text-editing systems. (Mack, Lewis and Carroll, 1987; Holt, et. al., 1986; Carroll, 1985; Cohill and Williges, 1985, 82). Very little, research, if any, seems to examine specific-purpose packages or to compare specific-purpose with general-purpose packages.

2.D.2. <u>Task Type</u>

The nature of the task may also have an effect on learning and using a software package. It certainly has an effect on the type of instructional material provided. One would expect structured tasks to lend themselves to structured design and training. However, the very nature of 'structuredness' when applied in a practical setting is not well defined. Set rules and ways of doing things are usually associated with being structured. The more this is so, the more amenable the problem is to a computerized solution, such as canned packages. At the other end of the spectrum are the decision support tools (DSS). Currently, however, while the task may be unstructured, the DSS tool to resolve it may require interaction in a structured way. This dissonance is sometimes the cause of frustrations. While research has investigated the effects of prior task experience on performance, it has not looked specifically at the nature of the task, and the effect this may have. Current available tools are not amenable to fully unstructured tasks, that is, do not as yet allow the user fully individualized operation, even if that is where the need lies. Expert system technology is likely to eventually remedy this.

2.D.3. Design Characteristics

While the list of possible package features is endless, there is, presumably, some consensus regarding the most important elements to be considered and included. These are enumerated and explained below and are derived from the discussions in Chapter 2.B., the Human Factors literature, and personal contributions of this author. The learning factors expected to be supported are shown in brackets. Most of the research has focused on interface command structures and command naming, and other aspects of screen design. Each of the Design Features are discussed in turn. The support we expect them to offer user learning are shown in brackets.

2.D.3.a. Design Features

i. **Interface Command Structure (speed,memory,effort)**

Much debate has taken place concerning the relative merits of different command structures, or means by which users may interact with software. (Shneiderman, 1987; Relles, 1981, et. al.). Command structures can be either:

- menu - selecting from a list of items or iconic images;
- command - issuing strings of letters or keywords;
- direct manipulation - interacting with manipulation devices, such as lightpens or voice;
- natural language - interacting using everyday parlance, which may also be through voice input and output.

Presumably these structures will impose different memory requirements on users. Commands require recall which is more taxing on the memory than is recognition, which is required in the other structures. The additional effort imposed on memory by command structures would imply that it is a less desirable structure than other structures. However, these structures offer the user more flexibility in the operation of the package. A pre-defined sequence of steps to execute the operation is not imposed as is characteristic of many menu structures. Direct manipulation environments, such as MacIntosh and Windows, are often viewed as being the most user-friendly and consistent interfaces. Nonetheless, users can get lost in the many windows or tiles appearing on the screen. Also, just as people attribute different semantic meanings to words, the same holds true for iconic images. Speed of performance is also usually sacrificed. (Shneiderman, 1987).

A number of studies have investigated these assumptions and have found that the answers are not as clear-cut as one would expect. A study by Ogden, et. al. (in preparation), reported in Paap and Roske-Hofstrand (1988), investigated the use of menus and commands in the same system. To their surprise, although users consistently preferred using one of the methods over the other, this preference was not necessarily for menus. Only one third of the users consistently used menus and, in fact, those with the least experience relied on using commands, while the experienced users selected menus. Lee, et. al. (1986) investigated novices using a hierarchical menu only versus a combined menu and keyword command structure, and found that the combined approach was preferred and led to better performance. Keywords acted as a way of bypassing the tedious menu hierarchies when needed.

Whiteside, et. al. (1985) looked at seven different database management systems with different command structures - menu, icon, command. Interestingly, novices had great difficulty with the menu structures, and the command structures proved to be best for all user categories. Shutoh, et.al. (1984) found that for novice, as well as advanced computer-assisted design system users, voice input was faster than menus. Hauptmann and Green (1983) found no difference in performance time or accuracy on a package offering all of menu, command or natural language structures. Users were novice to experienced programmers with no prior package experience. Napier, et. al. (1989) found that HAL's natural language interface resulted in higher performance and preference among inexperienced computer and spreadsheet users than with the original mixed - menu/command, Lotus, interface. In examining menu selection methods - 'enter selection and press return', 'enter selection only', 'highlight selection and press return', Dunsmore (1981) found that high school students preferred the highlighting method. While the 'enter selection only' option was slightly faster, it was also prone to resulting in more errors. Perlman (1984) investigated the relative merits of menu selection based on mnemonic terms, sequential lettering and sequential numbering. Mnemonics required the least amount of think-time, sequential lettering required the most, and sequential numbering was in-between.

Ehrenreich and Porcu (1982) studied the effects of various forms of command naming - truncation versus contraction, fixed versus variable length commands, non-systematic endings. Their subjects were military enlisted personnel. The formation or encoding of command names was easier by truncation than by contraction, if the rules were known and consistent. The length of the command or the inclusion of non-systematic endings did not have an impact on encoding but did on performance. The inclusion of endings lowered performance, that is, identification of the command names. Having rules to guide the encoding process was better than not having rules. In the decoding of commands, if the rules were known, then there was no difference in performance using truncation or contraction to form the commands. However, if no rules were given, then truncation was better. Commands with variable lengths were decoded more readily than those of fixed length, as were those with endings. Moses, et. al. (1980) found in their study that for encoding, truncation as well as no-pattern-in-command naming were superior to contraction, though truncation and no-pattern were not different from each other. No differences were found for decoding.

Two experiments by Landauer, et. al. (1983) continued this stream of research into command naming. In the first experiment, computer-naive subjects were asked to use 'natural language' to explain a text-editing task to someone. No agreement was found among subjects as to the choice of naming, and the command names currently used in the system were not among the ones thought of by the subjects. The follow-up experiment compared the old set of commands with a new set derived from experiment 1, and also with a random set. Performance time turned out to be the same, and, in fact, the old set was somewhat preferred. The results of these two experiments indicate that if a system is to have many users, it may have to have the naming commands imposed, but with explanations of their functions made explicit or, alternatively, recognize all synonyms. Grudin and Barnard (1985) found that even allowing users to create their own command abbreviations did not improve performance, neither did system command abbreviations alone, whereas providing the full commands along with their abbreviations did. Black and Moran (1982) composed eight command sets, and found that infrequent, discriminating commands resulted in faster learning and superior recall, while general words performed the worst. Interestingly, nonsense words perform well, but the results may be different over longer periods of use. In a concentrated experimental setting, users are, perhaps, more willing to tolerate imperfections than they might be if called upon to use the system for extended durations, and for work purposes.

Work with more creative interface concepts is surfacing. In 1984, Good, et.al., developed a system based on the actual behaviour of novices with an electronic mail task on a VAX computer. It used a command-line interface, which had no help, no menus, no documentation, and no instructions. The task was explained to the user in the context of a paper mail analogy, and each task was to be executed in order. Through a human interceptor, unbeknownst to the user, the system adapted to meet the needs of the 67 novice users taking part in the research. Each discovery was incorporated into the make believe, partially automated system, as it occurred. An example given was having the system adapt to be able to accept synonym command names for 'delete', such as 'erase', 'destroy', 'throwaway', 'clear'. The final system proved very successful for novice users.

Kantorowitz and Sudarsky (1989) suggest that being able to access more than one dialogue style in the same command may be desirable, such as for an intermediate user who, half way through a command, forgets the rest of the syntax and needs to revert to menu assistance. Using this concept they developed the system GUIDE, which uses DBASE III + to store the different user interface specifications. The Merlin statistical package designed by Peter Wade, and used in our Study 2, also, permits users to switch between menu and commands midstream.

For their part, Mahling and Lefkowitz (1989) developed the CRUISE interface which dynamically configures the interface based on functionality, by evaluating its knowledge bases of interface design knowledge, user model and dialog histories of the user with the system. They suggest that user classification is more than just computer experience level, and may include frequency of use, rate of retention, computer literacy, et cetera.

Wolf, et.al. (1989) have been working on the concept of a 'paper-like' interface, exploiting our usual way of doing business, adapting it to advanced technology. In this context, interaction with the computer is via a digitizing pen. The attractiveness of this approach for artists and musicians writing scores is immediately evident. An example of its application to business problems was given in relation to a spreadsheet problem. To designate the desire to sum a set of rows or columns, the user draws a line through them and selects the summation sign, sigma, from the menu options provided at the side of the worksheet. A math formula can be entered in the format usually written on paper, and the computer is able to perform the necessary calculations. It is an intuitively appealing approach.

ii. **Depth of Structure (speed,memory,effort)**

Not only is the choice of command structure important, but so is the depth of the structure. Debates revolving around the depth versus breadth of structures have not been conclusive. The deeper the structure, the more difficult it is for a user to remember and to navigate around in. (Shneiderman, 1982). However, tradeoffs have to be made for breadth. Having too many items on a screen can be cluttering and, therefore, confusing. If, however, users are trained and familiar with the items, this may not be problematic. (Burns, et. al., 1986; Badre, 1982). This is evident in industry where the Lotus, package has been favoured over packages such as Supercalc, which has a shallower menu structure. Deep structures, in most instances, would impose a greater requirement to remember under which sub-branches desired commands can be found. It also means that a longer time will be expended before the desired operation is accessed and executed, compared to shallower structures. In Ogden, et.al.'s study cited above (Paap and Roske-Hofstrand, 1988), users selected the menus only if access to the desired operation was immediate. The greater the number of levels, the more use seemed to decline.

iii. **Logic of Structure (effort,memory,comfort)**

If the structure of the package design is not consistent with the view the user has of how it should work, dissonance is likely to be high, creating anxiety which will be detrimental to performance and perception. Schlager and Ogden (1986) revised the manual for the SQL query language based on a model of expert users. They found that successful performance depended on whether the model was consistent with the operation of the system.

iv. **Consistency (effort,memory)**

This feature is considered high priority within and across packages, affecting ease of use and learning. (Houghton, Jr., 1984; Katz, 1983; Relles, et. al., 1981). Consistency in package conventions, functions and messaging is necessary if there is to be any retention and transfer of know-how, from one package to another. The necessity of remembering different command names and sequences at different junctures in the same package or different packages, in order to produce the same outcome, is likely to be very debilitating, as was discussed in the section on learning. Many examples of these inconsistencies exist within and across packages, though, research into their effects on performance has not been proportional to its importance. Karat, et. al. (1986) investigated novice users and those experienced with a wordprocessing system, on a different wordprocessing system. They found that syntax differences in the new system created a lot of difficulty for those with prior experience. Maskery (1985) likewise found, in leading naive users through an adaptive interface design for a statistical tool, that if the changes between the dialogue style were not consistent and predictable with the previous style, this led to confusion.

v. **Screen design (speed,memory,effort,comfort)**

This has to do with the cosmetic aspects of design issues of text size and location, use of figures, graphics, colour, sound, highlighting, et cetera. Burns, et. al. (1986) found that, by redesigning a display, novices were able to improve their performance speed and accuracy, while expert users showed no change in performance time, but had fewer errors.

Badre (1982) used experienced tactical decision makers to test three forms of information presentation - meaningful chunks in a familiar order, non-meaningful chunks, meaningful chunks in an unfamiliar order. As would be expected, information presented in meaningful chunks and familiar order resulted in greatest recall. However, non-meaningful chunks produced greater accuracy than did meaningful chunks in an unfamiliar order. In line with learning transfer theory, the possibility for negative transfer is greatest when there is some overlap, but not identical mapping. When there is no overlap, there is no transfer, positive or negative, so learning is not impeded. In a second experiment, Badre (1982) changed the placement of the information on the screen, which made no difference to recall. The same information content was maintained.

vi. **Flexibility (memory,effort)**

The greater the number of alternatives available with which to perform an operation, the greater the memory load and learning effort and the possibility for confusion during recall. No empirical studies were found investigating the effects of this. One of the reasons given for the demise of PL1 was that there were too many ways of performing a function and the efficiency varied greatly (Wade, personal communication).

vii. **Error Trapping and Recovery (comfort)**

The ideal system should have provisions for input error trapping, such as with parameter ranges, automatic correction, for example, transforming letters 'o' and 'l' to numbers '0' and '1', respectively, as well as the facility for undoing errors. There is the possibility that reliance on the system in this manner can lead to laziness and inattention on the part of the user, but this is likely to be trivial. The benefit realized from this 'preventive medicine' feature is likely to be peace of mind for the user.

viii. **System Response (speed)**

This feature refers to the timing delay between user input and system response, as well as execution time for operations to be performed. Research indicates that optimal ranges exist for the rates at which text is displayed and system responses are made. Users' performance diminishes outside of these ranges. While Dannenbring (1983) did not find response times affecting performance or satisfaction of either novice or expert programmers, Smith (1983) found that skilled and novice computer-assisted design system designers performed better with slower response times. These contradictory findings may be due to the different hardware interfaces involved, keyboard versus lightpen, respectively. Adding to the contradictions is Long (1976) who found that longer response times resulted in lower performance over time for both skilled and unskilled typists. Bevan (1981) investigated display rates for low and high reading skill abilities. As the rates at which text appeared on the screen increased lesson time decreased but at the cost of increased errors. Fast display rate was disliked by both groups of subjects, with low ability subjects preferring the lowest rate.

ix. **Data Input (speed,memory,effort)**

The manner of data input is also a factor in package use. Interactive data entry, typified by full-screen editors, facilitates user entry at novice levels, while batch entry may be preferred by expert users, who are interested in speed of entry. Systems which do not allow users to correct previous input, but requires them to start over, are likely to be viewed as burdensome, needing too much effort. Dunsmore (1984) found this to be the case in two studies of non-programmer computer users comparing the two methods of data entry and editing. Subjects in the line-editor group took 17% longer to learn the editor as compared to the screen-editor group and were not able to make as many corrections. The screen-editor groups in both experiments were able to complete 40% more corrections in a fixed time period.

x. **Macros (speed,memory,effort)**

The facility to create macros, that is, to automate frequent operations is seen as a way of accommodating higher levels of expertise, without jeopardizing novice requirements. The tedium of entering repeated operations is removed and operations are accelerated. While this feature has mostly been associated with expert use, it may, in fact, be a means of simplifying the novices learning task. The automation of frequent, repeated tasks under an appropriately named macro, such as 'PRINT' a document, relieves the user of the need to know the separate steps required to execute this operation. Knowing one command is all that would be necessary for the passive user.

xi. **Autosave and Backup (memory,comfort)**

Designed to reduce user anxiety over lost work, this feature provides a means for intermittent saving of work. This alleviates the strain of remembering to save, and the possibility of large amounts of work being lost due to system failures, power outages, user error, et cetera.

xii. **Autoadjustment (speed,memory,effort)**

Systems which automatically adjust analyses after changes in inputs are faster and require less effort on the part of the user. For instance, in the integrated package, Framework, changes in the cell of one spreadsheet are automatically reflected in other linked spreadsheets.

2.D.3.b. Hardware Interface

Hardware interfaces are those communication devices used to interact with the computer, the relative ease of use of which is in contention. While these devices are not part of the research study described in this thesis, a brief mention is made here. The devices include:

- mouse
- lightpen
- touch screen
- keyboard
- voice

Studies have investigated the contribution made by these different communication devices to ease of use and found mixed results. (Emmons, 1984; Ogozalek and VanPraag, 1986; Murray and Abrahamson, 1983). Touch screen and lightpen require less manual dexterity and mental effort than keyboard or mouse entry. However, this raises issues concerning screen layout, given the restrictions it places on the amount of text displayed and on the placement of the displays. Various keyboard layouts and cursor and function key locations have been tested, as well as the number of buttons to include on a mouse. The industry has not necessarily adopted the recommendations. For instance, though the DVORAK keyboard is a superior layout to the QWERTY keyboard, the QWERTY conventions established over the years in the typing world causes resistance to this change. Another example is the placement of the function keys. Pattern recognition capabilities in the human, and minimized hand movement, would indicate a greater ease with the placement of these keys to the left of the keyboard. The standard which seems to be evolving, however, is toward placement in a straight line at the top of the keyboard. It is questionable whether this convention is being adopted because of human factor considerations, or for the convenience of hardware developers. Card, Newell and Moran (1983) have proposed a model, the Keystroke-Level Model, by which expert performance times on a particular task, using any of these devices, may be predicted and compared. As this thesis will not address the hardware issue, the pertinent literature is not presented here.

2.D.4. <u>Assistance Characteristics</u>

A range of assistance tools are usually provided with a package to facilitate use and learning of the package (Shneiderman, 1987; Houghton, Jr., 1984; Relles and Price, 1981; Douglas, 1982). It would seem likely that the more help features available, the more secure and comfortable the user would be, because of the support this offers to memory. Some of the research suggests, however, that users are often oblivious to the assistance available to them or are unaware of how to make use of it. Help becomes useful when the user knows what to look for and how to operate the feature. In addition, while one would expect users to make the most use of online aids, there is some indication that users have a preference for or may be conditioned for offline aids. There are a variety of help aids, both online and offline, and the list continues to grow, but research has been concentrated on the effects of very few of these. The major emphasis has been on various forms of online and offline manuals. Some work has been done investigating the advisability of restricting access to certain options, especially at initial stages of learning, and into the effects of conceptual models. Each of the Assistance Features are discussed below and the support they offer to user learning are shown in brackets. They were derived from the discussions in Chapter 2.B., the Human Factors literature, and personal contributions.

2.D.4.a. Help Features and Learning Aids

i. **Manuals (memory,effort,comfort)**

Online and offline manuals provide documentation of the package and its operations. Detailed instructions in the use of the package are provided. These offer support to memory and peace of mind, by providing ready references. There has been some discussion on the level of detail of instructions that should be provided and at what stage of expertise, as well as on its form, content and organization. A study by Relles (1979) looked at novice and expert users of computers using online assistance with and without a printed manual, and using only a printed manual. Novices performed poorly with online assistance compared with using a manual only, while the opposite was true for experts. Similarily, Dunsmore (1980) found that novices performed best with a brief summary plus an offline manual, and worst a with brief summary and online manual. In contrast, Hiltz and Kerr (1986) found, in an exploratory study, that users tend to skim documentation whether online or offline, and rarely read offline documentation. This may account, in part, for the additional finding that one third of users never progressed to advanced features. Users tend to stick with whatever rudimentary operations they have learnt even when better methods exist. This is also true of programmers. Carroll and Aaronson (1988) did an exploratory study of users who were experienced with computers but not with business software, and found that both a how-to-do-it manual and a how-it-works manual are needed for effective learning and use. Schlager and Ogden (1986) found that by revising a manual to conform to the cognitive model of an expert, initial learning and retention could be improved. Charney and Reder (1986), on their part, found that assistance in creating a spreadsheet which was task-oriented improved performance, when subjects were presented with new problems, as compared to assistance which gave step-by-step instructions only. Research by Carroll (1985) found that giving novice users a reduced version of the full manual improved performance on initial tasks and transfer tasks to advanced topics which were covered in the full manual, but not covered in the reduced one.

ii. **System messages (memory,comfort)**

These online prompts inform users of the consequences of actions taken by them, but not why it occurred or how to correct it or proceed, such as 'illegal entry', 'invalid data', et cetera. They are, therefore, a support to memory and offer peace of mind to the user. Shneiderman (1982), in four experiments, manipulated message tone and wording of UNIX's text-editing system to investigate its effects on performance. The first experiment varied 'current system messages', 'improved message tone', 'increased specificity' and, 'improved tone and increased specificity'. Subjects were undergraduates in an introductory computer course. Only increased specificity, that is, more explicit descriptions of the error detected, improved performance and subjective evaluations, though all were better than the current messages. In the second experiment, the addition of messages to one of UNIX's text-editors, which responds with only a question sign to detected errors in user input, improved the performance and satisfaction of undergraduates who had prior experience on the UNIVAC text-editor but not on the UNIX. The third experiment compared 'current messages' with 'specific-courteous messages' with 'hostile and vague messages'. Interestingly, although the option of 'specific-courteous messages' was preferred overall, it did not outperform the 'hostile-vague messages', but was better than the 'current messages'. In a fourth experiment, the improved messages, for example, 'The system control program could not locate an account number on the first card image of your computer run', also outperformed the current messages, for example 'missing account'.

iii. **Tutorials (memory,comfort)**

Many packages provide online and offline training modules designed to instruct users in the software capabilities and how to use them. A tutorial, as these are called, is a sequence of instructional frames on particular operations of the package which provides explanations, examples and the opportunity for practice and correction. Roemer and Chapanis (1982) investigated the effects on performance of different writing levels in the tutorials. Technical and non-technical subjects with low, middle and high reading ability were chosen for the experiment. It was found that a fifth grade level was preferred by all, though performance was not affected by increased complexity in writing style. Nakatani, et. al. (1986) examined the wisdom of a conventional tutorial compared to one designed to elicit mastery levels of learning. As expected, the mastery tutorial produced far superior performance, but at the expense of speed. Czaja (1986) found that the online tutorial resulted in poorer performance than either a document-based or instructor-based tutorial.

iv. **Prompts (memory,comfort)**

In some instances, in response to erroneous input, users are prompted by a list of options, indicating the format and parameters permitted at that point.

v. **Keyboard templates (memory,comfort)**

These are memory aids indicating the command or operation associated with specific keys on the keyboard, and can be either online or offline.

vi. **Defaults (speed,memory,effort,comfort)**

Assistance may also be provided by the insertion of most-likely or customary entries, which the user then has the option of modifying. This is a learning-by-example type of approach, which is also used with form-fill-in type input formats.

vii. **Examples (memory,comfort)**

Examples of correct or valid input and actions, as well as non-examples, that is, examples of incorrect actions, are sometimes provided. These are an aid to memory and provide a level of comfort to the user. Magers (1983) found that including examples and, in particular, task-oriented examples, in a revised, enhanced online help system, improved performance. However, as these were not examined in isolation, performance improvements cannot be estimated exclusively for this feature.

viii. **Index (memory,comfort)**

Indices are usually a quick reference to a listing of all available commands and operations, with some explanation and sometimes examples of these. The emergence of hypertext systems can be used for a particular representation of indices. With hypertext representation, indices are imbedded in paragraphs of text which users may access by highlighting the indexed word or words. These selections can lead to definitions of the word or phrase highlighted, or to deeper explanations of their concepts. These systems can be used to provide information retrieval for reference or, more elaborately, as a means of providing course material for training. They allow for information to be presented in increasing levels of detail and explanation, making them an appropriate medium with which to deliver training. (Shneiderman and Kearsley, 1989).

ix. **Glossary (memory,comfort)**

A glossary provides a list of definitions of terms and commands.

x. **Unsolicited Help (memory,comfort)**

While the user usually solicits help, it may be desirable to have the system initiate providing help. For example, after prolonged delay in user entry, the system may prod the user for an entry or even suggest what the entry should be. Again, there may be a tendency to over-reliance on this feature, causing the user to be sluggish in learning.

xi. **Cautions (memory, comfort)**

Handy reminders or attention getters, though often given little attention by users, can be very useful. An example is a system seeking verification from the user as to whether a previously named file should be overwritten. Most of us who have ever used a package have vivid memories of inattentively responding 'yes' to the prompt to erase or overwrite a file and, split seconds too late, recognizing the devastating error made. The MacIntosh environment gives some safeguard against this type of error by providing a trash can in which deleted files are temporarily placed.

xii. **Checklists and Memory Jogs (memory, comfort)**

Traces of most-recent or most-used operations act as memory jogs or checklists of what has been or should be done. These reminders can ensure that all necessary steps are followed and may act as a reference point for novices. A fuller discussion with examples of these types of aids can be found in Section B of this chapter.

xiii. **Navigational Aids (speed,memory,comfort)**

These aids facilitate placement and movement within the package, and include trees, maps, control keys such as back-up, page forward, et cetera. These offer support to speed of movement, and memory and comfort, by providing means by which the user can locate himself in the package. This is, particularily, useful when navigating within deep menu structures.

xiv. **Instructive Feedback (memory,effort,comfort)**

In this scenario, users' input is assessed for correctness by the system, and errors, if any, are identified for correction before instructions can be followed. Correctness, or lack of it, is then conveyed to the user, in some instances, including options for correction. In the package Merlin$_t$, when an unusual request is made, the anomaly is pointed out to the user and confirmation to proceed is requested, for example, when the lower limit on a control chart is negative. Egan, et.al. (1987) provided diagnostic and remedial assessment of errors to users and found that both assessments reduced the tendency to make repeated errors. It was also found that error patterns exist in a sizeable number of users. This would indicate the feasibility of providing 'intelligent' feedback to users, and warrants further investigation. Support is offered to the user's memory, effort and comfort and facilitates learning.

xv. **Context Help (memory,effort,comfort)**

Help is provided pertinent to the point at which the user is currently experiencing the difficulty. This is a useful feature for reinforcing correct learning. Magers (1983) added context-sensitive help, in conjunction with other help aids, to existing online help which resulted in improved performance. This feature was not examined in isolation, however, so its effect cannot be judged independently. The Smartware II$_t$ help system is very useful in this regard.

xvi. **Expertise Accommodation (effort,speed)**

Many packages provide for a transition from lower to higher levels of expertise. In some packages, this is accomplished by ever-increasing access to a restricted range of commands and features, as described in the section below. In some packages, this is minimally accommodated, as in Lotus, by the choice of highlighted menu selection or by the first letter of the command. In other packages, this accommodation is more explicit. For instance, in Merlin, advanced users have access to an Express Mode which allows several operations to be specified on one command line which results in speeding up the process.

xvii. **Restricted Options (memory,effort,comfort)**

Modularizing and restricting the options available at certain points has the effect of localizing the impact of errors. It also aids learning by reducing complexity. If users are aware of the option, however, they may access these advanced features before they are ready for them. The Smart package at one time had this feature but removed it. Catrambone and Carroll (1987) found that, by providing a restricted version of the package, learning was accelerated and learning more features of the wordprocessor was encouraged. In Carroll and Kay's study (1985), a scenario training wheel format decreased training time, but had no effect on transfer tasks.

xviii. **Subject Matter Assistance (memory,comfort)**

Some computerized subject matter assistance may be currently available through independent CAL packages, for example, for statistical packages. However, this author has already noted the desirability of including this assistance alongside help directed at operating the packages (Thomas, J.D.E., 1989). It may be that some of the users' difficulty with packages stems from a lack of understanding of the underlying subject matter. Novice users are likely to feel more secure with this available assistance, as would experts appreciate access to these memory refreshers, if and when needed.

A study by Stephenson (1990) on student reactions to Minitab, in an introductory Statistics course, reported that students did not find the package of itself useful for learning about statistics. The addition of courseware could rectify this. The frustrations of using the package, between initial use and the end of term, also lowered their perception of the value of the computer for analyzing data after graduation.

xix. **Conceptual Models (effort,comfort)**

As users interact with a system, they develop ideas about this system and the way in which it works. (Carroll and Olson, 1988). This is often referred to as a conceptual model of the system. Debate has centred on the advisability of either providing models of the system to the user, whether on- or offline, based on the designer's model or on that of the user, or of designing the package such that it adapts to either of these individuals. (Wilson and Rutherford, 1989; Moray, 1987; Mancini, 1987). Presumably, if a match can be made of the user's model with that of the system, that is, design of the package and/or design of the assistance and explanations provided, or if the user can be made to understand the system's model, then learning will be easier because of more directly aimed analogies. The possibility for less effort on the part of the user to understand the system increases, and therefore, the user is likely to be more at ease operating the system. Schlager and Ogden (1986) presented users with a revised manual based on an expert's cognitive model of a database management system. This revised text produced superior initial training and retention. They also compared a novice's conceptual model versus a procedural model versus no model. They found that both models aided in accessing the database faster but did not differ from each other. When searching for unknown conditions, however, the conceptual model proved faster. Bayman and Mayer (1984) looked at users' mental models of calculators and found that they differ in their models, which are also often inaccurate. They found that teaching the model improved performance. In the teaching of Pascal, the programming language, Goodwin and Sanati (1986) compared the use of traditional course material versus material organized around a concrete model of the computer. They found that when the traditional method was used, they were able to predict programming performance based on prior programming experience, but prior experience was not indicative of performance when the concrete model was used. Bostrom, Olfman and Sein (1990) compared abstract models with analogical models, in relation to learning styles based on Kolb's classification. (The section following gives a description of this learning style measure). 'Abstracts' and 'actives' performed better with the abstract model, while 'reflectives' did better with the analogical model. In the first two studies conducted, which involved the creation of a budget and electronic mailing tasks, respectively, this finding was not significant, but it was in the third study, involving mailing tasks only. Study 1 used a budget worksheet as the analogical model and a system of algebraic equations for the abstract model. In studies 2 and 3, the analogical model was an office filing cabinet and the abstract model was a schematic diagram.

A number of design and assistance features examined in this section were said to contribute to learning, to the extent that support is given to reducing performance speed, alleviating memory load and cognitive strain or mental effort, and promoting peace of mind or psychological comfort. These can be summarized in tabular form to produce the mapping as shown in Table 2 below. This listing has commonality with useability issues given by Karat, et.al. (1992, p. 399) which is beyond the scope of this thesis but which includes:

"Use a simple and natural dialogue,
Provide an intuitive visual layout,
Speak the user's language,
Minimize the user's memory load,
Be consistent,
Provide feedback,
Provide clearly marked exits,
Provide shortcuts,
Provide good help,
Allow user customization,
Minimize the use and effects of modes, and
Support input device continuity."

Nielsen (1992, p.378) adds, "Good error messages and Prevent errors".

Table 2
Expected Links between Software Features and Learning Dimensions

Software Features	Learning Dimensions			
	Speed	Memory	Effort	Comfort
a. Design features				
Command Structure - menu/command	X	X	X	
Depth of Structure	X	X	X	
Logic of Structure		X	X	X
Consistency		X	X	
Screen Design	X	X	X	X
Flexible		X	X	
Error Trapping/Recovery				X
System Response Time	X			
Data Input - interactive/batch	X	X	X	
Macros	X	X	X	
Autosave/backup		X		X
Autoadjustment	X		X	
Autoformatting	X	X	X	
b. Help Features/Learning Aids				
Manuals		X	X	X
System Messages		X		X
Tutorials		X		X
Prompts		X		X
Keyboard Templates		X		X
Defaults	X	X	X	X
Examples		X		X
Index		X		X
Glossary		X		X
Unsolicited Help		X		X
Cautions		X		X
Checklists/Memory jogs		X		X
Navigational Aids	X	X		X
Instructive Feedback		X	X	X
Context Help		X	X	X
Expertise Accommodation	X	X	X	
Restricted Options		X	X	X
Subject Matter Aid		X		X
Conceptual Models			X	X

2.D.5. User Characteristics

It is expected that user characteristics, experience levels, psychostructure make-up, demographics, should have some effects on learning, whether primary or mediating. The assumption is that, if appropriate matchings can be identified and accommodated, this will reduce learning strain and promote ease and comfort, thereby reducing anxiety and enhancing performance. One study by Evans and Simkin (1989) seems to discard any of these notions. They used a questionnaire to elicit demographic information, academic achievements, prior computer training and experience, and TV, work and pleasure habits. They also collected information on cognitive styles using a modified Myers-Briggs Type Indicator instrument. They then related these to performance in a computer literacy course. The objective measures collected were grades on two midterm exams, the final, and homework assignments. They found no single variable, whether demographic, behavioural, cognitive or problem solving factor, which could best predict computer proficiency. Evidently, much more research is warranted.

2.D.5.a. Experience Levels (comfort,effort)

Demonstration of effective learning should manifest itself in the difference between novice and expert performances. For Norman (1980, p.75) the difference can be seen in:

i) smoothness - the more expert, the less hesitant and halting the motions;

ii) automaticity - actions eventually become unconscious, not requiring reminders;

iii) mental effort - more thought required initially to remember paths to solution;

iv) stress experienced - novice usually unsure of answer and so conscious of external validity, whereas expert is not concerned because sure of answer;

v) point of view - expert is able to look ahead and to plan the steps to be taken until it becomes an unconscious act and finds it difficult to articulate the precise steps taken to arrive at the solution. Experts are more goal-oriented, or forward-chaining, whereas novices tend to be data-driven and backward-chaining in their approach.

Rigney (1980, p. 335) explains the differences between the novice and expert. The expert is,

"less uncertain about the answers he advances because of his greater speed and fluency in performance, a richer store of appropriate knowledge in longterm memory, being able to process some operations differently, and having moved from slow, conscious control of operations to unconscious, faster levels."

The above characterization is certainly true of the transition made by learners of software. There is, therefore, a complex blend of the factors at work in the individual. Software users are required to know so many facets, semantically and syntactically, that it is impossible to be expert in all. Users will have varying degrees of expertise on various elements. While they may be expert on a particular package, they may be entirely novice on another similar package, especially given the lack of conventions currently in vogue across packages. Their knowledge of their particular task domain and that of the package subject matter is also subject to variability. The user may, at the same time, be knowledgeable in, for example, applying elementary statistical principles to the job, yet be only marginally so in the application of more advanced techniques. These various knowledge bases are also not static over time. (Carey, T., 1982; Martin, 1986; Arnett and Trumbly, 1989; Shneiderman, 1987).

Acquisition of software expertise/knowledge, then, can be summarized as being contingent on the extent of expertise in a variety of areas. These are identified as:

- **Computer**	- extent of prior knowledge of computers and corresponding disk operating systems;
- **Application software**	- extent of prior knowledge of similar package(s), in type and/or operation - extent of prior knowledge of dissimilar package(s), in type and/or operation
- **Application subject matter**	- extent of knowledge of package subject matter;
- **Task**	- extent of knowledge of job function and related tasks.

Shneiderman (1987) distinguishes three classes of software users for whom different support needs are identified. Novices are defined as, those with no syntactic (that is, vocabulary) or semantic (that is, concepts) knowledge of the software, or knowledge of the computer itself. These users require restricted vocabulary and access to tasks and options, constructive feedback, specific error messages, step-by-step online tutorials and well-written manuals. Intermittent users have some computer knowledge, some semantic knowledge of the software, or a similar package, but may have forgotten their syntactic knowledge of it. They need consistent and simple structure, easily accessible online help and manual. Experienced users have knowledge of all the areas and require shortcuts by way of abbreviations, strings of commands, et cetera, with online facility for periodic quick reference. These classifications and corresponding support needs are still unproven.

Schneider (1982) proposes five levels of expertise, with differing needs: Parrot, Novice, Intermediate, Expert, and Master. The Parrot has little or no computer experience and performs operations more by rote than by understanding, and needs examples and menu options. The Novice begins to gain some level of understanding though at a simple level. Menus will still be needed. Intermediates begin to understand underlying concepts and to distinguish optimal command sequences, which begin to become automatic responses. Commands or keywords are now appropriate. The Expert has an extensive grasp of all the concepts and understands their context. They are able to operate at the level of macros while retaining the understanding of the underlying concepts comprising them. This person is able to distinguish clearly between correct and incorrect operations. For these users, commands now need to be expanded to full programming languages. The Master is able to creatively use the system to stretch the limits of the system beyond that orginially anticipated by its designers. Evidently, a sophisticated programming language becomes essential.

The classification by Andriole (1983) is based more on function. Naive users are those who are inexperienced with computing, Managerial users are more experienced, more difficult to please and impatient with irrelevant output, and Scientific-Technical users are the most experienced and difficult to please.

i. Computers and Packages

The degree of prior experience and training on computers and software is expected to influence a user's performance with, and preference for particular software, because of the positive and negative transfer effects that are likely to prevail from one to the other. This is also likely to affect the level of anxiety experienced in using the computer. The learning literature suggests that these performance and anxiety levels will change as users progress along the learning curve of a particular package or packages. Novices and experts do not perform or respond to the unexpected in like-fashion. Yet this notion of expertise is very vague and not necessarily amenable to slotting into a schema such as proposed by Shneiderman (1987). An expert on one package can be a novice on another, or possess expertise in only certain aspects of the software. Expert/novice differences research has shown that experts in novel situations perform at the same level as novices in these same situations. There is still insufficient understanding of the transition from novice to expert in all areas of study to be able to fully grasp these notions as they apply to the learning of a package. Research, nonetheless, has attempted to come to grips with some of them.

Gilfoil (1982) examined the transition in the use of help across the learning curve. Over time, users' access to help was monitored on a package which allowed progression from the use of menus to commands. It was found that use of the help facility for menus decreased as experience increased in this mode, until the user went into command modes. At that point, help was again sought, regarding the command mode, until this mode was, likewise, mastered. Gugerty and Olson (1986) examined programming differences between expert LOGO and Pascal programmers and novices. Experts were able to debug faster and more accurately, probably due to the fact that they also seem to run program checks more often. Novice programmers, on the other hand, tended to add bugs. As with package use, experts know where and what to look for and how to correct it. Novices compound their problems.

A study by Teng and Jamison (1990) found that subjects with no computer or database experience rated the SQL query language as being more problematic for task resolution than DBASE III, while the opposite opinion was held by those with some experience. In general, users tended to prefer the query language they used last. Both categories of users rated DBASE III more favourably for usability features, and SQL more favourably for functionality features. In terms of performance, no association was found between this and the rating of the languages. It was found, however, that those subjects who learned SQL first, then DBASE III, had lower performance on post-tests using SQL than when using DBASE III. The subjects were 51 MBA students with some work experience. They received online training for the experiment, but the final assessment was based on offline paper-and-pencil query writing tests. Their subjective evaluations were collected from a questionnaire, using semantic differential scales, developed by the authors. Usability features included "frequency of errors, complexity, ease of learning, etc." Functionality included "power of the query language, flexibility, precision". They also included questions on "compatibility of a query language expression with a user's 'natural' way of expressing a request, or with a user's confidence in the success of a query". They did not say, however, whether these were put under the usability or functionality banners. The conclusion they draw from their study is that improving the usability of query languages will improve novice users' perceptions of these languages, but that experienced users additionally need functionality. They suggest two avenues of research. The first is detailed process-tracing to study differences in novice/expert thought processes in information retrieval. The second is refining an evaluation instrument to assess the **'quality'** of a query language.

Using college students with typing experience but no computer text-editing experience, Foss and DeRidder (1989) attempted to study the effects of transfer of learning trom other editors to the DEC editor and from the programming language BASIC to the DEC editor. They found that positive transfer occurred between the editors, but not from BASIC. Contrarily, Karat, et.al. (1986) in their investigation of naive users and users experienced with a wordprocessing system on a different wordprocessing system, found the opposite to be true. The naive users performed better than the users who had prior experience on a wordprocessing system. Ledgard, et.al. (1980) studied users who were inexperienced with text editors, those familiar with them, and those experienced with them. They were interested in whether experience had an effect on performance with different command structures - symbol versus keyword command. They found that keyword commands improved performance, but that this improvement was less marked for experienced users. Users seem able to overcome many deficiencies in design, even if this is at the expense of much time and effort. This adaptability of humans has the potential of lulling designers into a false sense of accomplishment, in the belief that they have created a 'design well done'. Out of need, users must often compensate for bad designs with their own ingenuity.

It has tended to be assumed that menus and icon-based packages will be easier to use than command structures, especially for novices. An interesting finding by Whiteside, et.al. (1985) was that commands proved best for performance, for all classes of users. In fact, novices had great difficulty with the menu systems. Their study investigated seven different database management systems with different command structures - menu, icon, command. Hauptmann and Green (1983) compared menu, command and natural language. Their subjects were programmers with experience ranging from novice to expert, but with no package experience. They found no difference in performance time or accuracy.

In terms of assistance provided, Relles (1979) found that experienced users performed better when provided with only online assistance than with only a printed manual, while novices needed the presence of a printed manual.

ii. Task and Subject Matter

Extent of familiarity with the task to be performed, and the requisite subject matter, may influence user performance and comfort using a software package. Limited knowledge means that an additional learning strain is imposed on the learner, thereby increasing anxiety, as might be exhibited by students in an introductory Statistics course. These students have to learn not only statistical principles but also the operation of the package. It may be that package design, which currently ignores this need, should not, as was spoken about earlier in Chapter 2.A.2. While less research has been focused on these aspects of experience than has been on package and computer experience, nonetheless, there have been some studies on prior-task experience.

Roberts and Moran (1983) studied typists with and without programming experience and found that those without programming experience were slower and spent three times longer in error states. Roth, Bennett and Woods (1987) differentiated their subjects further, into technicians with limited experience in diagnosing faults in electro-mechanical equipment, and limited training on both a previous package and on the expert system being used in the experiment; technicians with extensive experience but limited training on a previous package, technicians with extensive experience and training on both a previous package and the expert system. They found that those with the least and the most experience took an active role interacting with the system and making decisions based on the expert system's diagnoses. The expert system was viewed as a tool to be manipulated in order to arrive at solutions. This led to better and faster solutions. Where misinterpretations resulted, it was from a lack of understanding of the intended purpose of the expert system's request. This is often the case in using software. If the users knew what it was the system required, they probably could supply it. Burns, et.al. (1986) found that flight controllers, trainers and astronauts at NASA's Johnson Space Centre, with prior experience in a particular alphanumeric information screen display, also benefitted from improvements to it, in terms of making fewer errors, though their timing was unaffected. Improvements entailed giving comprehensible names and abbreviations to data items and headings; highlighting categories and grouping data by category and type of information; using consistent placement on particular areas of the screen and aligning data on the screen. Nielsen (1992), in his study of usability, found that usability specialists were better at finding usability problems than those without this experience and that those with additional experience with the particular type of user interface performed even better.

2.D.5.b. **Psychostructure (comfort,effort)**

Users have their own individual psychostructure, comprising Motivational and Affective traits, Learning Style and Cognitive Style, which may have implications for software ease of use design. (Carey, J., 1989; Weiner, 1986). Different perspectives exist as to whether an individual's learning style evokes his cognitive style (Dunn, et. al., 1979), or vice versa (Pask and Scott, 1972). This is probably because neither construct is mutually exclusive of the other. Dunn, et. al., (1979, p. 53) define Learning Style as "the way in which individuals respond to environmental, emotional, sociological, and physical stimuli", and Cognitive Style as "the ways in which responses are made because of individual psychological differences". The term 'cognitive style' will be adopted here to refer to both cognitive style and learning style.

i. Cognitive Style

A number of different dimensions and instruments on which to measure cognitive style can be found in the literature. Pask (1976) in his work categorized learning into Serialist, Holist, and Versatile strategies. **Serialists** are hypothesized to follow a more procedural or step-by-step approach, reminiscent of a bottom-up strategy. **Holists** proceed more globally at the outset, moving in and out from detailed to global perspectives as they proceed, a more top-down approach. **Versatiles** are able to adapt at will, or as the subject matter dictates.

The Kolb instrument (Fox, 1984) classifies learners as **Divergers** who predominantly use feeling and observation, and excel in the area of brainstorming or idea generation; **Assimilators** who operate on thinking and observation, and are good at inductive reasoning and abstract notions; **Convergers** who operate by thinking and doing, and lean towards practical applications; and **Accommodators** who operate by feeling and doing, and tend to get things done.

The Grasha-Riechmann Learning Styles Questionnaire (Charkins, et. al., 1985) categorizes individuals, for both learning and teaching styles, as **Dependent** - preferring straightforward lecture; **Collaborative** - preferring class discussion and group projects; **Independent** - preferring input in the course content and structure, and having the teacher as a resource person.

Witkin (1969) distinguishes **Field Independents** as those who perceive things in terms of their parts, while **Field Dependents** view them as a whole.

The Myers-Briggs instrument (Briggs Myers and McCaulley, 1985) categorizes individuals on four dimensions - **Intuitive/Sensing, Feeling/Thinking, Perception/Judgment, Introversion/Extraversion**, and is based in Jungian Theory. The four dimensions then combine to form sixteen categorizations. The intuitive person is future oriented and looks for underlying meanings, symbols and patterns, and is more theoretically directed. Sensing individuals, in contrast, are more drawn to the concrete, practical and present-minded, relying on observable facts. Feeling individuals weigh the importance and value of possible alternatives, while the thinking person engages in a logical appraisal of causes and effects. The perception person is open to new ideas and to change, while the judgment person prefers organized, planned, clear-cut decisions. Introversion suggests individuals who are more inwardly reflective, while extroversion suggests more outward- seeking individuals.

Ambardar (1984) used the field dependent/independent categorization to investigate differences in database accessing methods. She found that field dependent users preferred to use the sequential item number search method, while field independents preferred keyword search. She also found that both categories of user took longer if they used a non-preferred method. Field independents were able to separate the functional uses of the commands, while field dependents could not.

Using the Myers-Briggs Type Indicator (MBTI), Kern and Matta (1988) discovered that sensing individuals performed significantly better than intuitives on a multiple-choice Lotus$_t$ exam after having completed self-paced instruction in Lotus$_t$. This, they felt, was consistent with the indications of the sensing individual. Being more comfortable with searching for facts, this individual is more likely to have the predisposition necessary for self-paced learning. The extroversion/introversion and thinking/feeling scales did not predict the success of the self-paced instruction. The researchers did not examine the judgment/perception dimension because of studies which indicate that this dimension is correlated with the others. They tested a scale, which combines perception and judging, and which is said to indicate decision style. They found that sensing-thinking students performed better on the test than did intuition-feeling students. The thinking aspect of the individual which implies logical, organized behaviour, seemingly would provide the discipline necessary for self-paced instruction and contribute to these results.

Bostrom, Olfman and Sein (1990) chose the Kolb learning style classification scheme to study individuals performing a budget building task and electronic mailing tasks, using either a VAX mainframe or an IBM PC. Abstracts (convergers and assimilators) performed better than concretes (divergers and accommodators), but not significantly so, and actives (convergers and accommodators) were somewhat better than reflectives (assimilators and divergers). Users were assessed on accuracy, time spent to complete the tasks, and performance on a post-training quiz. This was investigated further in terms of the type of model of the task to be performed with the package which was provided, and the training provided. Abstracts and actives performed better with an abstract model and reflectives with an analogical model. As was discussed previously, the measure to reflect perceived expected future usefulness of the software used, was found to be higher for reflectives and abstracts when matched with application-based training, and for concretes when matched with construct-based training.

Reading ability was found by Bevan (1981) to be a factor in the relationship between performance and the rate at which the screen is filled with text. All ability levels disliked high display rates, but low abilities preferred the lowest rate. Roemer and Chapanis (1982) examined reading ability in relation to written tutorials and found significant differences in task completion times, errors made, and correctness scores. These were due to reading ability and not to the complexity levels of the written tutorials. In investigating spatial memory, Egan and Gomez (1985) found that low scores on the spatial memory test also equated with low performance.

Manual dexterity has implications for the operation of input devices in terms of speed, performance and comfort, whether typing at a keyboard, or in making use of interaction devices, such as mouse, lightpen, touch screen, et cetera. As these studies would fall under hardware interface considerations, and are not being analyzed in this thesis, they are not elaborated on here. However, this issue has implications for the speed and performance measures collected from individuals and used to assess ease of use of the package. The Keystroke-Level Model accounts for this contamination, and is discussed in Section E of this chapter.

ii. Motivation

There are varying motives underlying a user's attempt to learn and use a software package which may also be influencing factors. Whether the motives are compulsory versus voluntary; needed for the task or simply convenient; to be used in the long-term versus only for the short-term; for work, school or personal reasons, these will have different effects on users' motivation. Research has tended to examine the effect that interacting with systems has on motivation, rather than the effect motivation has on performance.

No studies were found which specifically examined motives for software use. When motivation was referred to in studies, it seemed to relate more to users' responsiveness than to why they wanted to use or learn a package. As discussed overleaf, Bostrom, Olfman and Sein (1990) found that individuals, classified as abstracts by the Kolb learning style instrument, indicated more expectation to make future use of the software supplied in the experiment if they had previously received application-based training, while this was equally true for those classified as concretes who were given construct-based training. Ogozalek and Van Praag (1986) discovered that elderly users in their experiment found more pleasure in using a computerized system than younger subjects. Here again, they, too, did not investigate the effects of motivation on performance per se.

iii. Computer Anxiety

This refers to the extent of fear or aversion to computerization and/or interaction with computers that is manifested in people but which should change with exposure to computers. Different factors may influence the level of anxiety evoked. This, in turn, may have an effect on performance and/or perception. Studies that have been done in this domain seem to focus on the factors impacting computer anxiety rather than the effect of computer anxiety on ease of use in terms of performance and perceptions.

A questionnaire survey, conducted by Howard and Smith (1986) to investigate computer anxiety among managers from various firms, found it to be minimal. Age and sex were not correlated with computer anxiety, neither were locus of control, cognitive style, or trait anxiety. Computer anxiety was negatively correlated to favourable attitudes to the computer and its impact on society, and even more so to actual computer experience. High anxiety levels were also found to be related to high Math anxiety levels. In Gilroy and Desai's study (1986), it was found that undergraduate and MBA students, who had used a wordprocessor in an English Composition course over a semester, experienced a reduction in anxiety measures from the beginning to the end of the semester, compared to those who had, instead, followed a programming course. Both groups experienced less anxiety than those who had had no exposure to computers. Women seemed to exhibit higher anxiety levels than men, while race and age were not found to have significant effects.

2.D.5.c. **Demographics (comfort,effort)**

It has been suggested that other factors influencing use and learning of software packages may be a consequence of differing educational levels, sex, age, work experience, social, familial, scholastic or work environments, et cetera. (Borgman, 1986). For instance, the controversial notion exists that women may be more creative than men, who are supposedly more analytically oriented, (the left/right brain hemisphere dichotomy, Gazzaniga and Ledoux, 1978). Given this, and assuming that a basis exists for construing that the use of a software package to perform analyses is more a creative than an analytical process, are women more likely than men to get better performance measures when learning to use a package, or is the reverse true? Environments fostering and promoting exposure to, and liking for innovation and novelty may also influence users' perceptions and, hence, performance.

Ogozalek and VanPraag (1986) investigated the desirability of voice input over keyboard entry for those they termed elderly versus younger individuals. Age and input device made no difference to performance, though all preferred voice input. Younger individuals showed less enthusiasm for using a computerized system and viewed it in terms of a work setting. Older individuals thought in terms of personal use. Gomez, Egan and Bowers (1986), however, did find performance differences due to age. Performance was higher for lower age categories. Academic majors, in another study, were found to produce differences in online library searching. Borgman (1986, 84) found that science and engineering majors were more successful at these searches than were humanities and social science majors. Interestingly, the most frequent library users had the most difficulty with online searching. Obviously, their familiarity with manual search procedures gave negative transfer effects which interfered with learning the new system.

2.D.6. **Instructional Strategy**

Users may adopt various strategies to learn a software package, ranging from strict following of regulated, tutorial-based instruction to completely self-directed, free-form learning and exploration. Most learning involves some mix of this spectrum, but there is an indication that individuals favour the self-directed approach. Exploratory research, conducted by Carroll and Mack (1984) on wordprocessing tasks on a text-editor, indicated that users have a strong preference for self-exploratory learning over the use of manuals. Another exploratory study, this one done by Hiltz and Kerr (1986), likewise found trial and error to be the most favoured learning method. They also found that users skimmed whatever documentation was made available, whether online or offline and, in fact, rarely referred to offline documentation at all. Surprisingly, they found formal training took longer for mastery level performance to be attained. With novices, it was found that providing a guided online tutorial increased the chance of subsequent use.

In terms of instructional material, research has concerned itself mostly with various forms of scripted instruction. This has been due mainly to our current level of knowledge and technology. Bostrom, Olfman and Sein (1990) compared application-based training with construct-based training, that is, problem-oriented training versus training based on syntax and functions of the package They found that those users who were reflectives and abstracts, (Kolb's instrument, Fox, 1984), had greater expectations of making future use of the software in the experiment with application-based training, while concretes indicated this with construct-based training. Self-paced instruction proved better for sensing individuals than for intuitives, as presented in Kern and Matta's (1988) study based on the Myers-Briggs Type Indicator classification of individuals. Black, et. al. (1989) looked at the effect of various forms of instruction on the performance of students in Grades 10 and 11 who had minimum computer experience. Learning and test time, and number of errors were measured on a post-instruction test, in which subjects were required to create a database. The instructional methods were all forms of scripted instruction. There were four methods - material which went from general descriptions to specific instructions; material which provided an explanation along with specific instructions; specific explanation of the system and the package; and explicit instructions. Explicit instructions accelerated initial learning, though task completion was faster and more accurate with general to specific instructions. The material containing an explanation with specific instructions resulted in less errors than either the specific-specific or explicit instruction groups. Holt, et.al. (1986) divided their training into general global explanations, detailed step-by-step instructions, and combined global and detail. The global instruction was inferior to both detailed and combined material. Teaching users the underlying conceptual model of the system was found by Bayman and Mayer (1984) to be another way to improve performance.

Cohill and Williges (1982) provided users with either no assistance, online assistance, or printed manual only. Those with no assistance performed the worst. The assisted groups were formed of combinations of user versus computer initiation of help; hard copy versus online manuals; user versus computer selection of help material retrieved. Performance was best when the user controlled the initiation of help and the material viewed, and had access to a printed manual, than when the computer took the lead. Borenstein (1985) compared UNIX manual versus natural language manual versus human assistance. It was found that having human assistance speeded performance and, not surprisingly, that the content of the manual was more important than the delivery vehicle. Cordes (1984) found that subjects working on various command structures, for example, original form-fill-in, improved form, commands, relied on online help even in those cases where a manual was provided. A form-fill-in presents to the user a blank or filled in form of the finished product on the screen which the user may then use, either to enter input or modify that which is shown. The original form and an improved version (it was not specified in what way) of this input method were compared to the use of commands. Watley and Mulford (1983), however, found that performance was better, that is, took less time and resulted in greater comprehension, with a hard copy manual than with online assistance. Dunsmore (1980) provided novice users with a brief offline summary description only of the system capabilities, brief summary plus online manual, or brief summary plus offline manual. The brief summary with the offline manual resulted in the best performance, while that using the online manual was the worst. Relles (1979) had the same findings for novices, who he found performed poorly with online assistance compared to having a manual only. Users with computer experience, however, performed better with online assistance than with a manual only. In line with these results, Czaja (1986) found that training via online tutorials produced worse performance results on transfer editing tasks, which subjects performed on an actual wordprocessor, than did training via either document-based or instructor-based offline tutorials. The online tutorial group made errors which seemed to indicate that they had not assimilated the operations required to perform basic editing procedures.

Timing of instruction has also been found to be a factor in performance levels. Maskery (1985) found weekly sessions promoted better learning than did daily sessions. Further, a break of five to six weeks led to an initial decrement in learning, followed by rapid improvement. This is consistent with the findings in a study by Bailey (1989). In the learning literature, frequency and duration of exposure are important ingredients for learning. (Favaro, 1986). A factor which has not been given much consideration, is the effect that training sessions on request may have on learning, compared to compulsory and/or scheduled sessions.

2.D.7. <u>User Role</u>

In interacting with a package, users can adopt either a passive or an active role. Current commercial packages impose a passive approach, although research is increasingly searching for ways to accommodate a more active role. Maskery (1985) investigated a statistical tool with an adaptive interface. The package progressed through a sequence, wherein one found the package leading with the user forced to make choices; the package leading with the user having free choice; and the user leading and having free choice. However, the adaptive nature of the design, from what seems to be menus to commands, confused users. The package determines which of three levels of interface to present to the user, based on questions posed on entry to the package. The second level is presented once the user has completed one task at level one and has entered the 'TERSE' command. The user is instructed how to circumvent the other two levels at sign-on, after having completed a session using level 2. The package Merlin, has a similar interface, in that users are presented with a menu at sign-on if the package does not detect an existing datafile. If a file exists, the user is lead straight to command mode. From either situation, the user is able to move between menu access and command mode. Use of this package in the field has not proved problematic for its users.

This portion of the chapter presented a cross section of factors which impact ease of use of software, with the associated research findings. The methodological approaches employed are discussed in the section following.

2.E. Methodologies to Study Ease of Use

McGrath (1964) classifies research methodologies into four categories: **field studies, experimental simulations, laboratory experiments, and computer simulations**. These can be arrayed along a spectrum which goes from exploratory, hypothesis formulation research on one end to confirmatory, hypothesis testing on the other.

Field studies attempt to maintain as natural a setting as possible, with minimum interference from the researcher. Interest here is in unearthing the factors contributing to a phenomenon of interest, and which are, as yet, not clearly understood or identified. There is interest also in maintaining the richness of the interplay of variables which, on the negative side, makes it difficult to attribute cause and effect, that is, identify particular variables as impacting the variable of study.

Experimental simulations try to reproduce "real" conditions in a laboratory setting and to investigate the phenomenon of interest within this context. Researcher interference is still at a minimum, allowing the variables from the subjects' natural behaviours to emerge and to interact.

In **laboratory experiments**, reality is not recreated, but rather variables which have been identified as influencing a phenomenon, are manipulated by the experimenter in order to determine their effect on this phenomenon. Confounding of variables is minimized as the researcher attempts, as much as possible, to control those variables not under investigation.

Computer simulations use mathematical models to represent specific aspects of reality, probabilistically varying the variables under consideration. Human subjects may or may not be part of the experimental process. This latter approach gives the experimenter the greatest control of the experimental variables and situation, but suffers most in terms of generalizability of findings to the "real" setting.

In investigating the human factors research, presented before in section D, one finds that the research has centred mainly on laboratory experiments, which attempts to control variables which have been identified and to analyze the impact of other variables on the phenomena of study. This is surprising since the nature of the variables involved in human factor research are not readily amenable to isolation and control. Additionally, the concept of ease of use has not yet been fully defined. One would, instead, expect to see more focus on exploratory, hypothesis formulation research. In fact, one is hard-pressed to find research using the methodologies at either extreme on the spectrum, that is, field-studies and computer simulations.

Various research methodologies were discussed in this section. The techniques and approaches to measuring ease of use are discussed in the next.

2.F. Evaluation and Measurement of Ease of Use

There are two main approaches which may be used when trying to assess the effectiveness of various design factors in achieving **ease of use**:

1. <u>Subjective - Questionnaire Elicitation</u>

a. based on characteristics of the package - degree to which package has or does not have the characteristics considered essential;

b. based on general reactions to the package - degree to which users like or do not like the package.

2. **Objective - Measures of Performance/Predictive Models**

a. based on actual performance measures - time taken to complete a task, errors made, help sought, et cetera;

b. based on predictions of performance - using models, such as the Keystroke-Level Model of Card, Moran and Newell (1980) and the Production Model of Polson and Kieras (1985), to provide a theoretical minimum time needed to complete any task on a given package

Although these measures may seem definitive of performance, Travers (1975, p. 47) cautions that outcomes are contingent on the conditions under which learning takes place and should be considered when assessing the extent of learning. These conditions include:

a) time distribution - time required to teach and learn using the methods employed;
b) task characteristics - easy or difficult to learn and in what respects;
c) consequences of erroneous responses;
d) cues provided - when and what provided;
e) incentives provided to stimulate motivation to learn;
f) other unidentified conditions - properly designed equipment, social climate, etc.

Measurement of ease of use or learning is evidently not a straightforward process. The approaches adopted in the literature will be presented in the following sections.

2.F.1. Subjective - Questionnaire Elicitation

2.F.1.a. Based on package characteristics

The impact on ease of use of various design factors summarized in this chapter is often viewed as being measurable and consistent for all users, as the various magazine reviews suggest. However, our discussion so far tends to refute this conclusion. Evaluation is more likely to be a subjective matter, being based on the user's level of expertise and preference. At this stage, the instruments available to assist in these evaluations are not adequate. Academic and technical journals and magazines contain articles describing usually informally constructed instruments. For example, Scriven's (1990) article discussed in Chapter 2.D., reports an evaluation for the wordprocessor WriteNow in which he used his proposed framework. It lists Resource Requirements - Low; Ease of Learning - B; Ease of Use - A-; Speed - A+; Power - B+; Safety - A+; Support - A-; Value - A; Ranking on merit - 1; Ranking on value - 1. Teng and Jamison (1990), for their part, list the rating items in their questionnaire on database language queries as: Simplicity of language; Debugging ease; Freedom from errors; Naturalness of language; Ease of understanding queries; Power of language; Precision of queries; Confidence in results; Ease of learning; Flexibility of language; Ease of query planning; Ease of translation from English; Succinctness of queries; Global ease of use. Subjects then rated the languages for these attributes on a semantic differential scale. Roberts and Moran (1983), evaluated the text-editors in their experiment against a checklist of editing tasks. An expert was asked to rate the packages based on how well they performed the various functions listed, either efficiently, awkwardly, or not at all.

In general, the instruments which do exist, address only the degree of existence or non-existence of a particular design feature, and often includes **Ease of Use** and/or **Ease of Learning** as one on a list of attributes to be assessed. The evaluation is usually one-dimensional, and not based on the multiple underlying factors which promote **learning and/or ease of use - memory, speed, effort, comfort,** as is being advocated in this thesis.

One study which was somewhat similar to our own, was conducted by Holcomb and Tharp (1991). They devised a list of usability attributes to which users assign ranks. Still, however, the evaluation is uni-dimensional and it does not capture the relative quality of the features of the package as an integrated whole. Their evaluation is also meant to assess usability, which as defined by them, does not equate with **ease of use**. Their list, therefore, includes features not relevant to ease of use, such as Reliability, Installation, Field Maintenance, Serviceability, and Advertising, and excludes others that are, such as Command Structure. They also did not separate Design from Assistance Features, and considered online assistance as a single attribute, rather than composed of many attributes.

2.F.1.b. **Based on general reactions**

Most existing instruments which measure general user perceptions were developed in line with early user involvement with computers, in which users made use of the output of a package, but did not interact with the computer to input data and produce output. As Doll and Torkzadeh (1988, p. 260) state,

> "Indeed, user information satisfaction instruments have not been designed or validated for measuring end-user satisfaction. They focus on general satisfaction rather than on a specific application and they omit aspects important to end-user computing such as ease of use."

Ease of use is only one of the contributing factors to user satisfaction. User satisfaction measures include issues external to the operation of the package itself, such as user involvement, user attitudes to change, quality and service of the MIS function, quality of output, timeliness and volume of information, et cetera, and are not relevant for this work. (Doll and Torkzadeh, 1988; Bailey and Pearson, 1983; Ives, Olson and Baroudi, 1983). Only those aspects of these instruments which relate to ease of use will be mentioned here.

The instrument developed by Davis (1989) measures ease of use as a function of subjective assessment of ease of learning, 'controllableness', clarity and understandability, flexibility, ease of becoming skillful and general ease of use. The final instrument asked six questions, 'Learning to operate Chart-Master would be easy for me', 'I would find it easy to get Chart-Master to do what I want it to do', 'My interaction with Chart-Master would be clear and understandable', 'I would find Chart-Master to be flexible to interact with', 'It would be easy for me to become skillful at using Chart-Master', 'I would find Chart-Master easy to use'. Davis, Bagozzi and Warshaw (1989, p. 991) measured ease of use on four items, 'Learning to operate WriteOne would be easy for me', 'I would find it easy to get WriteOne to do what I want it to do', 'It would be easy for me to become skillful at using WriteOne', and 'I would find WriteOne easy to use'. Doll and Torkzadeh's (1988, p. 268) instrument measured ease of use on two items, 'Is the system user-friendly?' and 'Is the system easy to use?'. Khalifa's (1990) instrument to measure user satisfaction with two artificially created interfaces, used a mix of comparative questions, 'Which interface do you think is more difficult to learn?', 'Once learned, which interface is easier to use?'; questions using nine-point semantic differential scales, 'Try to rate the learning speed for each interface', 'Try to rate the use speed for each interface (once learned)', 'Try to locate each interface on this satisfaction scale', 'Which interface do you prefer to use?'; and an open-ended question, 'Why?'.

These instruments measure general ease of use and only on very few items, and none of them address specific package features or the support of learning factors.

2.F.2. Objective - Measures of Performance/Predictive Models

2.F.2.a. **Based on performance measures**

Measures of actual performance are provided by Schneier and Mehal (1984), Harmon (1987), Guillemette (1989), and Borgman (1986). These include:

- Time on Task/Speed of Completion;
- Rate of Performance (time to attain mastery level);
- Warm up time after period of non-use;
- Errors Made - frequency and type - category and severity;
- Ability to recover from errors;
- Help Sought - frequency and type;
- Sequence Followed (optimality);
- Level of Performance Quality;
- Cost of Performance.

While the transition from novice to expert performance is usually accompanied by an increase in the speed with which an activity is completed, it may also be negatively correlated with accuracy in performance. A user can conceivably move rapidly through a software package while making numerous mistakes, but recover from these errors speedily because of knowledge of the software. In addition, while it is expected that a decrease in the number and severity of errors made and references to help should appear as mastery increases, mastery may also be reflected in the speed of recovery from error digressions.

Successful performance with a package will be reflected in the sequence followed by the user, and in the frequency and typ^r of errors made, assistance sought, and recovery made. Effective performance, in this case, refers to achieving the desired output, not necessarily using the optimal alternative. Efficient performance implies arriving at the correct solution using the fastest, most accurate, direct route.

The more efficiently and effectively a user moves from novice to expert levels should also translate into cost savings, both tangible and intangible. This would be evident in time saved as a result of progress from what is perceived to be relatively less productive training to increased productive application of this training, reflected in the rate of performance and quality of performance. Not only can these measures be assessed, as can the subjective measures, relative to the users' particular psychostructure and/or demographic profiles, but different methods of measurement are also possible. All of these possibilities add to the complexity involved in measuring software ease of use.

The methodology espoused by Roberts and Moran (1983) for evaluating text editors looks at four measures - the time it takes an expert to perform basic benchmark editing tasks; their time spent in error states; the time it takes a novice to learn to do the tasks; and assessed functionality of the package for more complex tasks. These measures are defined as:

Expert Learning:

Error time = Time spent in error states
Error-free time = Performance time minus the error time

A second pass through the task is made by the subjects to correct any remaining errors and this is included in the measurement of error and is given as:

Individual error score = Total error-free time / the number of editing sub-tasks performed

These measures are also compared to the expert performance time predicted by the Keystroke-Level Model.

Novice learning :

$$\frac{\text{amount of time spent in five self directed instruction-quiz cycles}}{\text{total number of tasks performed in post-training quizzes}}$$

Functionality:

The nine text editors which were used to test their methodology were evaluated on functionality based on an experienced user's rating of each package against a checklist of common editing tasks. This was done on the basis of whether the editor could perform the function efficiently, awkwardly, or not at all.

The authors found that the methodology was able to differentiate between the editors, and that no editor was superior in all the dimensions measured.

Murphy (1992), in his dissertation, recognizes the transitionality of learning a package interface. The learner can easily be in a situation which requires a move from performing a previously learnt set of operations to learning a new set. The measures proposed are learning effectiveness, learning retention, learning efficiency. These are based on using performance measures of one interface, in terms of iterations to achieve mastery and time required to do so, as a baseline reference for assessing another.

Learning Effectiveness is given as:

$$\frac{2*\text{OL REF} - \text{OL TEST}}{\text{OL REF}} \quad \text{where,}$$

OL REF is the number of iterations or time spent to achieve mastery at original learning session with baseline reference interface;

OL TEST is the number of iterations or time spent for interface being tested.

Learning Retention:

The same equation is used to measure retention, replacing the original learning measures with the measures collected at subsequent uses of the interface.

Learning Efficiency:

This measures the amount of iterations required to achieve the same level of performance attained at the original learning session of the interface. This is given as:

$$\frac{1 + \text{\# of iterations at original learning} - \text{\# of iterations at subsequent use}}{\text{\# of iterations at original learning}}$$

The author used this method to test five different interfaces - user-selected icon plus user-generated text label, system-provided icon plus user-generated text label, system-provided icon only, system-provided icon plus system-provided text label, user-selected icons only. Results indicate the method was able to detect differences between the interfaces.

The study done by Khalifa (1990) aimed at predicting ease of learning of a particular interface design. He defines ease of learning as a function of the complexity of the skill being acquired and the computer expertise of the learner. Two artificially created interface designs, one simple, one complex, were presented to novice and experienced computer users. The simple design permitted the same method, as the complex design, of selection of objects and operations from the display, but which remained on the screen. The complex design required the user to recall the methods of selection, recall being deemed more difficult than recognition, and enabled the task to be accomplished in less steps. The task required the use of a mouse-pointing device to add and connect nodes to create predefined networks. He hypothesized that the "complex" interface would be more difficult to learn than the "simple" interface, but would be easier to use once learned, hence preferred by experts. The "simple" interface would be preferred by novices. He measured learning time as:

Learning
Time:

Reading Time of manual + (Performance Time in training session - Performance Time in performance session)

Learning time was put equal to reading time in the event that the time in the performance session was greater than the time in the training session. Performance time was considered a measure of ease of use. Using these measures, he found support for his hypothesis. It should be noted, however, that an accommodation for differences in reading level was not accounted for, and that the interfaces were artificially created.

Barnard and Grudin (1988) suggest measuring user performance based on: time to completion; errors; efficiency - number of keystrokes, number of commands yielding improvements minus the number degrading performance divided by the total number of commands; learning difficulty - number of trials to criterion performance level, use of online help; retention - memory test, transfer test; difficulty of use - subjective assessment, comparison of alternatives.

2.F.2.b. **Based on predictive models**

The two predictive models of performance most frequently cited in the literature, and still offered in the 1990 edition of Human-Computer Interaction (Preece and Keller, 1990) as the definitive models for evaluation of ease of use, are presented here. The Keystroke-Level Model of Card, Moran and Newell (1980) predicts expert learning time to be a function of number of keystrokes to perform an operation, time to perform each keystroke, the mental effort to formulate and execute the operation, system response time, and factors associated with pointing devices. The Production Model of Polson and Kieras (1985) predicts learning time to be a function of the number of new production rules which have to be learnt. This model is also based on the GOMS model (Card, Moran and Newell, 1983) which decomposes tasks into goals, operations, methods of execution, and selection criteria for choosing methods which can be represented by production rules.

Neither of these models accounts for novice performance nor errors in performance (Karat, 1988). Additionally, they are not appropriate for package assessment by practitioners. These models may be appropriate in the design stages to evaluate different systems and to provide a baseline for actual use, but they require far too much time and technical skill to be used effectively by the uninitiated. They are not likely to be entertained as a potential evaluation tool in the business environment. At best, managers are probably only willing to expend enough time to make checks on a checklist of features. They are not likely to want to expend time on any quantitative assessment other than those related to the cost/benefits of the product.

The section following presents a summary of the existing research findings which were discussed previously in this chapter, in terms of software design characteristics, assistance characteristics, user characteristics, and methodologies and measurements employed in conducting these studies.

2.G. Summary of Existing Research Findings

2.G.1. Design Features - (Command Structure)

Several studies have examined the relative merits of the various command structures for different levels of users, but the findings have not been conclusive, neither with respect to preference nor performance. Some studies show menus to be more suited to novices than to experts, while others show novices experiencing difficulty with a menu system and preferring to rely on commands. Still other studies indicate a preference for voice interaction, and in some instances, a mixed system. Not surprisingly, the need for consistency in design is indicated in some of the studies, though this was not the focus of these research undertakings. (Foss and DeRidder, 1989; Napier, et. al., 1989; Mack, Lewis and Carroll, 1987; Ziegler, et. al., 1986; Lee, et. al., 1986; Whiteside, et. al., 1985; Maskery, 1985; Cordes, 1984; Perlman, 1984; Shutoh, 1984; Murray, et. al., 1983; Hauptmann and Green, 1983; Gilfoil, 1982; Tombaugh and McEwen, 1982; Dunsmore, 1981; Walther and O'Neil, 1974).

While performance and preference seem to be enhanced by natural language, users attribute different word meanings to terms used for commands. This causes obvious difficulties in the choice of command names if a standard is to be established for use across software packages. It has been found that these meanings can be taught without impeding performance, even when arbitrary names are assigned. This outcome may be due, however, to the concentrated mental effort at work under experimental conditions, and therefore, this retention may not be maintainable over longer time periods. (Grudin and Barnard, 1985; Landauer, et. al., 1983; Dumais and Landauer, 1983; Barnard, et. al., 1982; Black and

Moran, 1982; Ehrenreich and Porcu, 1982; Ledgard, et. al., 1980; Moses, Mendez and Ehrenreich, 1980). Along these lines, in studies by Burns, et. al. (1986) and Badre (1982), it was found that performance can be improved if information is meaningfully grouped, and in an order which is familiar to the user. This also has implications for the design of menus.

In terms of display and system response rates, research indicates that thresholds exist for the rates at which text is displayed and system responses are made above and below which performance diminishes. (Dannenbring, 1983; Smith, 1983; Bevan, 1981; Dunsmore, 1981; Long, 1976).

2.G.2. Assistance Features

A number of studies have investigated the level of assistance required to facilitate learning of software packages, and the results seem to indicate that step-by-step instructions and training on subsets of the package features aid initial learning, but can hamper transfer of learning. This seems to imply a need for both general, conceptual explanations, as well as specific, step-by-step, procedural instructions. (Black, et. al., 1989; Carroll and Aaronson, 1988; Catrambone and Carroll, 1987; Elkerton and Williges, 1987; Holt, et. al., 1986; Schlager and Ogden, 1986; Charney and Reder, 1986; Czaja, 1986; Carroll, 1985; Carroll and Kay, 1985; Walther and O;Neil, 1974).

Borenstein's study (1985) indicates that the content of the instruction is more important for learning than is the mode of delivery. In their study, Hiltz and Kerr (1986) found, from users' self-reports of learning methods, that offline documentation is the least adopted method of learning, with users preferring human assistance. This study investigated learning an electronic mail system which is specifically designed to minimize reference to offline documentation, so this may account for the findings. However, Borenstein's study (1985) also indicates a preference for a human trainer. In other studies, the results indicate that, in general, users perform better with manuals, but tend to prefer and to choose online self-discovery learning. (Carroll and Mack, 1984; Cordes, 1984; Watley and Mulford, 1983; Magers, 1983; Roemer and Chapanis, 1982; Cohill and Williges, 1982; Dunsmore, 1980; Relles, 1979). A consequence of self-discovery learning is that users learn and use more advanced features than seems to be the case with more formalized approaches. (Nakatani, et. al., 1987; Hiltz and Kerr, 1986). Because of the tradeoff often necessary between time and mastery learning, users will tend to forego learning advanced features and stick to the tried and true, even if it is less efficient. (Czaja, 1986).

Studies by Goodwin and Sanati (1986) and Egan, et. al. (1987) show that if the training material indicates direct relevance of the software to a task related to the users' requirements, this facilitates transfer learning, as does feedback pertinent to the task being performed at that time.

The tone and specificity of system messages have been shown, in four experiments conducted by Shneiderman (1982), to have an effect on performance. Clear, specific, courteous messages improved performance. Apart from the benefits of the clarity of the messages, there is some indication as well that this contributes to reducing computer anxiety. The use of less obscure technical terminology resulted in the interaction being perceived as less intimidating and threatening.

2.G.3. User Characteristics

A study by Evans and Simkin (1989) on the human factors influencing performance was unable to find any demographic, behavioural, cognitive or problem- solving factors which could best predict performance. Research has also not produced consistent results on the use and value of help, though experts seem better able to know when to seek help, and to understand the help received. They are better able to monitor themselves, exhibit a higher level of understanding, and know more complex procedures than novices, which is consistent with expert/novice cognitive literature findings. (Doane, 1986; Gugerty and Olson, 1986; Elkerton and Williges, 1984; Gilfoil, 1982; Relles, 1979).

In terms of transfer of expertise, prior package experience seems to produce more negative than positive transfer effects. (Borgman, 1986, 1984). In one study (Maskery, 1985), timing of the training was shown to be also important for retention and for transfer, again being consistent with findings in the learning literature.

There is some indication that intellectual capacity, as well as prior experience on the task, have an effect on performance, as does the matching of learning styles with type of assistance and interface. These finding, however, have not been firmly established. (Bostrom, Olfman and Sein, 1990; Kern and Matta, 1988; Roth, Bennett and Woods, 1987; Borgman, 1986, 1984; Gomez, Egan and Bowers, 1986; Egan and Gomez, 1985; Ambardar, 1984; Roberts and Moran, 1983; Roemer and Chapanis, 1982; Bevan, 1981).

In general, it is assumed that adults tend to resist learning new things. Notwithstanding, in one study, age was investigated and found to impact motivation more than performance, with older individuals showing greater enthusiasm towards computers than their younger counterparts. (Ogozalek and VanPraag, 1986; Gomez, Egan and Bowers, 1986).

The advisability of a particular command structure for a particular level of user has not been proven in the research. The findings are quite mixed, with some studies advocating menus for novices and commands for experts, while others indicate command structures advisable for both groups. One study found novices had difficulties with menus. This may have been a function of the wording of the menu and the levels of the menu which had to be navigated. The direction of the performance measures are also not always consistent with users' perceptions and preferences. (Karat, et. al., 1986; Ledgard, et. al., 1980; Burns, et. al., 1986; Whiteside, et. al., 1985; Elkerton and Williges, 1984; Murray, et. al., 1983; Gilfoil, 1982).

Howard and Kernan (1990) identify computer anxiety, alienation and attitudes as being distinctly separate constructs which may affect users' performance with software. They argue that past studies have mixed these constructs which has resulted in conflicting findings. Studies by Igbaria (1990) and Gilroy and Desai (1986) indicate that performance can be improved if computer anxiety is addressed in training and attempts made to overcome it. The studies by Shneiderman (1982) also indicate that using less threatening terminology in system messages can reduce anxiety and improve performance. It would also seem that women are more plagued by this distress.

In line with conventional wisdom, 'improvements' in design generally benefit all categories and levels of users, though it is difficult to assess just what constitutes improvement. In the studies cited, these are not always articulated. (Burns, et. al., 1986).

2.G.4. Methodologies Employed in Studies

Most of the research has been experimental in nature. Only a few exploratory studies can be found. The approaches to the various methodological issues, reported in the literature, are summarized below:

2.G.4.a. Software classes/types

Research has focused on investigating general-purpose packages, and especially on text editing systems. A small number have examined spreadsheets, database packages or statistical packages. Very little has been done on specific-purpose packages, or on comparisons to general-purpose packages. Systems used in the experiments are often specifically designed for the experiment or are sub-modules of a full-system, in order to control certain aspects of the design. A lesser portion of the research makes use of full-fledged commercial software.

2.G.4.b. Task

A complete task using a software package usually involves some aspect of inputting or retrieval, editing, manipulation or analysis, storage, and outputting. The activity normally entails getting into the program, attempting to get the program to perform a task which usually implies providing it with data, editing and manipulating that data, performing analyses on that data, viewing and outputting these data and analyses, and finally, terminating the program. Most tasks employed in the research have focused on some aspect of a structured unit task which explores only subsets of the complete task, such as deleting, searching a database, editing an existent document, et cetera. This limits investigation to specific operations of the package, rather than extending it to the overall use of the package that most users will encounter.

2.G.4.c. Context

Most of the experiments have been carried out under controlled laboratory conditions. Some have been conducted in uncontrolled laboratory conditions, but very few studies have been done in the field.

2.G.4.d. Subjects

The majority of subjects are taken from the university student population, though some are taken from the working world. The student subjects are usually taken from among those who will be entering the business environment and making use of the software tools used in the experiments. Given the increasing use of computers and software by users of every age, category and situation, for both work and personal use, these subjects can be considered representative samples of the user population, except in those cases where task experience and motivation may likely be contributing factors. Attempts have been made to gather information on the effects of prior package and subject matter experience from a variety of expertise ranges on learning and using packages. Research focuses most heavily on novices, followed by experts, with very little on those with intermediate proficiency.

2.G.4.e. Collection methods

Recording and video taping of protocols and keystroke traces are the most commonly used collection media. In some instances, where systems allow, keystroke traces and timings are collected online by the system. Written, online and verbal post-testing are also often administered to collect proficiency scores. Informally constructed verbal or written questionnaires, the contents of which are frequently not reported, are the usual means of soliciting user reactions, experience levels and demographic information.

2.G.4.f. Ease of use measures

Performance time is the most widely used measure, supported by error and help analyses. Different methods for measuring these are often used and different suggestions for refining them have been proposed by a number of researchers, as discussed previously in Chapter 2.E. This makes comparison of measures difficult, if not impossible. Users' subjective evaluations are often collected as well, using informally constructed instruments with few items. This makes them suspect in terms of the reliability and validity of the constructs being measured.

A glance at this summary of research studies is sufficient to realize that research in the area of software ease of use is weak as a whole, and no less so than in the area of measurement. This is elaborated on in the following section.

2.H. Limitations of Existing Research

Research is still not at the stage where it can be said that a cohesive body of knowledge exists. It is still hodge-podge, and based mainly on conventional wisdom. The different perspectives taken on **ease of use** make it difficult to arrive at a measure of **ease of use** and to compare findings. As discussed previously, some researchers distinguish between **ease of use** and **ease of learning**, whereby **ease of use** refers to expert performance and **ease of learning** refers to novice performance. (Khalifa, 1990). On the other hand, other researchers consider **ease of learning** to be a subset of **ease of use**. (Davis, 1989; Roberts and Moran, 1983; Whiteside, et.al., (1985); Carroll and Mack (1984). Additionally, they attribute different components to what facilitates this **ease of use/learning**, which results from the support given to mental activity, physical activity, and psychological comfort.

Most of the research areas, identified by our framework as having an impact on **ease of use**, have had only limited study. This indicates a great need for replicative research in order to confirm findings. One of the problems is that the basis on which research is conducted varies from researcher to researcher and has, hitherto, not been identified within the context of a comprehensive framework. Without this context, confirmation is going to be difficult to assert. The research tends to address features of the package being studied in isolation in the context of a laboratory experiment, as opposed to treating the package as an integrated whole. It is evident that the nature of the variables being studied is not readily amenable to being isolated. It is always questionable whether, once these isolated features are integrated in the context of a complete package design, the results will still hold. Only a few exploratory studies can be found, and these were conducted using only a small number of subjects, in restrictive conditions.

The difficulty in doing research in this area arises from the number of factors which may contribute to ease of use, and the complexity of the interplay between and among them, which the framework presented in Chapter 2.C. amply demonstrates. This makes it difficult to control and isolate variables, and to reliably test and report on them. In turn, this leads to fragmented research in which any single factor is examined by only a few studies. This, with the differing experimental settings, tasks, users groups, measurement techniques and instruments, et cetera, makes comparison of findings difficult.

Especially difficult to assess, and therefore to compare, is the degree to which the features examined adhere to 'good' design principles, or even what these may be. Also, the fact that some features, for example, Command Structure, are package dependent, makes it difficult, if not impossible to compare across packages.

Additional problems arise in comparing findings involving menu structures. Studies often do not differentiate between the different types of menu structures and the user category likely to be assisted by each. For instance, no distinction is made between menus which guide the user through predetermined, usual paths, and undirected menus, such as those offered in most spreadsheet packages. This latter type of menu provides, essentially, a memory jog of commands, but does not give assistance in the sequence of operations required. This guidance, or lack of it, is likely to impose different requirements on the user. Unguided menus tend to be designed as an assistance once expertise is gained and is, therefore, usually used and preferred by experts. On the other hand, one would expect guided menus to provide more support and assistance to novices. This guidance is likely to frustrate experts. Some of the contradictions in the findings on Command Structures may be a consequence of this differentiation not being made.

The issue of measurement of ease of use is an important one which has been largely ignored in the literature. Most of the measurement instruments found in the literature are not designed to address the measurement of ease of use specifically. Although Doll and Torkzadeh (1988) claim to rectify this situation, their instrument only included two items pertaining to ease of use. (Davis, 1989; Davis, Bagozzi and Warshaw, 1989). None of these instruments looks at specific package features. Articles in technical journals and magazines may offer a checklist of package features to assess existence or non-existence of the feature, but do not relate them to the factors which support learning. Instead, **Ease of Use** is usually considered just another one of the attributes of the package, and not a result of particular features being, or not being, in the package. (Scriven, 1990; Teng and Jamison, 1990).

Caution is warranted in the use being made of quantitative measures, such as time, errors and help calls made, as exclusive indicators of ease of use. For instance, there are tradeoffs to be made between speed and functionality. The more complex the tasks permitted by the software, the more complex the package has to be. This will have an impact on the speed and difficulty with which it is learnt. Additionally, the complexity and intricacy of package design dictates that ease of use should not be measured solely on one measure, and certainly, not solely on time, error and help call factors. Distraction time is not usually alluded to in research but may be an important consideration in using time, error and help calls as measures. In the performance of a task, it is assumed that the time elapsed, errors and help calls made are directly related to the thinking about and performing of the operations of the task. This may not always be true, but, instead, result from diversions caused by daydreaming, preoccupation with other matters, responses to environmental factors, et cetera. Laboratory studies have greater control over these factors than do field studies, but studies of all types should be aware of the potential interference that may enter the data collected.

It was discussed earlier that it is not possible to rely solely on predictive models. (Card, Moran and Newell, 1980; Polson and Kieras, 1985). The current models do not account for novice performance, and this lack of understanding is a major stumbling block to users attaining package expertise quickly and effortlessly. Software designers, as yet, do not know how best to provide assistance for this transition. While these models can be helpful in providing a baseline for performance, they are too unwieldy and technical to be used in the everyday business environment in order to make rapid package selections. The various tools which do exist are useful in the initial and beta-testing stages of software design and development, but they cannot meet the needs of a manager, or other business user, who must choose from a myriad of packages in the same class, such as choosing from the variety of spreadsheets or databases available on the market.

Another cautionary note must be sounded with respect to the comparison of findings concerning experience levels. Currently, no adequate criteria scheme exists on which to categorize and classify expertise in this domain. This may account for the lack of consistent and reliable results in this area. Some users classified in experiments as experts might perhaps be more accurately classified as intermediates. It may be wise to establish novice and beginner categories as distinct from intermediates. Also, it may be necessary to differentiate experience based on familiarity with other packages of like type and/or operation.

So far, the consensus seems to be that there is no consensus. The only results which can be said to demonstrate some stability seem to be those which hold that there is an optimal reading rate, and that there is the need for restricting the available features to which users, with no prior experience with computers and packages, should have access. This latter finding was not supported in practice, however, for one software manufacturer, Informix Software, found it necessary to remove this restriction from Version II of the integrated package SmartWare. Findings from the other studies are either contradictory or based on too few experiments for definitive conclusions to be drawn. In addition, instruments and measures used to assess performance have not been standardized.

The research has been by no means exhaustive, and the field is still wide open for contributions to be made on all levels, particularily in the area of evaluation techniques. No methods exist which have found favour and wide adoption within the research community, nor for that matter within the business community.

As a whole, the concept of **ease of use** is, as yet, not clearly defined or understood. This thesis, rather than attempting to rectify particular flaws found in prior research, will endeavour to shed more light on this concept, using a more holistic approach to its investigation than has previously been the case. This is discussed more fully in the following chapter.

This chapter examined **ease of use** of business software within a framework which identified **package class, task, design characteristics, assistance characteristics, user characteristics, instructional strategy, and user role** as important factors contributing to ease of use. The Learning literature and Human Factors literature were used to identify package features and as a basis for suggesting that design and assistance features support various learning factors identified as **speed of performance, support of memory, reduced mental strain, and psychological comfort**. Various methodologies and a number of measures of ease of use were also discussed. Chapter 3 which follows discusses the details of this particular research.

Chapter 3 - PROBLEM STATEMENT AND EXPERIMENTAL FRAMEWORK

This chapter discusses the purpose and background of this research, in the context of the experimental framework used. The variables examined and the research problem addressed are presented with the associated research questions and propositions derived therefrom. The discussion takes place in four parts:

3.A. Problem statement

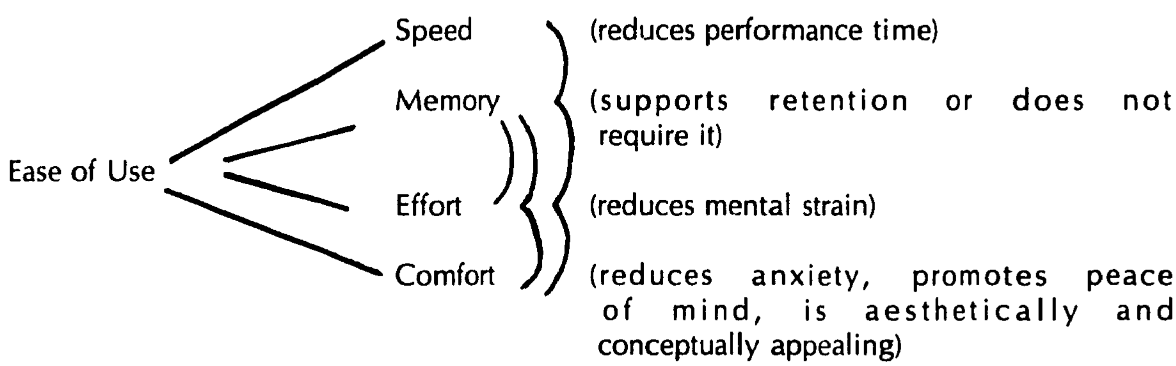
In the previous chapter, we discussed the lack of clear understanding and consensus among researchers and, indeed, users of what constitutes **ease of use and learning** of business software. One school views **ease of use** as an umbrella concept which encompasses **ease of learning** and which refers to expertise at all levels. (Davis, 1989; Roberts and Moran, 1983; Whiteside, et.al., 1985; Carroll and Mack, 1984). Another view of them is as competing concepts. A package designed for ease of learning is expected to facilitate novice performance and is likely to detract from ease of use, which is expected to facilitate expert performance, and vice versa. (Khalifa, 1990). In all instances, as we noted previously, the concept of **ease of learning** does not imply the deeper meaning understood in the educational environment. **Learning** here refers to being able to perform a set of operations required by the package to successfully complete a particular task and not necessarily being able to grasp the deeper, underlying concepts implied by these operations. In this case, the learning being measured may be of a rote nature rather than a measure of true understanding on the part of the user.

In this thesis, we take the former view of **ease of use** as a global concept encompassing **ease of learning**. We view **learning** as the stages of transition required to achieve mastery learning and **ease of use** as relevant to the entire spectrum of usage, regardless of level of expertise. We adopt this view because it accounts for the fact that the learning of a package is, essentially, never complete. The user is usually at a mix of levels of expertise, depending on the mastery attained on the various functions. The user is always in a process of learning new functions or finding new and better ways of performing ones already learned, whereby the user may be novice in some functions, intermediate in others, and advanced in others. To classify **ease of learning** as being applicable to novice users exclusively and **ease of use** as applicable to expert users, would, therefore, not take into account the user's knowledge of the current package as a whole, or other packages. We have suggested that one of the flaws of prior research has been a tendency to consider a package as isolated sets of features rather than as forming an integrated whole. **Ease of use**, therefore, should be concerned with this 'harmony' which a package design creates for all user categories.

In this thesis, we suggest that **ease of use** of a package will be determined by the extent to which it supports the learning dimensions of **speed, memory, effort and comfort** as depicted in Figure 4. Our arguments for suggesting these components were discussed in

Chapter 2.A. and supported by our discussions of the Learning literature and the Human Factors literature throughout Chapter 2. A software package will be considered **easy to use** if it is able to reduce performance time (**speed**), reduce memory load or not require it because of the simplicity of operation or the nature of assistance provided (**memory**), reduce mental effort thereby reducing errors made (**effort**) and is deemed comfortable to use (**comfort**). Evidently, what contributes to reducing the strain imposed on memory and mental effort should also contribute to the psychological comfort felt when using the package, all of which, in turn, should contribute to reduced performance time. The speed with which the task is accomplished is also likely to affect the user's perceived comfört when using the package.

Figure 4 - **Components of Ease of Use: speed, memory, effort, comfort**



Since these dimensions are interdependent and the support of them is likely to contribute to the reduction of performance time, we contend that performance time is the most inclusive measure relating to **ease of use**, including as it does in this thesis time to read help messages and to make and correct errors. We, nonetheless, acknowledge the importance users' perceptions play in the understanding of the concept of **ease of use**, as well as the importance of studying the nature and type of errors made and help solicited. These provide the explanatory insights into the performance time results. Our methods of measuring these are explained in Chapter 4.B.8.

We established a Benchmark Index against which to compare the performance time indices of the users in our study which we propose as a measure of **ease of use**. This Benchmark Index was established to be 3333 for the task used in our study, for the two packages. (See Chapter 4.B.5). The basis for this Index is explained in Chapter 4.B.8. This Index provides us with a baseline of what we might expect from expert user performance and we suggest that **ease of use** can be roughly assessed relative to this Index. The 'easier' a package is to use on first exposure, the closer will be the user's performance time index to this Benchmark Index. Further, if experts come closer to this Index than novices, then we can say the package is not equally 'easy to use' for all classes of users, rather, prior experience moderates package design. We can also say the package is 'easy to learn' to the extent that the improvement in performance on a retrial approaches this Benchmark Index.

We suggest that our view of **ease of use** may provide a potential basis on which to make a multi-dimensional evaluation of packages. For instance, a package design which offers greater support of these dimensions is likely to be easier to use than one offering less support. Secondly, this approach may be valuable in pinpointing more clearly those areas of design in need of being rectified. A package feature rated low in user performance speed or effort may indicate deficiencies in the number of operations required to perform a task while a low rating on memory may indicate deficiencies in the assistance provided for the feature. A low rating in comfort may point to poor screen design, confusing wording, or unclear conceptual models.

This backdrop formed the basis for our pursuit of a better understanding of what defines **ease of use**, in which we were concerned specifically with two main issues:

1) potential consensus among experts on the importance to **ease of use** of various package features and learning dimensions identified;

2) differences in perceived comfort and performance measures relating to **ease of use** resulting from differences in package design, user experience levels, and other user characteristics.

Our approach to this examination is discussed next.

3.B. Scope of research

We have already noted that the concept of **'ease of use'** of a package is variously defined. Any attempts, to measure and to evaluate it will continue to be inadequate until a better understanding of it is obtained. This understanding may perhaps be garnered by adopting an exploratory approach to researching the concept. Past research has tended to centre mainly on hypothesis testing and confirmatory studies. However, as was discussed in the previous section, research of that nature is appropriate when a concept has been fully explored and defined, in such a way as to have received consensus among experts in the area. This is not yet the case with the concept of **ease of use**. Much more needs to be known about what the term means to different classes of users, and in what contexts.

By taking a broader perspective in researching **ease of use**, the richness and complexity which characterizes the human/computer interaction phenomenon can be maintained and examined more authentically. This is lost with the more rigid experimental approaches adopted in most prior research efforts. The generalizability of the results of these micro studies to a more setting is always questionable. One is never sure once the micro parts are put together whether the whole will equal the sum of the parts. (See Chapter 2.E.). To validate the findings from these studies, an holistic approach must eventually be adopted.

Furthermore, the methods which have been developed to date in academic research to evaluate **ease of use** have also taken a very 'micro' perspective. As such, interest has tended to focus on evaluation at the design and beta-testing stages rather than at the marketing stage. (Khalifa, 1990; Polson and Kieras, 1985; Card, Moran and Newell, 1980). Evaluation methods used in academia and in industry, for finished products on the market, are far from optimal, relying mostly on one-dimensional checklists and personal opinions. (Holcomb and Tharp, 1991; Scriven, 1990; Roberts and Moran, 1983). (See Chapter 2.F. for a fuller discussion.). This suggests that there is a significant absence of knowledge, which neither the academic world nor industry has adequately filled. It might be useful, therefore, to step back and approach the issue from a more 'macro' perspective.

This thesis argues that **ease of use** should be a multi-dimensional concept. No single measure will be able to convey the full extent of its meaning. This thesis examines this multi-dimensionality from the perspectives of expert users and designers in the field, and of users with differing experience levels in actual use of a package. Different classes of users, with differing experience levels and differing needs and preferences, may view **ease of use** quite differently one from the other. Additionally, a package's **ease of use** is not likely to be accurately assessed from the consideration of one particular design or assistance feature as has been the case in prior research. (See Chapter 2.D.). It will rather be affected by the way in which all features interact to create a particular effect or environment, to which the user, with his own particular characteristics, then reacts. We are suggesting, that a package should be examined as an entirety, in as natural a setting as possible, in conformity with the usual manner of learning and using a package.

With these ideas in mind, the purpose of this thesis was to conduct an exploratory study of the concept of **ease of use** of business software, using a blend of field study and experimental simulation. The study was conducted in two stages as outlined below:

Study 1 - To compare experts' views of design and assistance features, and how they affect the learning dimensions, with expectations derived from the literature

Our basis for linking design and assistance features with four major learning dimensions was established in Chapter 2 and summarized in Table 2. This table is reproduced following. This aspect of the research sought to determine, first of all, whether experts agree on the importance of certain design and assistance features, and on the associations of these features with the learning dimensions identified. We were also interested in whether experts gave the same weights to the learning dimensions. Secondly, we sought to determine whether our expectations of the links between design features and learning dimensions, presented in Table 2, are supported in practice among experts. Expert designers' and users' perspectives on **ease of use**, defined in this way, were solicited from those in the field. The aim of this aspect of the study was to identify major factors affecting **ease of use** with a view to formulating a basis on which to make a 'qualitative' assessment of the contributions to **ease of use** made by a package's features. The existence of a feature in a package says nothing about the quality of the feature and its implementation in the package. A basis for establishing this 'quality' would greatly further the assessments made of the **ease of use** of packages. (Teng & Jamison, 1990).

Our study goes beyond previous work, such as that by Holcomb & Tharp (1991), Scrivens (1990), Teng & Jamison (1990) and Roberts and Moran (1983) which attempted to make this qualitative assessment using a uni-dimensional approach. (See Chapter 2.F.). We, first of all, distinguish package features based on those features forming an integral part of the software (Design features) and those that can be added on and can either be internal or external to the software (Assistance features). This has not been done previously. We also incorporate the notion of support for the learning dimensions, **speed, memory, effort** and **comfort**, not hitherto suggested. Prior work has often listed **ease of use/learning** and various of the learning dimensions as contributing to performance and perceptions, rather than being outcome variables. (Karat, et.al., 1992; Scrivens, 1990; Teng & Jamison, 1990; Davis, 1989).

Stage 2 - To evaluate selected packages for ease of use in actual use by users

Notwithstanding the expert opinion solicited in Stage 1, we were interested in examining **ease of use** from the perspective of those in actual use of packages. We were interested in what effect certain features identified as being important, by experts in Stage 1, had on **ease of use** in practice. For this purpose, users' performance and perceived comfort reports acquired during actual use of one of two packages were used to assess whether **ease of use** is contingent on the package's design and assistance features, whether moderated by users' experience levels, or due to other factors. We differentiated experience based on experience with packages of similar and dissimilar function to that used in our study, and the extent of this experience. Two statistical packages were used, Merlin$_t$, which has a mixed directed menu/command structure and concise online help index and Minitab$_t$, which has a command structure and full online manual. Merlin$_t$ was also tested with an additional online Hypertext-based index.

This study improved on prior work by adopting an holistic approach to the research which overcomes the limitations of generalizability of experimental research findings. It allows the many and complex factors impacting **ease of use** to be examined the context of one study. The study also compares a truly directed menu structure with a command structure which is a more realistic comparison of the two structures. Undirected menu structures are really only another way of presenting commands so as to eliminate the need to remember them, usually at the early stages of learning a package. We also supplemented the uni-dimensional data usually collected on user's subjective perceptions of the packages (Scrivens, 1990; Teng & Jamison, 1990; Roberts & Moran, 1983) with the support they perceived to have been given by the features of the package to the learning dimensions we identified. We endeavoured also to address the many user performance measures recommended by Barnard & Grudin (1988) which included time to completion tempered by errors made and online assistance sought, analyses of keystroke process traces of errors and help, and looked at learning difficulties by way of retest results, relations to the Benchmark Index and use of help. The only recommended measure not examined is the "number of commands yielding improvement to performance relative to those degrading it". As this requires very fine-grained analysis of the process traces, this is reserved for future research.

The section which follows presents the framework used to guide this research.

Table 2
Links between Software Features and Learning Dimensions Expected from the Literature

Software Features	Learning Dimensions			
	Speed	Memory	Effort	Comfort
a. Design features				
Command Structure - menu/command	X	X	X	
Depth of Structure	X	X	X	
Logic of Structure		X	X	X
Consistency		X	X	
Screen Design	X	X	X	X
Flexible		X	X	
Error Trapping/Recovery				X
System Response Time	X			
Data Input - interactive/batch	X	X	X	
Macros	X	X	X	
Autosave/backup		X		X
Autoadjustment	X		X	
Autoformatting	X	X	X	
b. Help Features/Learning Aids				
Manuals		X	X	X
System Messages		X		X
Tutorials		X		X
Prompts		X		X
Keyboard Templates		X		X
Defaults	X	X	X	X
Examples		X		X
Index		X		X
Glossary		X		X
Unsolicited Help		X		X
Cautions		X		X
Checklists/Memory jogs		X		X
Navigational Aids	X	X		X
Instructive Feedback		X	X	X
Context Help		X	X	X
Expertise Accommodation	X	X	X	
Restricted Options		X	X	X
Subject Matter Aid		X		X
Conceptual Models			X	X

3.C. Framework

A theoretical framework of the variables affecting software **ease of use** was presented in Chapter 2.C. We have suggested that using this framework, indicating where along the various spectra the particular research is positioned, will assist in making comparisons across research studies. Study 2 is described below within this context. As Study 1 was a survey, it is presented in more general terms.

Study 1

In Study 1, we were interested in expert opinion of the contribution that package features make to **ease of use**, regardless of type of package or tasks performed. We solicited mainly system analysts and designers, hence, their psychostructure make-up is likely to be similar.

Study 2

Figure 5 overleaf indicates, in **bold**, the position of our research along the dimensions of **Package Class, Task, Design Features, Assistance Features, User Characteristics, Instructional Strategy, and User Role**. These are discussed in turn, with respect to prior research, in the pages following Figure 5.

Figure 5 - Framework for Research Design

Dimensions	⟵ Spectra ⟶		
Class of pkg.	Canned	⟷	**Decision Analysis**
Pkg. Type	Payroll/ Inventory	**Stats/** OR	WP/DB/SS/CAD
Pkg. Operation			
-Input	**Structured**	⟷	Unstructured
-Process	**Structured**	⟷	Unstructured
-Output	**Structured**	⟷	Unstructured
Task	**Structured**	⟷	Unstructured
Task Elements			
-Input	**Predefined**	⟷	Varied
-Process	**Predefined**	⟷	Varied
-Output	**Predefined**	⟷	Varied
Design Chars.	**High**	⟷	**Low**
-memory			
-speed		**varying degrees**	
-effort			
-comfort			
Assistance	**High**	⟷	**Low**
-memory			
-speed		**varying degrees**	
-effort			
-comfort			
User Chars.			
-exp-pkg.	Novice	⟷	**Expert**
-exp-subj.	Novice	⟷	**Expert**
-psy-anxiety	**High**	⟷	**Low**
-psy-motive	**High**	⟷	**Low**
-psy-cog.stls*	**Methodical**	⟷	**Free-spirited**
-demogs.		**varied**	
Instructional Strategy	Scripted	⟷	**Exploratory**
User Role	Passive	⟷	**Active**

(*The possible cognitive style dimensions are far more complex than this simplistic representation implies. See Chapter 2.D.5.ii.a. for a more rigorous treatment.)

The following is a brief discussion of each of these dimensions, as they pertain to this research:

Package Class

The focus of this work is on the **ease of use** of packages which have been designed to support decision analyses. Two statistical packages, which may be considered to fall under the semi-structured banner of this category of packages, were used in the research. This is in keeping with past research which limited its investigations to packages of the same class and type. (Green and Gilhooly, 1990; Napier, et. al., 1989; Mack, Lewis and Carroll, 1987; Holt, et.al., 1986; Carroll, 1985; Whiteside, et. al., 1985; Cohill and Williges, 1985, 1982). Statistical packages were chosen because, unlike wordprocessors and spreadsheets, which have been the focus of many past research efforts, they contain a distinct separation of data input, data analysis, and data output. It, therefore, simplified the execution and collection of task performance data.

Task

To overcome the requirement for subject matter expertise, our experiment was confined to a structured task, for input, process and output. The task identified for in this experiment consisted of: data entry, data editing and listing of this data on the screen, saving the data and exiting the package. The experimental task was defined in this manner, as independent of the statistical packages being used, because the functions are common to all packages, regardless of class or type, and because no subject matter expertise was required. In addition, it was easily administered and is the first step required to perform statistical analyses with a package, and indeed, inputting is the required first step in any package.

Design and Assistance Features

The two statistical packages chosen differed on important design and assistance features identified in the Stage 1 study by the expert panel as being important for **ease of use**, notably command structure and form, and structure and content of assistance. The Merlin$_t$ package can be characterized as having a mixed command structure, employing both menu and commands. Its online assistance consists of a listing of available commands, with a brief explanation, plus some context dependent explanations. The Hypertext-based help addition to the Merlin$_t$ package consists of an Index from which users can access more and more detailed instructions and examples. Minitab$_t$ is a command driven package with an online duplicate of the full offline manual. Manuals, whether on- or offline were ranked first in importance for **ease of use** by our panel of experts, to be discussed in Chapter 5.A. Past research has investigated different command structures and forms of assistance. Some consensus seems to have been found for the advisability of full on- and offline manuals versus reduced manuals for different classes of users. (Carroll and Aaronson, 1988; Carroll, 1985; Dunsmore, 1980; Relles, 1979). The findings have not been as conclusive for command structures. (Napier, et. al., 1989; Ogden, et. al., in Paap and Roske-Hofstrand, 1988; Lee, et. al., 1986; Perlman, 1984; Shutoh, 1984; Hauptmann and Green, 1983; Dunsmore, 1981). The possible reasons for this were discussed in Chapter 2.H.

User Characteristics

A range of computer expertise was sought among the subjects. A sizeable body of research has investigated the impact of experience levels on package learning. This has mainly been done in regard to the extent of experience that users had with the particular package being investigated. (Foss and DeRidder, 1989; Roth, Bennett and Woods, 1987;

Burns, et. al., 1986; Gugerty and Olson, 1986; Karat, et. al., 1986; Whiteside, et. al., 1985; Hauptmann and Green, 1983; Roberts and Moran, 1983; Gilfoil, 1982; Ledgard, et. al., 1980; Relles, 1979). Our research interest goes beyond this usual focus of expertise, and expands it to include the impact of expertise in one or several packages, of the same or different type, on the package being investigated.

An appropriately arbitrary mix of psychostructural and demographic components can be assumed to have formed the sample of subjects, since a relatively large sample was obtained, 294 in total. The sample, however, was mostly commerce undergraduate students with associated characteristics. No attempt was made to control or assess the impact of user characteristics, though some of the literature indicates that performance with and preference for a particular package design may be influenced by these individual characteristics. (Bostrom, Olfman and Sein, 1990; Evans and Simkin, 1989; Kern and Matta, 1988; Ogozalek and Van Praag, 1986; Gomez, Egan and Bowers, 1986; Howard and Smith, 1986; Gilroy and Desai, 1986; Borgman, 1986, 1984; Ambardar, 1984).

Instructional Strategy and User Role

Subjects were expected to use nothing but online assistance to learn about their assigned package and to perform the task given. It was an entirely self-paced, exploratory adventure. This strategy, which allows the user an active role in learning and doing, is the one most adopted and preferred by users, as was seen in the literature findings previously discussed. (Kern and Matta, 1988; Hiltz and Kerr, 1986; Carroll and Mack, 1984).

The variables which were studied in this research are summarized in the next section.

3.D. Variables studied

Study 1

Figure 6a below categorizes the variables which were examined in Study 1.

Figure 6a - **Variables in Study 1 Research Design**

Controlled Independent Variables	**Dependent Variable**
Design Features Assistance Features Learning Dimensions	Ease of Use

In this study, we were interested in the perceived contribution that various identified design and assistance features and learning dimensions have on **ease of use**, from the perspective of expert users and designers. We were also interested in their perceptions of the associations between the learning dimensions and the package features.

Study 2

Figure 6b below categorizes the variables which were examined in Study 2.

Figure 6b - **Variables in Study 2 Research Design**

	Mediating Variables		
Controlled Independent Variables	**Controlled Constant Variables**	**Uncontrolled Variables**	**Dependent Variable**
Design Features Assistance Features	Package Class Task Instructional Strategy User Role	User Characteristics	Ease of Use

The research compared the **ease of use** of two packages, in the same category, which differed in design and assistance features. Package class, task, instructional strategy and user role were held constant for all users, while the other mediating factors found in user characteristics were assumed to be fairly evenly distributed across the packages chosen for investigation given the large sample size. Only the user characteristics, package and computer experience, gender, competence in quantitative subject matter and computer anxiety were singled out for investigation of their impact on **ease of use**.

Computer and package experience was differentiated, and its impact investigated, according to packages of similar and dissimilar function. **Four** levels of expertise were established, each of which was differentiated further into **novice, intermediate and expert** sub-levels. The basis for this classification is discussed fully in Chapter 4, Section 4.B.4., which follows. The main levels were defined as: Level 1 - those completely inexperienced with any package of any type; Level 2 - those with experience with packages not of the type being investigated; Level 3 - those with experience with only one package of the same type as the one being investigated; Level 4 - those with experience with more than one package of the same type as is being investigated. The resulting grid of the experience levels investigated is shown below in Table 3. Additional insights into the role played by experience and learning was obtained by studying the effect of retention on performance and perceptions. Some subjects volunteered to repeat the same experiment a week later.

Table 3
Level Classification Scheme

Statistics Package Experience	Microcomputer Package Experience		
	Novice	Intermediate	Expert
1 - No Pkgs	L1N	-	-
2 - Varied, No Stats	L2N	L2I	L2E
3 - Varied + 1 Stat	L3N	L3I	L3E
4 - Varied + > 1 Stat	L4N	L4I	L4E

Legend
L1N - minimal, or no, computer or package experience
L2N - novice with various packages, but not with statistics packages
L2I - intermediate with various packages, but not with statistics packages
L2E - expert with various packages, but not with statistics packages
L3N - novice with various packages, including one statistics package
L3I - intermediate with various packages, including one statistics package
L3E - expert with various packages, including one statistics package
L4N - novice with various packages, including more than 1 statistics package
L4I - intermediate with various packages, including more than 1 statistics package
L4E - expert w[illegible] various packages, including more than 1 statistics package

As discussed previously, we used the **Speed** component of **ease of use** to assess **ease of use**, since it reflects the consequences of the other dimensions. Users' performance, with respect to time indices, were developed from computerized traces of users actually using the packages. The other dimensions provided supporting explanation of this outcome. Users' subjective perceptions, 'liking', relating to the **Comfort** component of the packages were collected, as well, further insights were gained from analyzing the traces of errors and help calls made, which were felt to be reflective of the **Memory** and **Effort** components of **ease of use**. The methods for measuring these are discussed in more depth in the following chapter.

The research questions and propositions which emerged for investigation, are presented below.

3.E. Research Questions and Propositions

Four major research questions were considered worthy of investigation in our pursuit of this better understanding and reconstructing of the notion of **ease of use**. These, with associated propositions, are enumerated as follows:

Research Question 1 - **Expert Consensus**

1.1 Is there agreement, among experts in the field, on: (1) the importance of certain Design and Assistance Features in determining **ease of use**, (2) the importance of the Learning Dimensions identified as influencing **ease of use**, (3) the links between features and the learning dimensions they support, and, (4) whether these match expectations which were derived from a review of the literature?

The importance that users assign to design and assistance features and to the learning dimensions supporting ease of use will probably be dependent on their particular needs and preferences. (Todd and Benbasat, 1991; Teng and Jamison, 1990). However, there is an assumption, as yet unsupported in the literature, that experts perform better with, and prefer command structures, while novices perform better with, and prefer menu structures. This was discussed more fully in Chapter 2.F. The assumption in research has been that similar categories of users will produce similar results. As only expert opinion is being sought in this aspect of the research, we will make the assumption that they will exhibit agreement. The propositions may be stated as:

Proposition 1.a. - There will tend to be agreement on the relative rankings assigned to the Design and Assistance Features identified as contributing to **ease of use**.

Proposition 1.b. - There will tend to be agreement on the relative weights assigned to the Learning Dimensions identified as contributing to **ease of use**.

If the theoretical assumptions developed in Chapter 2 and produced in Table 2 have validity, then agreement should be found in the matches made between design and assistance features and the learning dimensions. The propositions can be written as:

Proposition 1.c. - There will tend to be agreement on the links made between Design and Assistance Features and the Learning Dimensions they support.

Proposition 1.d. - The agreements found in Proposition 1.c. will match expectations.

Research Question 2 - **Package Differences**

2.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, as a consequence of differing Design and Assistance Features?

Menu systems have been generally viewed as being easier to use and requiring less memory recall than command systems, especially for novices (Shneiderman, 1987), though, as discussed in Chapter 2, this has not been unequivocally established and, indeed, both forms are being replaced by icons in recent designs. Some studies, in fact, show that mixed systems improve performance over that shown with systems using one form only. (Lee, et. al. 1986). Additionally, the literature seems to indicate that the use of brief manuals to learn a package results in faster learning than a full manual, be it online or offline, for novice classes of users. (Carroll, 1985; Dunsmore, 1980). Based on this, we make a leap for all classes of users and expect that, compared to Minitab$_t$, Merlin$_t$ should be the preferred and easier-to-use package, having a mixed menu and command structure and a concise online help system, comprised of a listing of commands with brief explanations. The Hypertext-based online help index, with examples and expanded explanations, should further enhance ease of use and preference. These features would also place less strain on the user's memory requirements, in subsequent uses of the package. Minitab$_t$, on the other hand, is command driven and has a full online manual, making it more unwieldy and more difficult to recall operations. The propositions associated with this research question can, therefore, be given as follows:

Proposition 2.a. - Merlin$_t$ will produce better performance measures than Minitab$_t$.

Proposition 2.b. - Merlin$_t$ will be given better perceived comfort ratings than Minitab$_t$, (except for users who already have a certain level of experience with Minitab$_t$).

Proposition 2.c. - Users of the Extended Hypertext-based Merlin$_t$ will have better performance measures than either the stand-alone Merlin$_t$ or Minitab$_t$, and will have better perceived comfort ratings than either of them.

Proposition 2.d. - Users of Merlin$_t$ will show a greater improvement in performance, on subsequent use of the package, than will Minitab$_t$ users.

As indicated in actual practice, users are highly adaptable to even the most uncompromising designs if the need to use the system outweighs the imperfections in its design. (Bennett, 1983). As familiarity grows, so does tolerance. It seems reasonable to assume, therefore, that with subsequent uses, a package may be perceived more favourably than on first encounter. This leads to the following proposition:

Proposition 2.e. - There will be a general improvement in the perceived comfort rating for each of the packages, on subsequent use.

In Stage 1 of the research, we postulated that different design and assistance features would support different dimensions of learning, namely speed, memory, effort, and comfort. This was based on our general assessment of the Human Factors and Learning literature rather than on particular findings in the literature with respect to these concepts. We also suggested that the learning dimensions are interdependent, that is, our performance time index or **Speed** incorporates time spent in errors and help which we equate with **Memory** and **Effort**. The extent of errors made and help needed should affect **Comfort**, all of which should affect performance time. We were, therefore, interested in examining the relationship between perceived comfort and the performance indices. To test our intuition, the propositions are given here as:

Proposition 2.f. - Performance measures and perceived comfort ratings will be better when the dimensions, *Speed*, *Memory*, *Effort*, and *Comfort* are perceived to be supported by more attributes in the packages than when less are perceived to be supported.

Proposition 2.g. - Better perceived comfort, or 'liking' ratings will result in better performance indices.

Research Question 3 - **Experience Level Differences**

3.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to differences in experience levels?

Table 3, which summarizes the experience levels, is reproduced here for reference throughout this section.

Table 3
Level Classification Scheme

Statistics Package Experience	Microcomputer Package Experience		
	Novice	Intermediate	Expert
1 - No Pkgs	L1N	-	-
2 - Varied, No Stats	L2N	L2I	L2E
3 - Varied + 1 Stat	L3N	L3I	L3E
4 - Varied + > 1 Stat	L4N	L4I	L4E

Legend
L1N - minimal, or no, computer or package experience
L2N - novice with various packages, but not with statistics packages
L2I - intermediate with various packages, but not with statistics packages
L2E - expert with various packages, but not with statistics packages
L3N - novice with various packages, including one statistics package
L3I - intermediate with various packages, including one statistics package
L3E - expert with various packages, including one statistics package
L4N - novice with various packages, including more than 1 statistics package
L4I - intermediate with various packages, including more than 1 statistics package
L4E - expert with various packages, including more than 1 statistics package

Level 3 users have had exposure to only one package, a statistical package, Minitab. These users are likely, therefore, to be entrenched in this package and to be resistant to learning new ones. This will have an impact on their performance and perceived comfort with the package. The proposition is given as:

Proposition 3.a. - L3 users of various packages, including one statistics package, will perform worse on Merlin, and give it lower perceived comfort ratings than L2 users, who have had experience with various packages, not including statistics packages.

Level 2 users have experience with several different types of packages, though not statistical packages. These users, nonetheless, are likely to be more flexible in switching from one package to another because of their exposure to different package structures and interfaces, than would those completely inexperienced with any packages. The same should be true of Level 3 users who have the additional advantage of prior exposure to a statistics package. Also, assuming from Proposition 2 that a mixed menu structure with concise online help produces superior performance and perceived comfort results, then it can be taken for granted that, as both packages are new to Level 2 users, the package with menu structure and brief online help will produce better performance measures and be preferred by this level of user. The propositions are given as follows:

Proposition 3.b. - L2 users with experience with various packages, not including any statistics packages, and L3 users of various packages, but only one statistics package, will outperform L1 - minimal, or no, experience, but give lower perceived comfort ratings, regardless of package treatment.

Proposition 3.c. - L2 users with various packages, not including statistics packages, will perform better on Merlin, than on Minitab, and give it better perceived comfort ratings.

Levels 3 having had prior exposure to one statistics package which included Minitab$_t$, are likely to be resistant to a new package and therefore give a lower rating to the unfamiliar package. However, these users because of their varied experience should be more adaptable to a new package and so not have their performance adversely affected. The associated proposition can be written as:

<u>Proposition 3.d.</u> - L3 users with one statistics packages, which was Minitab$_t$, as well as other packages, will give lower perceived comfort ratings to Merlin$_t$ than to Minitab$_t$ and perform better.

Level 4 users have had exposure to a variety of statistical and other packages. These users are likely to be adept at navigating among different package interfaces and structures. They should, therefore, have less trouble adapting to new packages than would users from other levels. The proposition can be stated thus:

<u>Proposition 3.e.</u> - L4 users of various packages, including more than one statistics package, of which Minitab$_t$ is one, will outperform all other levels, on Merlin$_t$.

In line with propositions related to Research Question 2, novices can be expected to prefer and to perform better with menu systems and brief manuals. As such, these users should exhibit better perceived comfort and performance measures with Merlin$_t$ over Minitab$_t$, and there should be no differences in the level of perceived comfort between the novices of any of the levels, using the same package. Novices in Levels 2, 3 and 4 should not perform very differently from those in Level 1 because their exposure to packages has been minimal, and not enough to have become habituated to them. The propositions are given as:

Proposition 3.f. - Novices, regardless of level, will have the same level of perceived comfort on the same package.

Proposition 3.g. - L1 - minimal, or no, computer or package experience will rate and perform better with Merlin$_t$ than Minitab$_t$.

It seems reasonable to assume that novices, being inexperienced with packages and their features, are not in the habit of seeking online help and, so may not readily access it. They are also not likely to be sure of what to look for when they do. Users also seem to have a propensity for not reading manuals or referring to help screens. (Hiltz and Kerr, 1986). A proposition can, thus, be formulated:

Proposition 3.h. - Novices will make less help calls than Experts, on Merlin$_t$.

Although the Hauptmann and Green (1983) study suggests that computer experience alone is not sufficient to account for user performance with an unfamiliar package, conventional wisdom suggests that greater experience should facilitate performance. To answer this debate, we test the following proposition:

Proposition 3.i. - Experts and Intermediates will have better performance time and error indices and perceived comfort ratings than Novices, on Merlin$_t$.

Research Question 4 - **Other Factors**

4.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to other factors, such as computer anxiety, gender, and quantitative competence?

Some of the research indicates that users' anxiety traits (Gilroy and Desai, 1986), as well as that generated by the environment in which they operate, can influence performance with and liking for a package. It seems reasonable that novices, grappling with an unknown entity, may experience greater anxiety using computers than experts. We can state the propositions, then, as:

<u>Proposition 4.a.</u> - Those with higher anxiety scores will have worse performance and perceived comfort scores than those with lower anxiety scores, regardless of package treatment.

There has been some discussion that males may tend to be more facile with computers than females, on the assumption that men are more mathematically and machine-oriented and because of the premise that this may be a function of the areas of the brain which are predominantly used. On the other hand, women may be more creative which may be an asset given the nature of interacting with software. (Gazzaniga and Ledoux, 1978). A proposition then can thus be stated:

<u>Proposition 4.b.</u> - Males will exhibit higher performance scores than females.

Conventional wisdom has suggested that mathematical and quantitative abilities should be contributing factors to the learning and use of computers. A study by Evans and Simkin (1989), however, did not support this. To confirm their finding, we propose the following:

<u>Proposition 4.c.</u> - Higher reported quantitative competence will result in better performance and perceived comfort scores.

Chapter 4, which follows, explains the design and methodology used in conducting the research to investigate these questions and associated propositions.

CHAPTER 4 - RESEARCH DESIGN

This chapter discusses the approach that was taken to conduct the research, the development of the instruments used, the packages selected, the samples chosen, the measures and analyses performed, and the experimental situations. These are presented in the context of the two studies undertaken:

4.A. Study 1 - Expert consensus on software design and assistance features leading to ease of use.
4.B. Study 2 - Quasi-experimental evaluation of effect of package and experience level differences, and other factors, on ease of use.

4.A. Study 1 - Expert consensus on software design and assistance features leading to ease of use

4.A.1. General approach

A table was developed which theoretically matched various package design and assistance features with identified learning dimensions or components of **ease of use - speed, memory, effort, comfort** (See Table 2, Chapter 2.D.3.). An Evaluation Form, based on this table, was sent to expert designers and users in the field to:

1) determine whether there was agreement on the importance of particular design and assistance features identified for ease of use;
2) determine whether there was agreement on the importance of the learning dimensions identified as contributing to ease of use;
3) determine whether there was agreement on the learning dimensions supported by each feature;
4) determine if there was support for our expectations derived from a review of the literature.

(See Appendix A for Form and kit provided).

4.A.2. **Instrument development**

Based on a review of the literature, together with personal experience, a list was prepared of potentially important package design and assistance features and learning dimensions influencing **ease of use**. (See Appendices A.3. and A.4.). The instrument was tested on the members of the thesis committee, who can be considered experts in the design and use of software. Following their comments, a few features were added to the instrument, for example, Flexible, Autosave/backup, Autoadjustment, but there were no additions to the learning dimensions. Some basic changes were made to the wording of the instructions so as to ensure greater clarity, and a glossary was attached defining the features and dimensions, thereby outlining accepted meanings of the concepts. (See Appendix A.2.).

4.A.3. **Task description**

A panel of expert users and designers was asked 1) to rank a list of software features thought to influence software **ease of use**; 2) to assign weights of importance to the identified learning dimensions, out of 100%, as they relate to ease of use in general, and not as they relate to any feature in particular; 3) to relate each of the design and assistance features to one or more of the four learning dimensions **(speed, memory, effort, comfort)**. Subjects were allowed to add features and/or dimensions and to incorporate them into their assessment. An example of a completed form is shown in Chapter 5.A., Figure 11, p. 5.A.3.; and 4) we sought to determine whether these findings agreed with our expectations derived from the Human Factors and Learning literature.

4.A.4. **Panel selection**

Expert users and designers, with diverse backgrounds, were solicited from industry and academia. Those candidates from industry who agreed to take part in the research came from the Engineering Departments at Telesat, Ottawa, Canada, CAE Electronics, Montreal, Canada, and Pratt and Whitney, Montreal, Canada. The subjects from academia came from faculty members in the Computer Information System Department at Bryant College, Rhode Island, U.S.A., from graduate students in an Artificial Intelligence course at McGill University, Montreal, Canada, from employees in the Computer Services Department of Concordia University, Montreal, Canada, and from part-time faculty members in the Department of Decision Sciences and Management Information Systems at Concordia University, Montreal, Canada. Participants were sought who made extensive use of general-purpose packages in the performance of their work activities, either directly, or by using the packages to design other systems. It should be noted, that although their work environment imposed different system requirements, their educational formation is quite similar, most having Bachelors in Computer Science or MIS. This may indicate a predilection towards consensus among them.

4.A.5. **Survey details**

The Evaluation Forms were distributed and collected by a contact person at each of the organizations involved. A covering letter was attached to the Form asking that it be completed independently. A Proficiency Questionnaire, described in the section following, was also attached, in order to verify that the experience levels of the participants were in line with those being sought for this aspect of the research. Subjects filled in the forms at their convenience. (See Appendix A for survey kit).

4.A.6. **Measures**

Four measures were of interest in this portion of the research. These are:

i) the degree of consensus on the ranking of design and assistance features;

ii) the degree of consensus on the importance of the learning dimensions for ease of use;

iii) the degree of consensus on the matching of learning dimensions with design and assistance features;

iv) the degree of convergence between the findings in the field and our expectations.

4.A.7. **Analyses**

The average frequency of the expert panel's responses were analyzed for consensus of our experts and conformance with the expectations established in Table 2, Chapter 2.D.3. Multivariate analysis was performed on the weights of importance assigned to the learning dimensions and the Friedman H-statistic was used to assess the overall levels of consensus. (Neter, Wasserman and Kutner, 1985; Meddis, 1984).

4.B. Study 2 - Quasi-experimental evaluation of effect of package and experience level differences. and other factors, on ease of use

4.B.1. **General approach**

This aspect of the research addressed the issue of the impact on **ease of use** of different package design and assistance features and the moderating effects of user experience levels and other factors. The performance and perceived comfort of users, with different experience levels, psychostructure makeups, and demographic profiles, were examined in actual use with one of three package treatments: (1) package 1, Merlin$_t$ - based on a mixed menu/command structure, (2) package 2, Minitab$_t$ - based on a command structure, and (3) package 1, Merlin$_t$ with a hypertext-based online help index. These packages are representative of the types of package design which users often face, in the usual performance of statistical decision analyses. Recent improvements to design are just beginning to emerge in the mainstream. A large percentage of statistical work is still being done on mainframe computers with awkward and outdated interfaces.

Subjects in the experiment worked independently on a task, with only online assistance as an aid. Some were asked to repeat the task a week later so as to assess **ease of use** relative to retention. In this research, we were interested in the initial reactions and performance levels of different user categories using the packages. No training was provided, nor was there any attempt to lead users to levels of mastery. Subjects were expected to use online assistance exclusively to learn about the packages and to perform the task. It was an entirely self-paced, exploratory adventure. Users were given instructions on how to sign on to the network and to access the package to which they were assigned. That was the only instruction given, as a minimal amount of experimenter interference was desired. Users were asked to proceed through the task as they would in the usual undertaking of such an assignment, with the exception that no assistance would be provided nor could it be sought outside the online assistance available with the software. Assistance was given only when subjects were totally unable to progress, and this took the form of directing the user to seek the available help. In the case of a subject being in a loop, the experimenter moved the subject to a point in the package where commands could again be issued or help activated.

No control could be exerted on consultations engaged in outside of the experimental setting with respect to those who repeated the task a week later. This was not a problem in the case of Treatment 1 - Package 1, or Treatment 3 - Package 1 with hypertext online help, because this package was not currently being used in any courses. To minimize the possibility of contamination in the Treatment 2 - Package 2 group, the subjects were told they would probably be assigned to a different package the next time. Also, the fact that the packages were not readily accessible outside of the experimental situation should have further minimized this potential problem.

A secondary aspect of the research was to investigate whether experience level had a moderating effect on the package treatment results. We were particularly interested in the effect of prior experience with packages of like-type, in this case, statistical packages. Information on experience levels was obtained from a Proficiency Questionnaire administered to the subjects assigned to the various package treatments. (See Appendix A). This is explained in Sections 4.B.3. and 4.B.5. following. This information could not be obtained beforehand, so subjects could not be randomly assigned, nor assigned in equal numbers to each of the package treatments. Subjects also had to be accommodated according to the time slots which suited their schedules. The experiments took place at the beginning of the 1991 winter and summer terms, before the students had started any computer work for their courses, and were held on two days, in two two-hour slots, with experiments running simultaneously in two computer labs.

4.B.2. **Package selection**

Two statistical packages, Merlin$_t$ and Minitab$_t$, were used in the experiments. In the design of statistical packages, there is a distinct separation of input, processing and output operations, which is not the case in most other packages, such as wordprocessors or spreadsheets. For this reason, statistical packages were chosen for the experiment. This separation of operations afforded greater control over the task being performed, which is discussed later, and ensured that the basis of comparison of the two packages was focused on their differing interaction styles. These particular statistical packages were chosen because they differ on important design and assistance features, yet were quite similar in the operations required for the task. (See Figure 10, p. 153). These packages had the added convenience of being available on the Concordia University Computing Services PC-LAN Instructional Network. This permitted us to collect data on multiple users simultaneously. In addition, each package has the capability of tracking keystroke-level performance. For consistency, the Norton$_t$ TimeMark software was used to monitor the total elapsed session time measures on both packages, although Merlin$_t$ has its own facility for tracing which, additionally, includes time stamps between each keystroke. (See Appendix H for an example of Merlin$_t$'s keystroke tracing).

Treatment 1 - Package 1, Merlin$_t$

Merlin$_t$ has a mixed menu/command structure. Inputting and editing requires structured line-entry, but the operations are invoked, in some cases by menu, and in others by command. Initially, the system presents the user with menus; however users have the option of inputting a command string instead. Editing is performed in the editing sub-system which is command driven. Analyses are performed in a structured menu environment.

Output can be achieved either by menu or commands. Merlin$_t$ does have the facility, in Express-mode, for entirely command-driven operation, similar to the Lotus$_t$ style command sequence, but users were not informed of its existence specifically. This was in order to keep the command structures between the packages as different as possible. The existence of this operation is referred to in the online documentation provided, but users would require the offline documentation to know how to invoke it accurately.

Conventional wisdom suggests that directed menu systems should be faster to learn, if not to execute, and easier to learn, since only single keystrokes are required. They are not necessarily faster in execution, however, this depending on the number of levels the user must navigate to perform an operation. Merlin$_t$'s menus are quite shallow so execution should be fairly fast and selections easy to remember, thereby reducing the mental strain imposed on the user. So far, however, the research on these issues has not been conclusive, as discussed in Chapter 2.H. Lee, et. al. (1986) found in their experiment that a mixed system was preferred and resulted in better performance by their subjects. Whiteside, et. al. (1985), on the other hand, found that novices had difficulty with the menu-based database management systems in their experiment, and that all categories of users were better off with the command systems. Paap and Roske-Hofstrand (1988) found that their experienced users tended to select the menu option, while the least experienced used the commands.

Merlin$_t$'s online documentation is designed around commands and their definitions, which are succinctly explained. Help is invoked by simply typing "?". Mental effort should, therefore, be reduced, and the easy access to help should provide some psychological comfort to the user. An overview of the package and its operation is available at sign on and is accessible by command thereafter. More detailed explanations can be found in an offline manual, but this was not provided to users. Examples of the Help features are given in Appendix D.

Treatment 2 - Package 2, Minitab$_t$

Minitab$_t$, has a command-driven structure. This package permits inputting and editing either by line-entry or full-screen editor. Users were restricted to line-entry because performance in the editor cannot be recorded. Analyses and output are accomplished by entering commands. Command structures are likely to impose greater demands on memory and mental effort, thereby impacting performance speed.

The Minitab$_t$ online assistance is a dense manual-based document accessible by the command "**HELP HELP**". More specific help can be obtained by typing **HELP** followed by the menu number associated with the type of help desired, for example, **HELP COMMANDS 1**, to get help on inputting and editing. Users were not provided with any offline documentation. For those unfamiliar with the package, the mental effort required to navigate a full online manual should have a negative effect on their psychological comfort. Also, the difficulty and time required in locating pertinent information suggests that performance speed should be adversely affected. Work done by Carroll (1985) has shown that reduced manuals are more effective for novices than full blown manuals. (See Appendix C for examples.).

Treatment 3 - Package 1 plus hypertext online help

An adjunct experiment had some users perform the task using Package 1, Merlin, with additional assistance available from a hypertext-based online help index. The index was designed to provide assistance and serve as a reference. It includes examples and explanations of concepts. This should further contribute to the psychological comfort and reduced mental effort required of the user. (See Appendix D for examples of these screens.).

Appendix B compares the existence and absence of the various design and assistance features outlined in Chapter 2.D. in each of the packages. Appendices C and D show examples of the requirements needed to perform the task in each of the packages used in the experiment.

4.B.3. **Instrument development**

Users' subjective assessments required the development of an instrument to measure their perceptions of the packages. The features and dimensions included on the instrument (see Appendix F), are the same as those found on the Evaluation Form, which experts were asked to assess in Section 4.A. This Perception Questionnaire, which was developed by this author, asks for an overall rating of their perceived comfort with the package, 'liking', and then requires a direct dichotomous yes/no response concerning the perceived contribution of the various design and assistance features to **speed, memory, effort, comfort.** Notwithstanding the limitations of a Yes/No response, given the required length of the experimental task, which was one hour, and the length of the questionnaire, it was deemed important not to overload the subject further by requiring more fine-grained assessment. Additionally, we were interested in whether perceived support of these learning dimensions had an impact on performance and perceived comfort. A few features were excluded, being judged to be irrelevant for the particular packages, in deference to the length of the questionnaire. The instrument was tested on two individuals, one who could be considered intermediate in experience, and the other, a novice. No changes to the instrument were indicated from their comments. Information on demographics and anxiety levels were procured from a Proficiency Questionnaire, which is described below and can be found in Appendix A.

The Proficiency Questionnaire, developed by this author, was administered to subjects before beginning the task described below, to elicit information on certain user characteristics. (See Appendix A). In particular, the instrument provided information on experience and demographics, including age, gender, scholastic achievement, work experience, extent of computer and package experience, where and how obtained. The classification of frequency of use was borrowed from Igbaria (1990). A computer anxiety measure, developed by Raub (1981), was included which was also taken from Igbaria (1990). Reliability coefficients for this measure as high as .85 have been reported. Subjects' experience levels were determined from their responses to subjective assessments of expertise and frequency of use of the various packages they listed. The determination of experience level is explained more fully in the next section. The instrument was tested on subjects in a small pilot study. No changes were made to the content of the instrument, only to its aesthetic layout.

4.B.4. **Experience level rating**

4.B.4.a. Establishment of Cumulative Microcomputer Package Experience Levels

As experience levels and frequency of use are not necessarily the same for each of the packages reported by subjects, a composite evaluation of cumulative microcomputer package experience had to be made. Classification of subjects into package experience levels was determined from the Proficiency Questionnaire. In the absence of the opportunity to test user proficiency in each package, and to account for inaccuracy in user self-assessments, it was decided that these assessments of expertise would be modified by taking into account the frequency of use reported. A grid was established influenced by Lee 1986, and an experience level was assigned to each package reported. A composite rating to a cumulative level of package experience then had to be made subjectively by the experimenter. This was cross-checked with the ratings made by an independent expert designer and user. There was 89.6% agreement in the ratings made. A third expert was asked to resolve any discrepancies. As statistical packages were being used in the experiments, experience with them was given more weight than the others. Less importance was given to experience reported for wordprocessors than for spreadsheets, databases, statistics, and programming languages, while spreadsheets, databases, and statistical packages were weighted more heavily than programming languages. The grid used to determine experience on the various packages and compositely, is shown following in Figure 7:

Figure 7 - <u>Rating Scheme for Microcomputer Package Experience Levels</u>

Frequency of Microcomputer Package Use Reported	Microcomputer Package Experience Levels Reported			EXPERIENCE LEVELS ASSIGNED
	Expert	**Intermediate**	**Novice**	
Never	Intermediate	Novice	Novice	
Less than once/month	Intermediate	Novice	Novice	
Once/month	Expert	Intermediate	Novice	
Few times /month	Expert	Intermediate	Novice	
Few times /week	Expert	Intermediate	Novice	
Once/day	Expert	Expert	Intermediate	
Several times/day	Expert	Expert	Intermediate	

As an example, one subject's reported experience with various packages is given below. The rating of this person's cumulative microcomputer package expertise was established as follows:

Package Type	**Experience Level Reported**	**Frequency of Use Reported**	**Experience Level Assigned**
Lotus	Expert	once/month	Expert
Supercalc	Intermediate	< once/month	Novice
Multiplan	Intermediate	< once/month	Novice
Wordperfect	Intermediate	once/month	Intermediate
Wordstar	Intermediate	once/month	Intermediate
Basic	Intermediate	< once/month	Novice
Fortran	Intermediate	< once/month	Novice
Pascal	Intermediate	< once/month	Novice
Minitab	Intermediate	once/month	Intermediate

Composite rating of cumulative microcomputer package experience: Intermediate

4.B.4.b. **Establishment of Experiment Experience Levels**

Additionally, in our study, we were interested in the effect of different experience levels with packages of similar and dissimilar function, that is, statistical packages versus other packages, on performance and perceived comfort. It was, therefore, necessary to classify users according to the various Microcomputer Package Type Experience they reported:

- minimal, or no, experience with any packages;
- experience with various packages other than statistics packages;
- experience with various packages, including one statistics package;
- experience with various packages, including more than one statistics package.

In relation to these, cumulative microcomputer package experience levels were assigned to users as described before in Section 4.B.4.a. This lead to the placement of users into the Experimental Levels used for comparison in the study, based on the scheme shown below in Table 3.

Table 3
Level Classification Scheme

Statistics Package Experience	Microcomputer Package Experience		
	Novice	Intermediate	Expert
1 - No Pkgs	L1N	-	-
2 - Varied, No Stats	L2N	L2I	L2E
3 - Varied + 1 Stat	L3N	L3I	L3E
4 - Varied + > 1 Stat	L4N	L4I	L4E

Legend
L1N - minimal, or no, computer or package experience
L2N - novice with various packages, but not with statistics packages
L2I - intermediate with various packages, but not with statistics packages
L2E - expert with various packages, but not with statistics packages
L3N - novice with various packages, including one statistics package
L3I - intermediate with various packages, including one statistics package
L3E - expert with various packages, including one statistics package
L4N - novice with various packages, including more than one statistics package
L4I - intermediate with various packages, including more than one statistics package
L4E - expert with various packages, including more than one statistics package

In the example given in Section 4.B.4.a., this subject, being rated as having intermediate cumulative package experience and having experience with only one statistical package, would be assigned to Experimental Level - L3I.

4.B.5. <u>Task selection</u>

An informal pilot study had been conducted with MBA student subjects to ascertain what level of task could be expected to be accomplished within a one-hour time frame, using each of the packages, and given the varying experience levels of subjects. They were required to perform data entry and editing, and a regression analysis. Based on the findings from the pilot, the task was scaled down to a data entry and editing task, which is required by all packages, regardless of class or type, and does not require subject matter expertise. This ensured that the basis for comparison of the two packages lay mainly in the differences between their interaction styles. This is further assisted by the distinct separation of input, processing and output operations characteristic of line-editor-type statistical packages.

The task was sufficiently simple and structured so that users of all levels of expertise were able to perform it, with different levels of success. The task required six distinct sub-tasks - creating a file of data and labels, viewing or listing the file on the screen, editing an observation, saving the file, and exiting the program. See Appendix G for the task, and Figure 19, to be discussed later, for the exact steps required by each package to perform it. The task was to enter a dataset of 20 students' marks on two tests. Users were asked to verify their input and to make any necessary corrections. A forced editing task was required, in which users were directed to change Student 15's Test1 score to 55. They were then instructed to save and exit the package. The only assistance provided was online help, for which written instructions on how to access in were given to them. A week later, on a voluntary basis, some subjects repeated the same task, on the same package, in order to determine **ease of use** in terms of retention, and to see whether their perceived comfort ratings and performance scores improved with use. The interval of a week was imposed by the fact that, after this time, the subjects would have become involved in class assignments using the computer and, in some cases, the Minitab$_t$ package, thereby possibly changing their initial experience level classifications.

4.B.6. **Sample selection**

Those who took part in Study 2, to assess users' performance and perceptions in actual use of a package, were taken from the faculty and student body of Concordia University's Decision Sciences and Management Information Systems. Faculty and research assistants were already familiar with Minitab$_t$ and other statistical packages, as well as with other types of packages, but not with Merlin$_t$. The students who participated were undergraduate students from two introductory Statistics courses and two introductory Management Information Systems courses, having agreed to take part in the experiment in return for remission of one assignment in those courses. These students either had no prior experience with statistical packages or had experience only with Minitab$_t$, plus varied experience with other types of packages.

Some attempt to increase motivation was introduced, as mentioned, since students participating in the experiment were exempt from one assignment in the course from which they were solicited. Also, they were told the results of the experiment could be beneficial for future students, if not for them, since the results may help the university in choosing packages for its courses. An attempt was made to minimize anxiety by emphasizing that their level of performance on the experiment in no way formed part of their evaluation in the course. On the other hand, this could have had the reverse effect of promoting laziness or carelessness. Given the size of the sample, 234 students, other user characteristics were assumed to be arbitrarily distributed across the sample.

4.B.7. <u>Conduct of Experiment</u>

The experiments with student subjects were conducted in the student computer labs at Concordia University. Subjects were assigned arbitrarily to two labs, one containing 20 IBM PS2's and the other, 35 Olivetti machines, connected to Concordia University Computing Services PC-LAN Instructional Network. Some subjects worked on Treatment 1 - Package 1, Merlin$_t$ in one room, while others worked on Treatment 2 - Package 2, Minitab$_t$ in the other room. Subjects were not assigned in equal numbers to package treatments but, for simplicity, were assigned by class. We attempted, however, to have an equal number of subjects on both machines, in each of the Minitab$_t$ and Merlin$_t$ treatments, because of the difference in speed between the PS2 and Olivetti models. Subjects in Treatment 3 - Package 1, Merlin$_t$, with the hypertext-based online help index were each required to use two computers, one housing the original Merlin$_t$ software, and the other, the extended help system, which reduced the number of subjects who could be processed at one time. Olivetti's were used for both systems. At the time the experiments were conducted, the software in which the ancilliary online help system was written did not permit integration with the original Merlin$_t$. Some of these sessions were videotaped using two cameras to ascertain that users did, in fact, make use of the additional help, since traces could not be obtained for it. The experiments with faculty and research assistants took place in the Faculty

Information Services Computer Room in the Department of Decision Sciences and Management Information Systems. These subjects were assigned only to the Merlin$_t$ package as they were already familiar with Minitab$_t$. Our interest in these subjects focused mainly on that aspect of the research which studied the impact on performance and perceived comfort of different experience levels, with the same type of package. These subjects used an Olivetti machine.

4.B.8. <u>Dependent variables</u>

Ease of use of each of the packages was assessed and compared on the basis of a performance time index, and supported by error and help indices - Score/Time Index, Error/Score, Help/Score. The overall time, the raw number of errors and the raw number of help calls were obtained directly from the keystroke-level traces of each subject, which were recorded on individual diskettes. The errors and help calls were summarized by type, operational or conceptual, according to the sub-tasks performed. Perceived comfort or 'Liking' was measured by a single 5-point likert scale question.

4.B.8.a. Measures

Performance indices were based in part on scores reflecting the success in completing the assigned tasks. A straight comparison was not possible since some individuals completed the task before the end of the hour allotted to the experiment. It was, therefore, necessary to derive indices which could be compared on a common basis. This scoring is discussed in more detail in Section 4.B.8.b., following.

The performance indices were computed as follows:

Score/Time Index = sum of scores on sub-tasks*100/elapsed time of session [1]
Error/Score Index = number of raw errors*100/sum of scores on sub-tasks
Help/Score Index = number of raw help calls*100/sum of scores on sub-tasks

[1] Time recorded on subjects' traces made by Norton$_t$ time stamp

Using this method of measurement, a possible benchmark for expert performance was established as a rough guideline for possible user performance. This benchmark was established to be 3333, for both packages. This represented error-free completion of 100% of the experimental task in three minutes, by the designer of the Merlin$_t$ package on Merlin$_t$ and this author on Minitab$_t$.

The time measures were ultimately compared, for the three package treatments, to assess their **ease of use**, in terms of performance, relative to the average rather than relative to the Benchmark Index. This was because the Benchmark Index being the same for both packages, the base is the same for both packages. The Index was, also, representative of only one expert's performance on each package and is likely to be moveable across many experts and many trials.

A higher score on the Score/Time Index indicates better performance. As this measure includes time spent making and correcting errors and accessing help, we also compared the other performance measures to further explain the results. A higher score on the Error/Score Index indicates a less favourable index. A higher score on the Help/Score Index indicates more use of help which is not necessarily indicative of the quality of performance. Retention was also evaluated based on the improvement to the performance and perceived comfort results obtained on a retest compared to the initial test.

As noted, in addition to the performance measures, users' overall perceptions of the packages, or perceived comfort with the packages, were also measured. These were obtained from the Perception Questionnaire previously discussed. The responses were compared for the three package treatments. The questionnaire also asked users whether they perceived **speed, memory, effort, and comfort** to be supported by the various design and assistance features in the packages. These were used to determine whether perception of support for these learning dimensions helped perceived comfort and performance.

The effect on these performance and perceived comfort measures relating to **ease of use** were also investigated for experience, gender, computer anxiety and perceived quantitative competence.

4.B.8.b. Relative rating of difficulty for sub-tasks

Scores were assigned to each sub-task based on the author's perceived conceptual difficulty of the sub-task, much as an exam marking scheme would be established, and on the assumption of relative independence between the sub-tasks. This decision was taken so as to facilitate comparison across the packages, which it was felt could only be achieved if scoring for the sub-tasks was determined independently of the package being used. We chose this approach because we felt that assigning separate weights to each package based on its particular difficulty in executing the individual sub-tasks would tend to minimize the variability between the packages. Without a common base of reference, comparisons would be meaningless. Each sub-task was conceptually assessed for its level of difficulty relative to the others, such that the total for all sub-tasks summed to 100%. The score assigned was arbitrary, and for simplicity of computation, was assigned in multiples of 5's. The same scoring scheme was applied to both packages, thereby preserving comparability. Scoring was developed as follows in Figure 8:

Figure 8 - Establishment of Sub-Task Scoring for relative difficulty

Score	Task	Conceptual difficulty
35	Sub-task-1-Enter data	- conceptualize separation of data and labels - conceptualize row-wise versus column-wise entry of data - find operation to achieve data input - find operation to end input
15	Sub-task-2-Name labels	- conceptualize separation of data and labels - find operation to insert labels
25	Sub-task-3-Change element	- conceptualize changing portion of file - conceptualize changing row or element only - find operation to accomplish change
10	Sub-task-4-View data	- conceptualize difference between on-screen viewing and printing on printer - find operation to view file on screen
10	Sub-task-5-Save data	- find operation to save data in a file - understand assigning of a filename to the data input
5	Sub-task-6-Exit package	- find operation to terminate session
100%		

In each of the packages, the operations required to accomplish these sub-tasks were compared in order to ascertain if significant advantage would be given to either one using this scoring scheme. The packages were deemed to be "relatively" comparable in complexity, taken at face value. The comparison is shown below in Figure 9:

Figure 9 - Comparison of Tasks on Merlin$_t$ and Minitab$_t$

Task	Merlin$_t$ - mixed menu/command	Minitab$_t$ - command
Entry	1. make selection from DATA menu[1] - enter data row-wise or enter DATA command - make selection from menu - enter data row-wise 2. enter <carriage return> to end data entry	1. enter command READ columns - enter data row-wise or enter command SET column - enter data column-wise 2. enter END command to end data entry (Full-screen editing was not permitted)
Naming	1. select Y from menu following[1] data entry - enter labels one per line or enter NAME command - enter labels one per line	1. enter NAME C1 = 'label1' c2 = 'label2'
Change	1. enter EDIT command - enter r el(row,column) command - enter new element or enter r row15 command - re-enter row	1. enter LET column(row) = new element (Full-screen editing was not permitted)
Viewing	1. select Y from menu following naming[1] - select from menu - all,row,column or enter PRIN command - select from menu - all,row,column or enter EDIT command - enter print commands, eg. p *-all, p el(15,2)-one element	1. enter PRINT column numbers
Saving	1. enter SAVE command - enter filename	1. enter SAVE 'filename'
Exiting	1. enter STOP, DONE, BYE, QUIT, or OFF - enter filename to save, or carriage return or enter S and exit without SAVE option	1. enter STOP

[1] At initial sign-on, Merlin$_t$ leads the user through the steps required to enter data, add column labels and list the data on the screen. Subsequent sign-on's require the user to select these options from the main menu.

The number of errors on each sub-task, using the particular package, should be an indication of the difficulty in accomplishing the task with that package. This was used to verify the weights which we arrived at conceptually, and to ensure that they did not unduly favour one package over another. This is discussed in the analysis chapter, Section 5.B.9.

We cannot compare our objective **ease of use** results with those of the other authors cited in the thesis since, in each case, our measures are defined differently. Roberts and Moran's (1983) study included time spent in error states in their performance time measures. They also provided training to their subjects and permitted the opportunity for error correction. Murphy (1992) attempted to measure the user's transition to mastery levels of learning, and was interested in the time it took to reach said levels with competing designs. Khalifa's (1990) subjects were given training on the mock interfaces used in his experiment, and his performance time measure was calculated by adding the time it took to read the manual to the time spent in training, and then subtracting the time spent during actual performance. Models such as Card, Moran and Newell's (1980) Keystroke-Level Model and Polson and Kieras's (1985) Production Model, we have already noted, are not adaptable to a business context, in terms of a user faced with evaluating and making a choice from among competing packages. They require very fine-grained analysis of the times on each sub-task, which, in the case of the Keystroke Model, also includes the time to read the instructions before beginning the task. As Minitab, does not permit the tracking of keystroke-level time intervals, the detailed analyses of these authors cannot be replicated with our data. Our subjects were not given training, nor was any provision made to allow them to attain mastery levels. We simply investigated whether improvement or deterioration resulted after a week of non-use of a package, and whether this was dependent on the package being familiar or un-familiar.

4.B.9. Analyses

The Tukey-Kramer Comparison of all pairs was used for the univariate tests of mean results on performance and perceived comfort relating to **ease of use,** by package treatment, experience level, reported competency in quantitative courses, gender, and anxiety. The Student's t was used to assess the level of significance of the results for the particular propositions which were tested. Multiple regression and logistic regression analyses were then used to analyze the predictive and explanatory power of these variables for the performance and perceived comfort measures relating to **ease of use**. (Neter, Wasserman, and Kutner,1985). Frequency distributions were used to summarize the demographic data collected on the Proficiency Questionnaires. The MacIntosh SAS JMP, statistics package was used for all the analyses.

This chapter presented the details of the research design and experiment setting used in the thesis. Chapter 5 which follows will present the analyses, findings and implications resulting from this research.

Chapter 5 - RESULTS, ANALYSES AND FINDINGS

This chapter presents the results, analyses and findings of our two studies. For each study, a detailed description of the data and the results of the statistical analyses are presented separately and discussed in the context of the research questions which were posed. The discussion is, therefore, divided as follows:

5.A. Study 1 - Expert consensus on software design and assistance features leading to ease of use
5.B. Study 2 - Experimental evaluation of effect of package and experience level differences, and other factors, on ease of use

5.A. Study 1 - Expert consensus on software design and assistance features leading to ease of use

In Study 1, we were interested in establishing whether a consensus exists among experts in the field concerning that which influences the **ease of use** of a package. We suggested that features included in a package contribute to its **ease of use** to the extent that they support speed, memory, effort and comfort. As suggested by Holcomb and Tharp (1991, p.50),

> "Usability is not *all or nothing* but relative; thus it should be possible to measure adherence to a set of general usability principles and compute a *relative* usability rating for a given interface".

We were interested in investigating whether, indeed, it is possible to identify the general usability principles for which a relative usability rating may be derived. As discussed in the

Introduction, Smith and Mosier (1984) established a list of 679 guidelines for designing the user interface but these have not been widely adopted: 1) because of the difficulty in implementing them and 2) because there may not be agreement on these guidelines. The authors also did not suggest a prioritization of these guidelines nor a basis on which this could be established.

The research question we posed was:

Research Question 1:

Is there agreement, among experts in the field, on: (1) the importance of certain Design and Assistance Features in determining **ease of use**; (2) the importance of the Learning Dimensions identified as influencing **ease of use**; and (3) the links between Features and the Learning Dimensions they support. Finally, (4) to the extent that there is agreement among the experts, does this match expectations derived from the Human Factors and Learning literature?

A panel of experts was asked separately to rank a list of Design Features, and a list of Assistance Features, according to their importance for **ease of use** of a package. They were then asked to indicate the associations they perceive exist between the Features listed and the Learning Dimensions identified as being **Speed, Memory, Effort and Comfort**. They were also asked to assign a weight of importance, for **ease of use**, to these Learning Dimensions. The weights assigned had to add up to 100%. A completed sample form and description of the items are attached for reference throughout this chapter in Figure 10.

It should be noted that the Design and Assistance Features are not necessarily independent. Rather, it is the interplay of all of the features in a package which contributes to its overall ease of use. Neither are the Learning Dimensions independent. The requirements placed on Memory and on Mental Effort by the package, will have an impact on performance Speed and on the psychological Comfort in using the package. Likewise, Memory and Mental Effort are dependent on each other. For this reason, we allowed tied rankings that is, more than one feature could be assigned the same rank. They could also assign more than one learning dimension to each of the features.

The basic statistical analyses carried our were:

²Rank (ordinal) - function of ¹Features (nominal)

¹Features (nominal) - associated with ³Dimensions (nominal)

Comparison of Means of ⁴Weights of Dimensions (interval)

Note: Superscripts refer to the associated variables on the filled-in portion of the form overleaf.

In this section, we present the profile of the panel of experts surveyed in this study and the analyses of Study 1. The analyses are presented and discussed in relation to the propositions which were developed in Chapter 3.D.

Definitions and Descriptions of Features and Learning Factors

Learning Factors:

Speed - time required to perform an operation
Memory - support offered to recall, retention over time
Effort - cognitive complexity, mental strain required to perform operation
Comfort - impact on anxiety, peace of mind, aesthetic appeal

Design Features:

Interface Command Structure - menus, commands, direct manipulation, natural language
Depth of Structure - levels required to perform an operation
Logic of Structure - makes sense, is understandable
Consistency - not ambiguous or contradictory in conventions, functions, messaging
Screen Design - cosmetics/aesthetics - highlighting, animation, layout, etc.
- grouping meaningful - placement on screen logical with respect to purpose
- language/wording - clear, understandable, inoffensive

Flexible - more than one way to perform an operation
Error Trapping/Recovery - warning and/or correction of errors outside parameter ranges, or of "o","l" (letters) to "0","1" (numericals) in data input
System Response Time - elapse time between user input and response from system
Data Input - full-screen editor or line editor
Macros - automation of frequently used string of operations
Autosave/Backup - inputs intermittently saved by system
Auto-adjustment - changes in input reflected automatically in analyses and graphs
Auto-formatting - automatic formatting of input into decimal places, scientific notation, currency, etc.

Assistance Features:

Manuals - on and offline manuals explaining package features and how to use them
System Messages - messages generated by system due to unanticipated entries
Tutorials - on and offline training modules on how to use the package
Prompts - warning of errors made, how to correct and why occurred
Keyboard Templates - on and offline memory jogs of function keys associated with operations
Defaults - expected or anticipated responses supplies which are modifiable
Examples - examples of correct or valid operations and actions, could also include non-examples
Index of commands and operations - quick reference to commands and operations
Glossary - definitions of terms and commands
Unsolicited Help - help supplied at system initiation after prolonged delay in input
Cautions - warnings against overwriting files, deleting file, etc.
Checklist & Memory Jogs - traces of most recent or most used operations
Navigational Aids - trees, maps, control keys
Instructive Feedback - input assessed for correctness and communicated to user with possible action
Context Help - help related to place error made
Expertise Accommodation - accommodation of novice to expert transition
Restricted Options - restricted to certain operations at certain junctures
Subject Matter Aid - assistance with subject matter concepts, e.g. statistics, accounting principles
Conceptual Models - mental models of the system and how it works

Figure 10
Sample of Completed Evaluation Form by Expert Panel

SOFTWARE EVALUATION INSTRUMENT

Factors important for learning in general, and for learning of software packages can be identified as those which:
a. support Speed, that is reduce performance time;
b. alleviate Memory load, that is promote retention or not require it;
c. minimize Mental effort, that is reduce mental strain;
d. provide Psychological comfort, that is reduce anxiety, promote peace of mind.

The purpose of this task is to formulate a mapping of software design and assistance features with these learning factors. You are asked to do this in three stages.

1. Assign a weighting out of 100% to each of the learning factors, as defined above - speed, memory, effort, comfort - to indicate the expected importance they have for your own learning of a package. Equal weighting is permitted but weights must add to 100%. Place weights in the brackets below the learning factors.

2. Rank the design features listed below, from 1 to 12, then the assistance features following, from 1 to 19, according to importance for your learning and use. Place your assigned rank in the rank column. Equal rankings permitted.

3. Match these features with the associated learning factor(s) you expect to be supported by the inclusion of the particular feature in software by placing an 'X' in the box.

e.g.	Speed	Memory	Effort	Comfort
Feature 1	X		X	
Feature 2		X	X	X
Etc.				

Design Features:	Rank	Speed (50)%	Memory (20)%	Effort (10)%	Comfort (20)%
1 Interface Command Structure - menus, commands, etc.	1	X	X	X	X
2 Depth of Structure	12	x	x		x
3 Logic of Structure	3	x	x	X	X
4 Consistency	10		x	x	x
5. Screen Design - cosmetic/aesthetics - meaningful grouping - language, wording	8	x	X		X
6. Flexible	9	x			X
7. Error Trapping/Recovery	7	x	X		x
8. System Response Time	2	x			x
9. Data Input - full screen, line editor	6	x		X	x
10. Macros	5	X	x		x
11 Autosave/backup	11	X		x	x
12 Auto-adjustment	4	x		x	x
13 Auto-formatting	13				x

5.A.1. **Characteristics of the panel members**

For this study, individuals were sought who could be considered fairly expert in the use of many packages, of diverse operation and functions, for developing Decision Support Systems for their own use, or for use by others. Survey forms were given out to 100 individuals, from a variety of working environments. (See Section 4.A.4.). Sixty seven questionnaires were returned. However, nine of these were omitted from the analysis because the participants failed to meet the experience level requirements desired for this portion of the research. This was ascertained from the information reported on the Proficiency Questionnaire, included with the survey kit. (See Appendix A). The number of packages used, frequency of use, and levels of expertise determined eligibility for the study, as explained in Chapter 4.B.4.

In terms of job function, the breakdown was as follows:

Distribution of Sample by Job Function	
System analysis type functions	73%
Management Information System (MIS) Professors	15%
Managers	6%
MIS Teaching Assistants and Graduate Students	6%

The system analysis functions reported included computer analyst, database management system consultant, database administrator, systems engineer, systems development manager, systems programmer, computer services manager, systems user support.

The self-assessed package experience levels reported were as follows:

Distribution of Sample by Experience Level	
Expert	63%
Intermediate	37%

Those reporting themselves as intermediates were considered appropriate for the study, given the frequency of use, the number of packages used, the years of experience reported, and the fact that our study is an exploratory one.

The number of packages, which were reported used on a regular basis, ranged from 1 to 17, were broken down as follows:

Distribution of Sample by No. of Packages Used	
< 5 packages	18%
5 packages	15%
6 packages	9%
7 packages	18%
> 7 packages	40%

Not surprisingly, given the population polled, none of the panel reported having low competence in quantitative courses. The breakdown was as follows:

Distribution of Sample by Quantitative Competence	
Expert	70%
Intermediate	30%

The number of years of work experience ranged from 1.5 years to 30 years, the average being 8.7 years. The breakdown was:

Distribution of Sample by Years of Work Experience	
< 3 years	14%
3 years	18%
4 - 6.5 years	14%
7 years	12%
> 7 years	42%

The panel was predominantly male, which is the current characteristic of the population targeted:

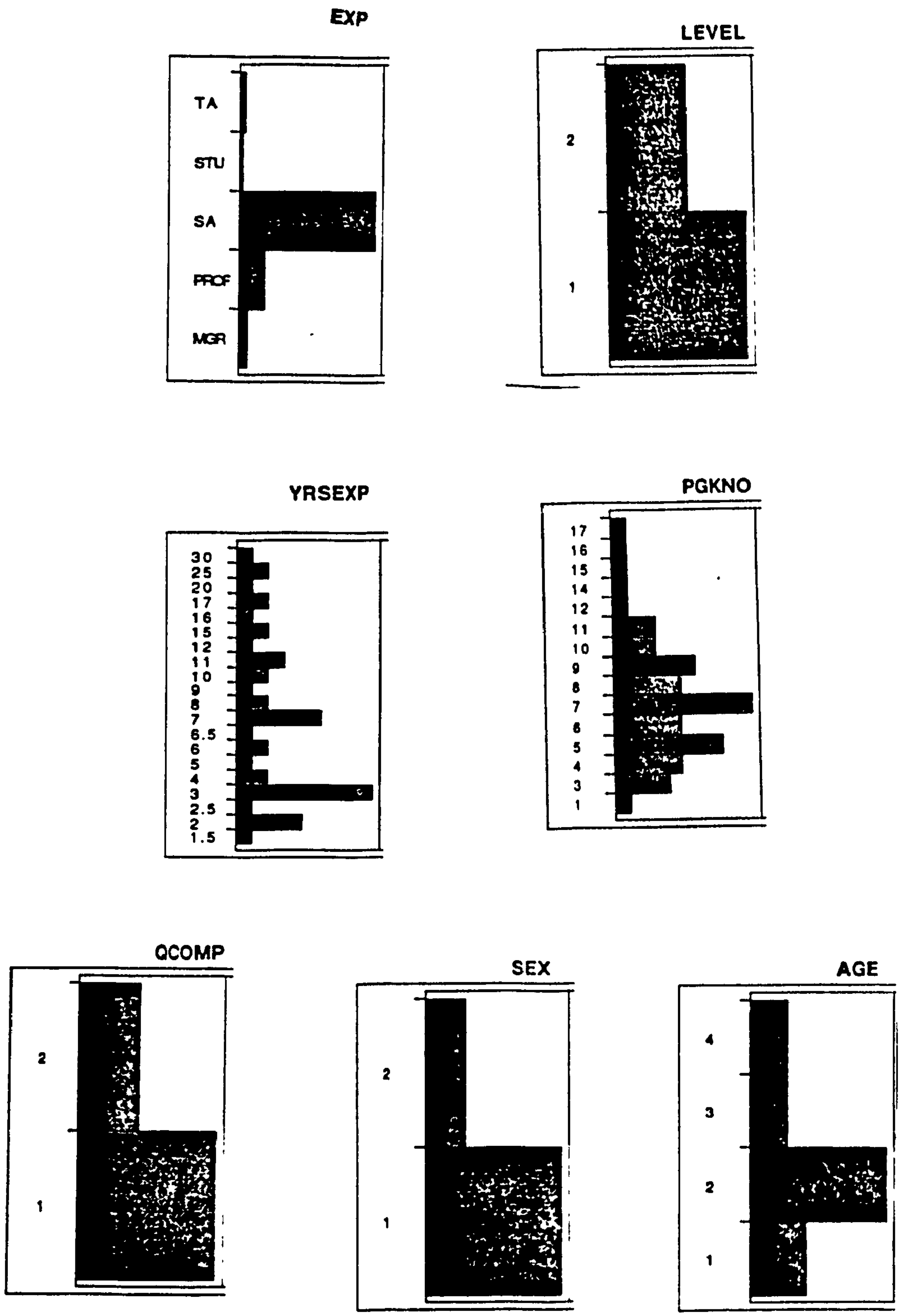
Distribution of Sample by Gender	
Males	77%
Females	23%

The age of respondents was as follows:

Distribution of Sample by Age	
< 25 years	22%
25 - 34	50%
> 34	28%

The histograms of these statistics are attached in Figure 11. We discuss the results of the rank of the features in the next section.

Figure 11
Histograms of
Expert Panel Profiles



5.A.2. **Analyses of consensus on rankings**

The first proposition stated:

Proposition 1.a. - There will tend to be agreement on the relative rankings assigned to the Design and Assistance Features identified as contributing to **ease of use**.

We were interested in knowing whether expert users perceived the various Design and Assistance Features to have the same level of importance for **ease of use**. We, therefore, asked the panel to rank the Features separately, for Design and Assistance, in terms of the importance they perceive them to have for **ease of use.** We applied multivariate analyses and Friedman's H statistic to the data to determine if the differences between the average ranks associated with each individual feature were statistically significant. We analysed the agreement among our experts using Kendall's coefficient of concordance. (Meddis, 1984). The findings, for this aspect of the study, are discussed under the headings:

5.A.2.a. Design Features;
5.A.2.b. Assistance Features;
5.A.2.c. Conclusions.

5.A.2.a. Design Features

The average Ranks assigned to the thirteen Design Features, which are summarized in Table 4 following, along with their frequencies. The table is arranged in order of the average ranks. Histograms of the frequencies are shown in Figures 12.a. and 12.b. As the histograms show density, no scales are shown. However, the frequencies can be read in the table.

Table 4 **SUMMARY of DESIGN FEATURES by RANK**								
Features	**Ranks** Coefficient of Concordance W = .39, p < .01							**Relative Variance of the Individual Ranks**
	Rank Order	**Average Ranks**	**Std Dev**	**Relative Frequencies of Ranks** 1-2	3-5	6-10	11-13	
F1 - Command Structure	1	2.7	2.2	66	23	11	0	11
F4 - Consistency	2	3.7	2.5	43	38	19	0	15
F3 - Logic of the Structure	3	4.6	3.0	32	42	26	0	20
F5 - Screen Design	4	5.8	3.3	25	27	40	9	25
F8 - System Response	5	6.2	3.5	7	40	51	2	28
F6 - Flexibility	6	6.4	2.5	17	33	44	6	15
F9 - Data Input	7	7.3	3.2	8	31	49	12	23
F7 - Error Trapping/Recovery	8	7.4	3.5	18	19	44	19	28
F2 - Depth	9	8.0	3.4	4	25	47	24	27
F10 - Macros	10	9.3	2.5	0	20	53	27	14
F11 - Autosave	11	9.5	3.2	0	20	35	45	24
F12 - Autoadjustment	12	9.7	2.8	0	17	39	44	18
F13 - Autoformatting	13	10.4	2.4	0	6	41	53	13

Figure 12a.
Histograms of
Relative Frequencies of Ranks
for each Design Feature

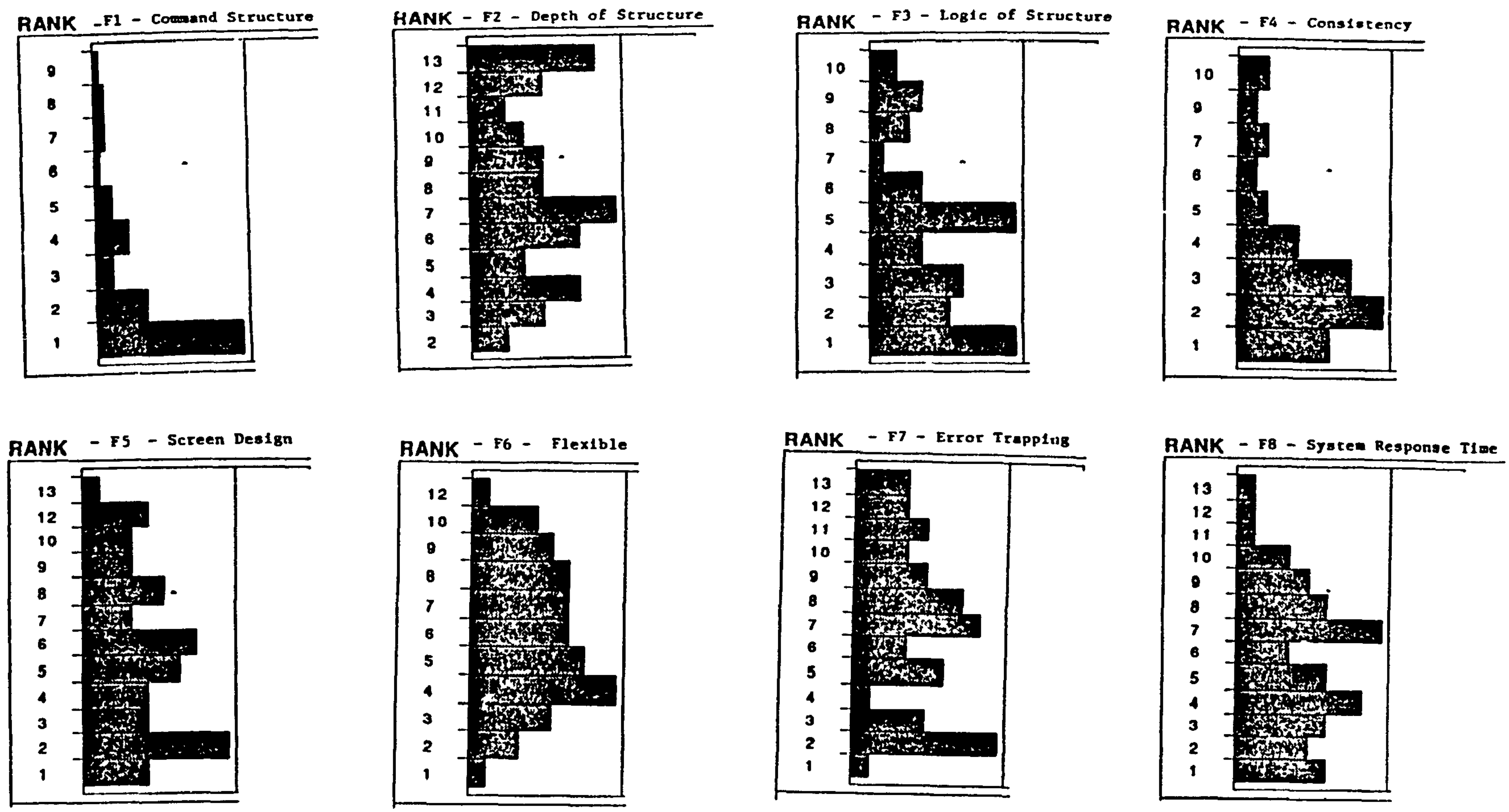
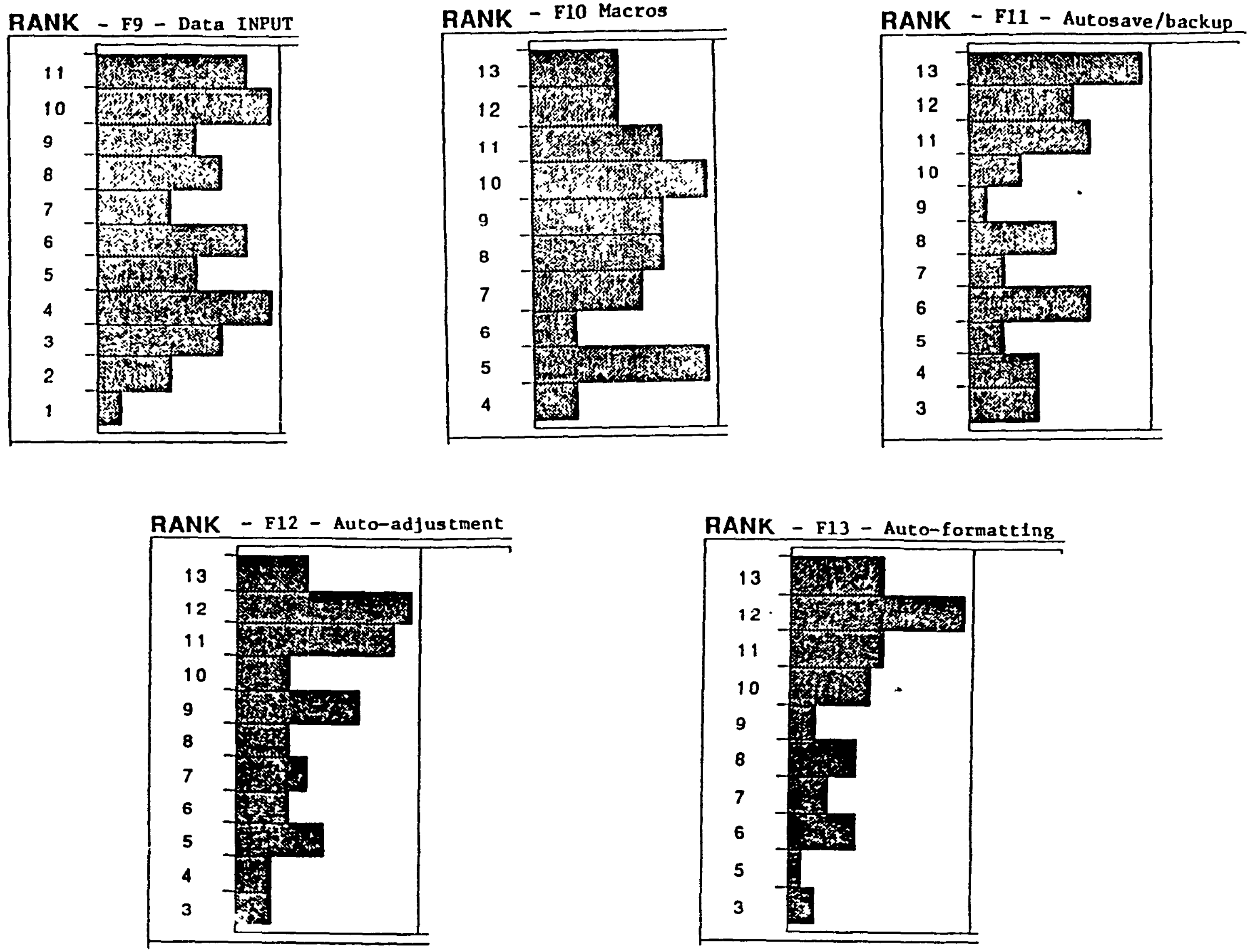


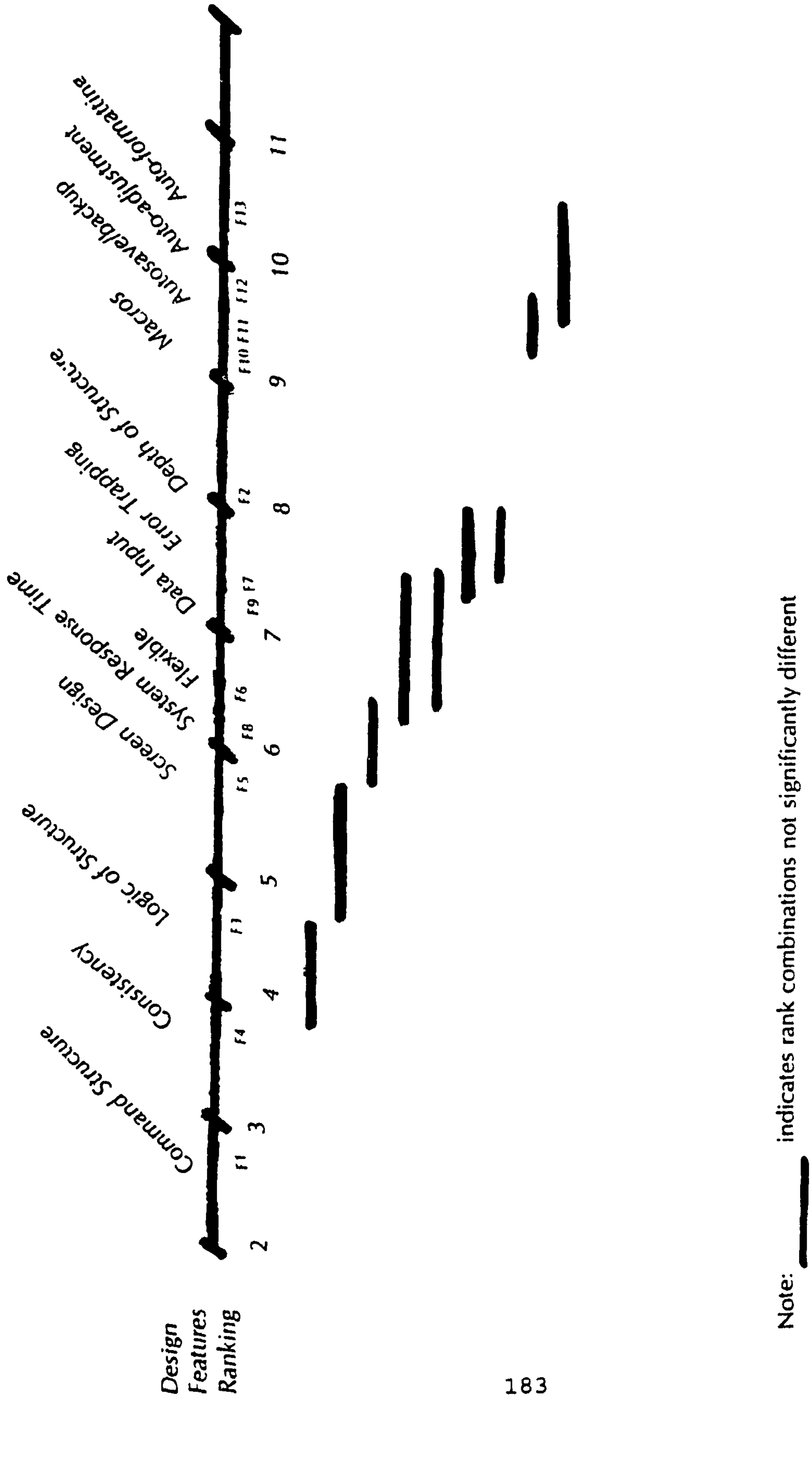
Figure 12b.
Frequencies of Design Features Ranks



The degree to which the panel agreed on the rankings they assigned to the features overall was not very high. This was based on Kendall's coefficient of concordance measure, W, where a value of 0 indicates no agreement and 1 indicates perfect agreement. (Meddis, 1984). Our measure of concordance of the ranks assigned by our expert panel was 0.39. Although the agreement was low, the level of significance was $p < .001$ which would seem to indicate that the ranks were not assigned at random and that there may be a tendency for our experts to agree on certain features. We, therefore, examined the average ranks.

There were significant differences, at $p < .01$, in the average ranks assigned to the individual features based on Friedman's H statistic and the multivariate analyses performed. (Meddis, 1984). In Figure 13, we have attempted to show these relationships, indicating by a solid line those features which are not significantly different from each other. All others are significantly different. Figure 13 indicates that F1 - Command Structure was of primary importance to our expert panel and that the average rank assigned to it was significantly different from the average rank assigned to all other features. F4 - Consistency and F3 - Logic of the Structure were also ranked significantly higher than the other features, except for F1 - Command Structure, but were not significantly different from each other, and F3 - Logic of the Structure was not different from F5 - Screen Design. Least important were F10 - Macros, F11 - Autosave/backup, F12 - Auto-adjustment and F13 - Auto-formatting, which average rank was significantly lower than all other features ranked above them. F13 - Auto-formatting was ranked significantly lower than all features, except F11 - Autosave/backup and F12 - Auto-adjustment. F10 - Macros was also significantly lower than all other features, except F11 - Autosave/backup and F12 - Auto-adjustment.

Figure 13
Significant Differences between Ranks
assigned to Design Features



As the Design Feature Command Structure, on average, was considered most important by our expert panel, our analysis suggests that the attention given to this aspect of package design in research is, indeed, warranted. The fact that prior research findings have been contradictory and inconclusive may be suggestive of a change in the way we evaluate the different approaches. For instance, perhaps the nature of the task should be considered, and the question posed should be 'Are menus better for structured tasks?' We also already mentioned the need to make the distinction between directed and non-directed menus. (Ogden, et. al., 1988; Lee, et. al., 1986; Whiteside, et. al., 1985; Shutoh, 1984; Napier, et. al., 1989; Dunsmore, 1981; Perlman, 1984).

Less specific attention has been given to Logic of the Structure, that is, 'Is the structure understandable; does it make sense to the user?', and to Consistency in design. These, however, are important to our panel. The research that has been done in these areas does indicate that lack of attention to these design features results in problems for users. (Karat, et. al., 1986; Schlager and Ogden, 1986; Maskery, 1985). These two features are more important for ease of use, according to expert users, than Screen Design, which includes the aesthetic arrangement and display of items on the screen, the meaningful grouping of items, language and wording used, et cetera, which has been given a fair amount of attention in research. (Burns, et. al., 1986; Ehrenreich and Porcu, 1982; Moses, et. al., 1980; Landauer, et. al., 1983; Black and Moran, 1982; Grudin and Barnard, 1985; Badre, 1982).

A fair number of studies have also looked at System Response (Dannenbring, 1983; Smith, 1983; Long, 1976; Bevan, 1981), and it is relatively important to our expert panel, having been ranked fifth.

The ranks assigned to Data Input and Depth of Structure are somewhat surprising. These were ranked fairly low, seventh and ninth, respectively. One would have thought that the number of levels needed to perform an operation and the mode of data entry, line entry versus full screen editing, would have been closely aligned to Command Structure and, therefore, ranked more highly. Some of the research studies do indicate, however, that in terms of Depth, users adapt to the number of levels. (Burns, et. al., 1986; Badre, 1982). There is evidence for this in the popularity of the Lotus, package, notwithstanding the many levels of its structure. On the other hand, two studies by Dunsmore (1984) on line versus screen editing indicated superior learning times and performance times for those using screen editing, perhaps because of the immediate confirmatory feedback on actions taken.

Another measure of agreement was evaluated to allow one to focus on individual features one at a time, as opposed to W which considers them as a whole. To determine the extent to which the panel agreed on the ranks assigned to the individual Design features, we derived a relative measure of the individual variances, relative to the maximum possible variance of the rank on a scale of one to thirteen. A zero relative variance would indicate perfect agreement among the panel as to the rank assigned to the feature and maximum relative variance of 1 would indicate half of the respondents assigning a rank of 1 to the feature and the other half assigning 13. This measure is shown in the last column of Table 4. Evidently, the most agreement was in the ranks assigned to those features ranked first and last. The variance is larger for those features ranked in the middle, nonetheless, their relative variance was also reasonably low, indicating that our panel of experts were fairly in agreement as to the ranks assigned to the individual Design features. They were particularily in agreement with respect to those given to Command Structure, Consistency, Flexibility, Macros, Autoadjustment and Autoformatting. The low measure of overall concordance may be due to the greater variance in the ranks assigned to the middle ranking features.

Our expectation, then, that expert designers and users would agree on the ranks assigned to the various Design Features was relatively unsupported with respect to the features taken as a whole, but was reasonably supported for ranks assigned to certain individual features. With respect to these latter, the level of attention given in research to the various features seems appropriate. However, in some instances, the research emphasis may need to be reconsidered, as suggested by the importance given to Consistency and Logic of the Structure, and the lesser importance given to Screen Design, mode of Data Input and Depth of the Structure by our expert panel.

The fact that our expert panel, though given the option, did not add any Design Features to the list may either suggest our listing of the important features was complete and appropriate, or that they had never given it any thought. One person added "Getting the right answer"; however, as this is not a feature but an outcome, it was not considered. The person also did not include it in the rankings made.

5.A.2.b. Assistance Features

Table 5, following, summarizes the nineteen Assistance Features according to the average Ranks assigned to them, with associated frequencies. Figures 14a. and 14b. show the histograms.

Table 5

SUMMARY of ASSISTANCE FEATURES by RANK

Features	Rank Order	Average Rank	Std Dev	Relative Frequencies of Ranks 1-2	3-5	6-9	10-12	13-18	Relative Variance of the Individual Ranks
	Ranks Coefficient of Concordance W = 25 p < 01								
F1 - Manuals	1	5.3	5.0	49	18	21	0	12	27
F2 - System Messages	2	5.9	4.3	31	33	20	10	6	20
F4 - Prompts	3	6.0	3.9	29	26	31	12	2	16
F8 - Index	4	7.7	4.7	18	28	25	16	13	24
F7 - Examples	5	7.9	5.1	20	31	21	10	18	29
F15 - Context Help	6	8.2	5.2	20	20	28	12	21	29
F3 - Tutorials	7	8.5	5.5	23	18	26	10	23	33
F6 - Defaults	8	8.5	4.5	10	31	23	22	14	22
F11 - Cautions	9	9.6	3.9	10	14	35	21	20	17
F5 - Templates	10	9.7	5.3	10	27	18	20	25	33
F14 - Instructive Feedback	11	10.4	4.7	4	22	24	24	26	24
F9 - Glossary	12	10.7	4.4	6	10	35	23	256	21
F19 - Conceptual Models	13	11.5	5.9	14	10	16	20	40	37
F13 - Navigational Aids	14	11.8	4.8	2	18	27	14	39	25
F16 - Expertise Accommodation	15	12.6	5.3	4	19	16	8	53	30
F12 - Checklist/Memory Jogs	16	13.4	3.6	0	12	10	23	55	14
F18 - Subject Matter Aid	17	13.6	4.6	2	12	14	16	56	23
F10 - Unsolicited Help	18	13.7	4.8	4	6	18	13	59	25
F17 - Restricted Options	19	14.0	4.4	6	8	6	15	65	21

Figure 14a.
Histograms of
Relative Frequencies of Ranks
for each Assistance Feature

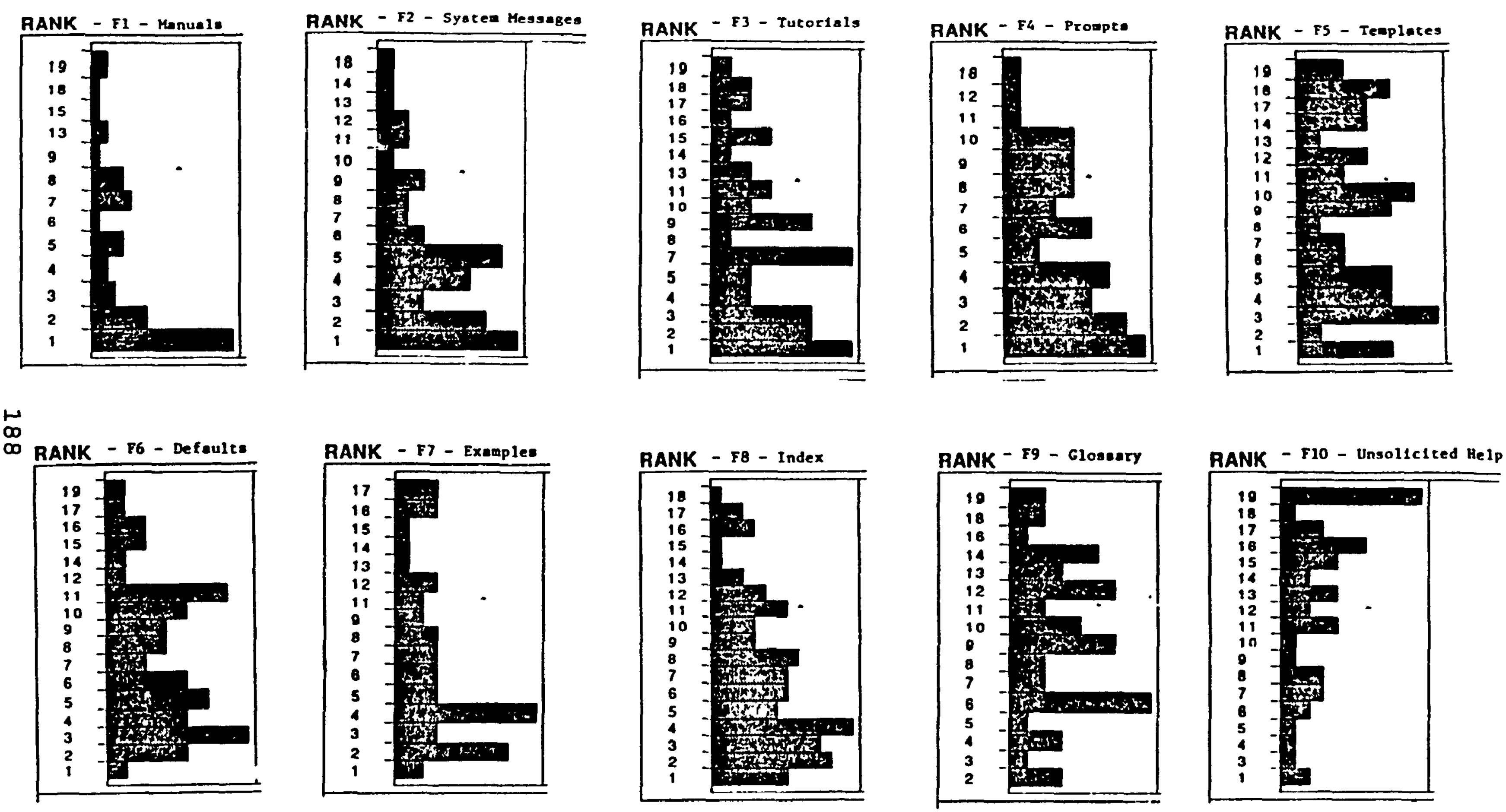
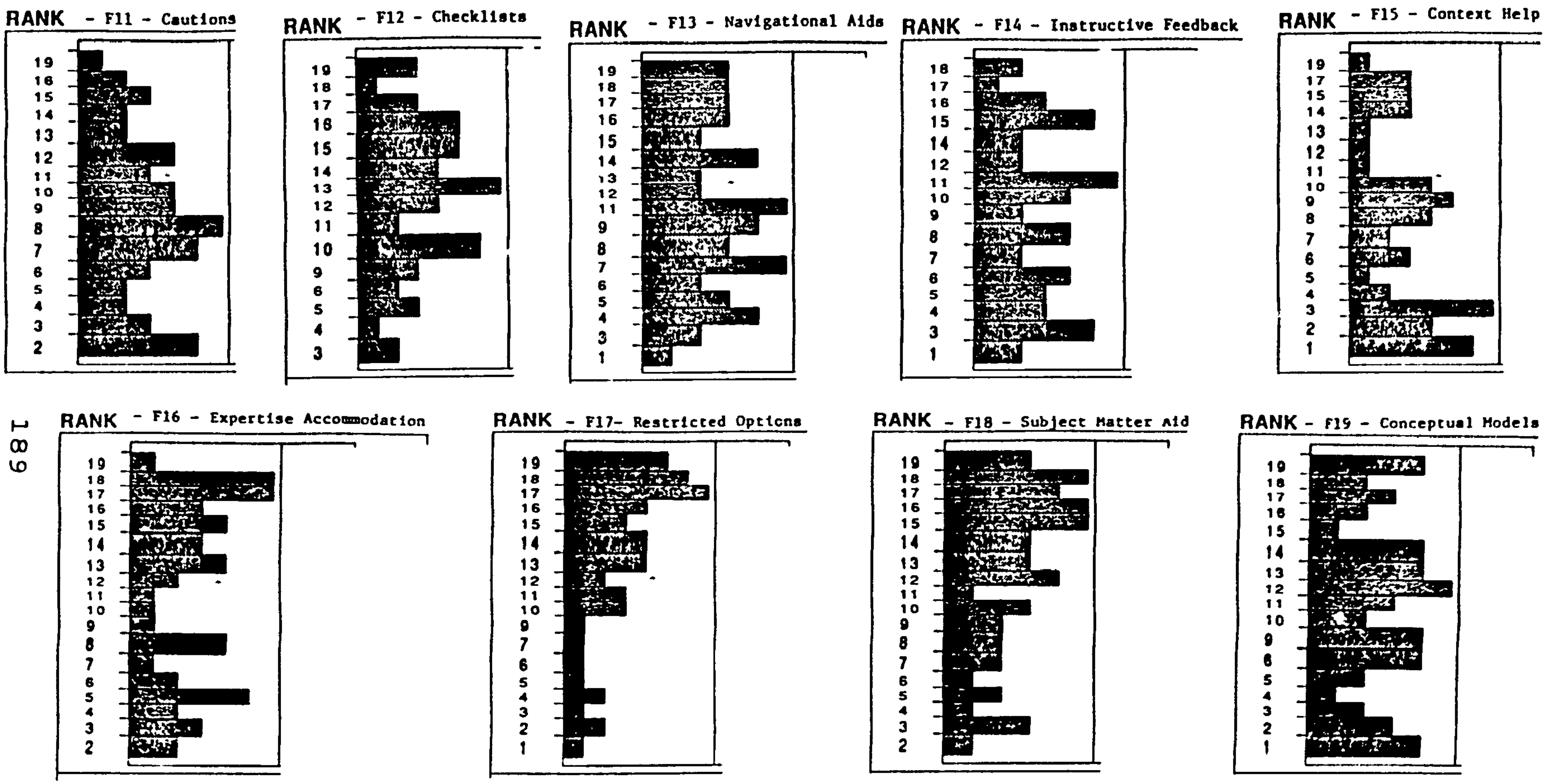
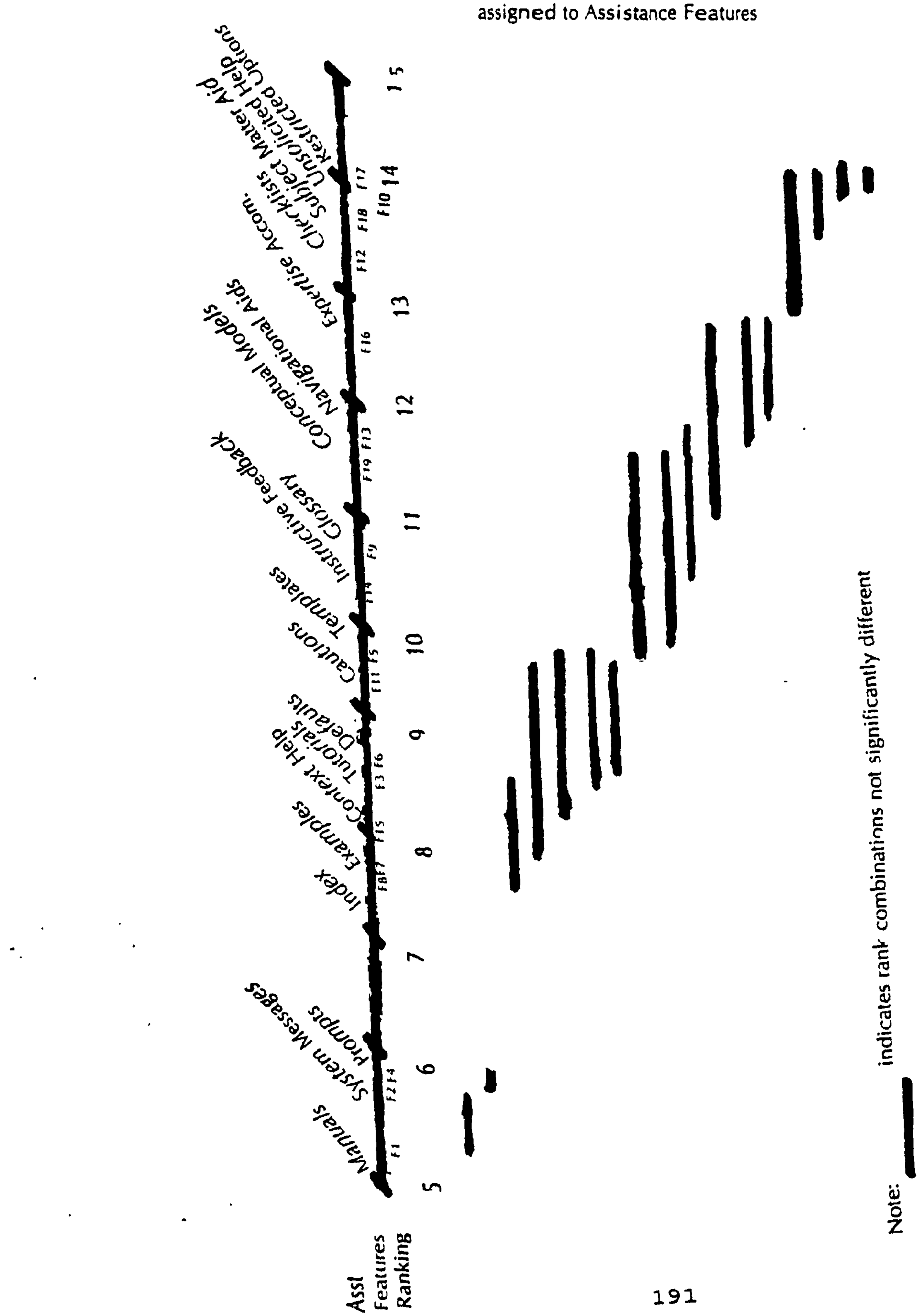


Figure 14b.
Frequencies of Assistance Features Ranks



As was found with the Design Features, the agreement of the panel was quite low, with respect to the rankings assigned to the Assistance Features overall. Kendall's coefficient of concordance, W, was 0.25. Again although the agreement was low, the significance level $p < .001$ would seem to indicate that the experts did agree on certain features which an analysis of the average ranks showed. Friedman's H statistic and the multivariate analyses showed significant differences between the average ranks of the individual features at $p < .01$. (Meddis, 1984). These differences are depicted in Figure 15, following. From this figure, we see that F1 - Manuals, F2 - System Messages and F4 - Prompts are not significantly different from each other, but are ranked significantly higher than all other Assistant Features. At the lowest ranking end of the scale, F12 - Checklists and Memory jogs, F18 - Subject Matter Aid, F10 - Unsolicited Help and F17 - Restricted Options were ranked significantly below all other Assistance Features, but not different from each other, nor from F16 - Expertise Accommodation. The variance of the average ranks assigned to each of the features was again quite low, indicating a fair degree of agreement among the expert panel members. The level of the variance was fairly different from feature to feature which, once again, may have accounted for the low overall concordance found.

Figure 15
Significant Differences between Ranks
assigned to Assistance Features



A first place average ranking assigned to Manuals, whether on- or offline, is not surprising. Many research studies have investigated the value of online versus offline manuals versus no manual, as well as the content of the manuals. (Relles, 1979; Dunsmore, 1980; Hiltz and Kerr, 1986; Carroll and Aaronson, 1988; Schlager and Ogden, 1986; Charney and Reder, 1986; Carroll, 1985). However, as with Command Structure, the findings have not been conclusive. While expert users indicate that they consider manuals important, some of the evidence suggests that they do not make use of them. The study by Hiltz and Kerr (1986) found that users only skimmed documentation and rarely read the offline documentation. The relevant question may more appropriately be what type of manual is most valuable. Work by Carroll and Aaronson (1988) found how-to-do-it and how-it-works manuals effective, while Schlager and Ogden (1986) found that a manual which conformed to an expert user's cognitive model of the system improved learning and retention.

System Messages, ranked second, have likewise been given attention in research. Shneiderman (1982) did a series of experiments on the impact of wording on performance. Ranked third are Prompts, that is, warnings of errors made, how to correct them and why they occurred. Less attention has been given to this area, both in research and in practice. Expert users, however, obviously consider this important.

Indices - quick references to commands and operations, Examples, and Context Help were also considered fairly important Assistance features. These were ranked fourth, fifth and sixth, respectively. Magers (1983) examined the effects of Examples and Context Help, and found performance was enhanced by the inclusion of these features. The emergence of Hypertext systems also attest to the growing interest in providing accessibility to information via a particular application of Indices. In this context, the Indices are embedded in text or graphics, which users can access for more detailed information on a topic.

Defaults, Cautions and Templates were considered as moderately important relative to the previous features discussed, being ranked eighth, ninth and tenth, respectively. Tutorials, ranked seventh, was also deemed moderately important by the panel. This is surprising. One would have thought that one would rely heavily on the tutorials provided with a package to foster ease of use and, therefore, assign it a higher rank. It may be that the respondents estimated **ease of use** strictly on the basis of ongoing use, and considered Tutorials for preliminary start up. There is anecdotal evidence to suggest that users do not avail themselves of the various assistance media provided to them with packages. Carroll and Mack (1984) and Hiltz and Kerr (1986) also demonstrated that users tend to prefer a learning-by-doing, exploratory approach to learning software.

Glossary and Instructive Feedback, in which the user's input is assessed for correctness and then communicated to the user, possibly with what the correct action(s) should be, were ranked fairly low, eleventh and twelfth, respectively. The study by Egan, et. al., 1987, suggests that research into diagnosis and remedial assessment of errors can be productive. One would have thought that users would find Instructive Feedback more important than seems to be the case.

Relative to the other features, the average ranks associated with Navigational Aids, Checklists and Memory Jogs, Subject Matter Aid and Unsolicited Help were low fourteenth, sixteenth, seventeenth and eighteenth, respectively. It is interesting that Conceptual Models, ranked thirteenth, Expertise Accommodation, ranked fifteenth, and Restricted Options, ranked nineteenth, should have been ranked so far down on the scale. These are some of the dominant themes in research. (Catrambone and Carroll, 1987; Carroll and Kay, 1985; Carroll and Olson, 1988; Wilson and Rutherford, 1989; Moray, 1987; Mancini, 1987; Schlager and Ogden, 1986; Bayman and Mayer, 1984; Goodwin and Sanati, 1986; Bostrom, Olfman and Sein, 1990). We thought that Conceptual Models would have been closely linked to the Design Feature, Logic of the Structure meaning, 'is clear, is understandable', which was ranked second. Our panel, being composed of experts, may already have their own conceptual models of their systems and so not see the benefit of being provided with one. In terms of Expertise Accommodation and Restricted Options, perhaps expert users do not consider these facilities for novice/expert transition important because they are already expert. In addition, some packages, such as Lotus$_{tm}$, DBase III+$_{tm}$ et cetera, already provide these facilities to some degree. In Lotus$_{tm}$, users can interact either by selecting commands using a cursor or by typing the first letter of the menu options, novice users usually choosing the former, and experts the latter, to the point of creating macros of commands. In DBase III+$_{tm}$, users have three levels of interaction - menu, dot prompt commands or programming language. In the case of both packages, some novices may never progress beyond menus, either by choice or from being unaware of the other facilities available, depending on their exposure and needs.

Our measure of the individual variances relative to the maximum possible variance of the rank on a scale of one to nineteen indicates low variances in the individual ranks. These are shown in the last column of Table 5. A zero relative variance would indicate perfect agreement among the panel as to the rank assigned to the feature and maximum relative variance of 1 would indicate half of the respondents assigning a rank of 1 to the feature and the other half assigning 19. It would seem that experts are reasonably in agreement on the ranks to be assigned to individual Assistance Features, in particular, the importance of System Messages and Prompts for ease of use and the relative unimportance of Cautions and Checklist and Memory jogs.

Once again, there was low support for agreement among the panel on the rankings assigned to the various Assistance Features taken as a whole but reasonable support for agreement on rankings assigned to individual features.

Research efforts have not been completely in line with expert perceptions of certain individual features. Little has been done in practical terms to integrate and study Prompts, that is, warnings of errors, why they occurred and how they may be corrected. The work that has been done with respect to intelligent systems have not, as yet, found its way into commercial software to any significant degree. Features less important to users, such as Conceptual Models and Expertise Accommodation, have been given more prominence in research.

5.A.2.c. <u>Conclusions</u>

Proposition 1.a., which supposed that our expert panel would assign the same importance to the various Design and Assistance Features, can be considered partially supported, based on the low degree of variance that was found on the average rankings given to the individual features, even if their concordance on the actual sequencing of the rankings was relatively low.

The rankings in the study by Holcomb and Tharp (1991), which was the only one found similar to our own, are not directly comparable to our findings. First of all, whereas we differentiated our **ease of use** features based on Design and Assistance Features, they did not. They were interested in Usability, which as they define it, overlaps, but does not equate to our definition of **ease of use.** Some of our features were, therefore, different. Nonetheless, the Design and Assistance Features that we found to be among the most important for our panel, namely Consistency and Manuals or Written Documentation, which is their term for manuals, were also those identified by their user sample. The Design Feature - Command Structure, which was ranked first by our panel, was not one of the items in their list. System Messages, ranked second in our Assistance Features listing, was at the lower end of the rankings in the Holcomb and Tharp study. However, all of their features were ranked above 75%, on a scale of 100%. It should be remembered that they did not separate Design from Assistance Features. In their study, the most important feature was Functionality, defined as being able to accomplish the task and to perform it reliably. We did not consider this to be a factor contributing to ease of use, though it is important for usability.

5.A.3. Analyses of consensus on learning dimension weights

The second proposition stated:

Proposition 1.b. - There will tend to be agreement on the relative weights assigned to the Learning Dimensions identified as contributing to **ease of use.**

We were interested in determining whether our expert panel considered the learning dimensions - **Speed, Memory, Effort, and Comfort** of equal importance in **ease of use,** and if not, whether they were consistent regarding the weights they did assign to them. A multivariate analysis was performed on the data to compare the average weights assigned to each of the dimensions. This will be discussed under the headings:

5.A.3.a. Design Features;
5.A.3.b. Assistance Features;
5.A.3.c. Conclusions.

5.A.3.a. Design Features

The importance of the learning dimensions for **ease of use** seems to go in decreasing order of Speed, Memory, Effort, Comfort. The average weights for each were 28.8%, 25.3%, 24.1%, 21.8%, respectively, with standard deviations of 16.5, 10.0, 10.7, and 12.3, respectively. The multivariate t-test showed no significant differences, $p < .3$, between these average weights. There was, however, quite a range of percentages assigned to each of the learning dimensions. Table 6, below, shows the distribution of percentages assigned The histograms are shown in Figure 16, following.

Table 6
Weights assigned to Design Features

	Speed	Memory	Effort	Comfort
N	47	47	47	47
Mean	28 8	25 3	24 1	21 8
S D	16 5	10 0	10 7	12 3
Lower 25%	5-20	5-20	5 20	5 10
25%	20-25	20-25	20-25	10 20
25%	25-40	25-30	25-30	20 30
15%	40-50	30-40	30-40	30-40
Upper 10%	50-75	40-50	40 50	40-60

In terms of **Speed**, the percentage of respondents assigning the different levels of weights were as follows:

Weights assigned to Speed											
Weights Assigned	75	50	45	40	35	30	25	20	.15	10	.05
% of Respond-ents	4 3	8 5	2 1	21 3	2 1	8 5	10 6	19 1	2 1	17 0	4 3

In terms of **Memory**, the weight assignment was as follows:

Weights assigned to Memory									
Weights Assigned	50	40	35	30	25	20	15	10	05
% of Respond-ents	4 3	10 6	2 1	23 4	19 1	23 4	6 4	8 5	2 1

Effort was assigned weights as follows:

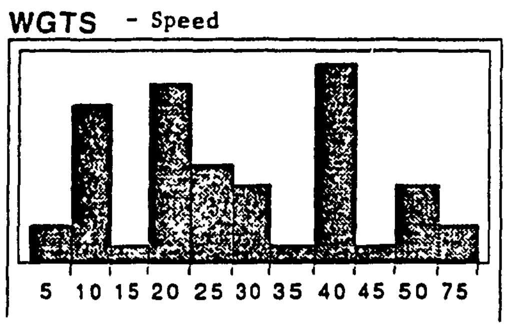
Weights assigned to Effort								
Weights Assigned	50	40	35	30	25	20	15	05
% of Respond ents	4 3	8 5	4 3	23 4	12 8	25 5	4 3	4 3

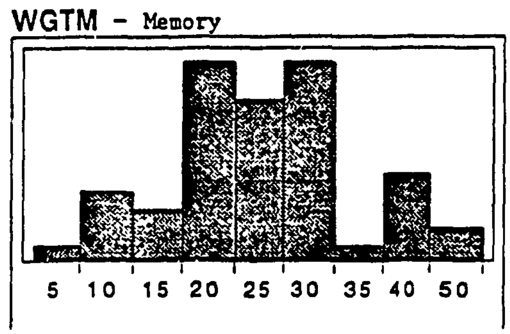
Comfort was assigned weights as follows:

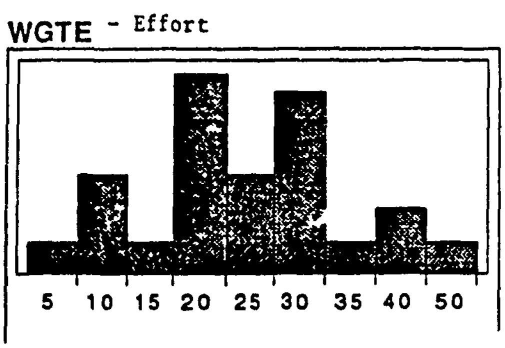
Weights assigned to Comfort										
Weights Assigned	60	50	40	35	30	.25	20	15	.10	05
% of Respond-ents	2 1	4 3	4 3	2 1	14 9	14 9	25 5	6 4	14 9	10 6

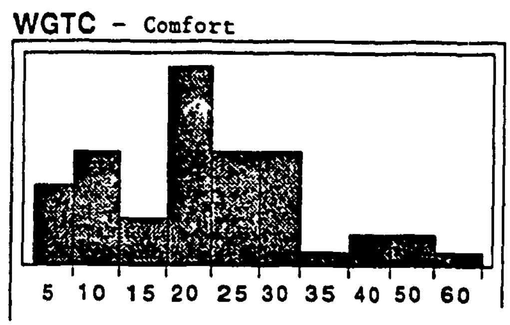
The Speed dimension has the most spread of responses. The weights .40, .20 and .10 were assigned by 21.3%, 19.1% and 17% of the responoents, respectively. The other dimensions have a more normal looking distribution, as seen on the histograms attached in Figure 16, with the majority of respondents assigning weights of. 20, .25, or .30. Memory was assigned these weights by 66%, Effort by 62% and Comfort by 55%, whereas Speed was so assigned by only 38%. It would seem that while users generally agree on the importance of the support which Design Features offer Memory, Effort and Comfort, their priorities are different for the support of Speed. As noted earlier however, the dimensions were equi-weighted, that is, not statistically different.

Figure 16
Histograms of
Weights (Design)









5.A.3.b. Assistance Features

The average weights assigned to Speed, Memory, Effort and Comfort for Assistance Features were 27.3%, 25.0%, 24.9%, 22.8%, respectively, with standard deviations of 16.3, 10.4, 9.2, 11.0, respectively. The distribution is comparable to that assigned to Design Features, with Speed given more weight than the other dimensions. Memory and Effort are about the same weight, and Comfort is given the least weight. Again, no significant differences, $p < .6$, were found between any of the dimensions and the range of weights assigned was wide, as seen in Table 7, below, which shows the distribution of weights. The histograms are given in Figure 17, following.

Table 7
Weights assigned to Assistance Features

	Speed	Memory	Effort	Comfort
N	43	43	43	43
Mean	27 3	25 0	24 9	22 8
S D	16 3	10 4	9 2	11 0
Lower 25%	5-10	10-20	5 20	5 15
25%	10-25	20-25	20-25	15 20
25%	25-40	25-35	25 30	20 30
15%	40-50	35-40	30 38	30 40
Upper 10%	50-75	40-50	38 50	40 50

In terms of frequencies, the following weights were assigned to **Speed**:

Weights assigned to Speed										
Weights Assigned	75	50	45	40	35	30	25	20	10	05
% of Respond-ents	4 7	7 0	2.3	14 0	2 3	9 3	18 6	16 3	23 0	2 3

Memory was given weights as follows:

Weights assigned to Memory								
Weights Assigned	50	40	35	30	25	20	15	10
% of Respondents	2 3	18 6	4 7	9 3	18 6	23 3	11 6	11 6

The weights assigned to **Effort** were as follows:

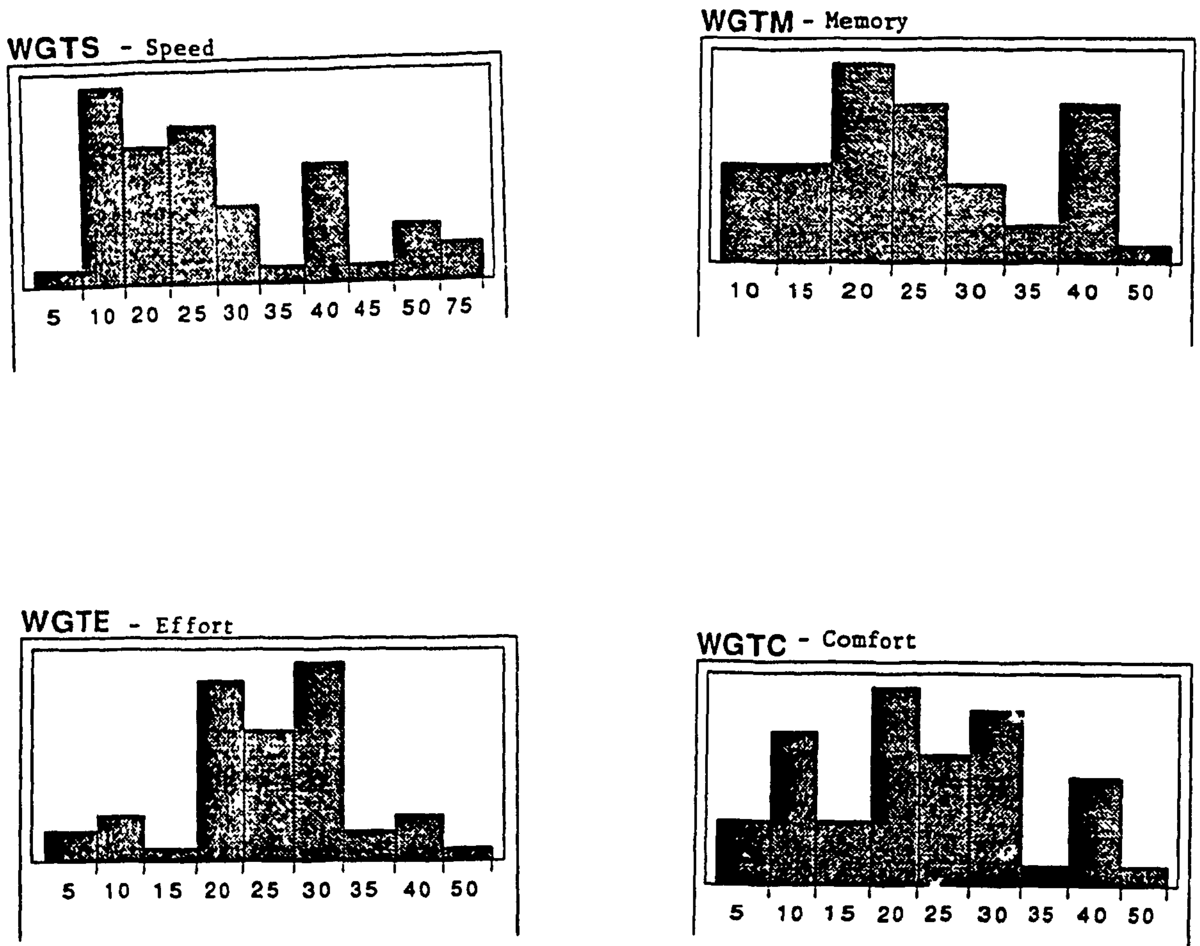
Weights assigned to Effort									
Weights Assigned	50	40	35	30	25	20	15	10	05
% of Respondents	2 3	7 0	4 7	27 9	18 6	25 6	2 3	7 0	4 7

In the case of **Comfort**, the weights were assigned as follows:

Weights assigned to Comfort									
Weights Assigned	50	40	35	30	25	20	15	10	05
% of Respondents	2 3	11 6	2 3	18 6	14 0	20 9	7 0	16 3	7 0

The weights assigned to Speed, Memory and Comfort were quite variable. Only 44.2%, 51.2% and 53.5%, of respondents assigned weights of .20, .25, and .30 to Speed, Memory and Comfort, respectively, compared to 71.1% who assigned these weights to Effort. A relatively large portion, 14%, 18.6%, and 11.6%, assigned a weight of .40 to Speed, Memory and Comfort, respectively, and 23%, 11.6% and 16.3% assigned a weight to .10 compared to the percentage assigning these weights to Effort, where only 7% of respondents assigned weights of .40 and .10. A more even distribution was found for the Effort dimension, with the majority of respondents, as we noted, 71.1%, assigning weights of .20, .25 and .30. These were the weights most frequently assigned to Memory, Effort and Comfort for Design Features. This can be seen more clearly on the histograms attached in Figure 17. It would seem that, whereas users agree on the importance of the support of Effort needed to be given by Assistance Features, they have different priorities for Speed, Memory and Comfort, though these differences are not significantly different on average.

Figure 17
Histograms of
Weights (Assistance)



5.A.3.c. Conclusions

The assignment of weights to **Speed, Memory, Effort and Comfort** was not significantly different for Design Features nor for Assistance Features, and the dimensions were considered equally important. Essentially, based on the frequencies, the panel had greater agreement on the importance of the support which Design Features offer Memory, Effort and Comfort, in terms of the actual weights assigned, but have less concensus for the support of Speed, as seen in the greater variability in weights assigned to this learning dimension. In terms of Assistance Features, there was more concensus on the support of Effort, but less concensus on the support necessary for Speed, Memory and Comfort. Nonetheless, as the differences were not significant, there is no evidence on which to reject Proposition 1.b.

5.A.4. **Analyses of consensus on links between features and learning dimensions**

The third proposition stated:

Proposition 1.c. - There will tend to be agreement on the links made between Design and Assistance Features and the Learning Dimensions they support.

We were interested in ascertaining whether expert users agreed on which Learning Dimensions were supported by each of the Design Features and each of the Assistance Features. We examined the relative frequencies using a cutoff of 50% of respondents, or better, as indicative of the Feature being identified with a particular Learning Dimension. This will be discussed under the following sections:

5.A.4.a. Design Features;
5.A.4.b. Assistance Features;
5.A.4.c. Conclusions.

5.A.4.a. Design Features

Table 8, following, shows the Learning Dimensions which were associated with each of the Design Features, with the relative frequency with which each dimension was associated with the feature by the expert panel.

Table 8

LEARNING DIMENSIONS associated with DESIGN FEATURES

Features	Learning Dimensions Supported	Relative Frequencies			
		S	M	E	C
F1 - Command Structure	SMEC	66	54	69	69
F2 Depth	SE	52	48	61	32
F3 - Logic	SME	50	65	69	48
F4 - Consistency	MEC	37	66	74	55
F5 - Screen Design	MEC	31	56	76	93
F6 - Flexibility	E	49	34	77	32
F7 - Error Trapping/Recovery	C	456	32	46	61
F8 System Response	S	95	11	13	35
F9 Data Input	SEC	59	26	77	59
F10 Macros	SE	74	42	60	25
F11 - Autosave/Backup	C	36	26	43	65
F12 - Autoadjustment	E	48	21	55	47
F13 Autoformatting	SE	50	20	59	45

Legend S - Speed, M - Memory, E - Effort, C - Comfort

The table shows the Design Features to be associated most with the **Effort** dimension. Ten of the thirteen features were associated with this dimension. **Memory** was the least mentioned of the dimensions. It was associated with only four Design Features. We had expected both Memory and Effort to be the dimensions most often associated with the design features and Comfort the least. The Learning Dimensions which we had expected to be associated with the Design Features can be seen in Table 10 which follows in Section 5.A.5.a. The features, distinguished according to the learning dimensions, are listed below.

Features associated with the **Speed** dimension were:

F1 - Command Structure
F2 - Depth of the Structure
F3 - Logic of the Structure
F8 - System Response
F9 - Data Input
F10 - Macros
F13 - Autoformatting

Features associated with the **Memory** dimension were:

F1 - Command Structure
F3 - Logic of the Structure
F4 - Consistency
F5 - Screen Design

Features associated with **Effort** were:

F1 - Command Structure
F2 - Depth
F3 - Logic of the Structure
F4 - Consistency
F5 - Screen Design
F6 - Flexibility
F9 - Data Input
F10 - Macros
F12 - Autoadjustment
F13 - Autoformatting

Features associated with **Comfort** dimension were:

F1 - Command Structure
F4 - Consistency (borderline)
F5 - Screen Design
F7 - Error Trapping and Recovery
F9 - Data Input
F11 - Autosave/Backup

In summary, the results indicate that the panel differentiated the Design Features according to the Learning Dimensions, associating more features with **Effort** than with any of the other dimensions, though a fair number were associated with **Speed** and **Comfort**. The least number of features was associated with **Memory**.

5.A.4.b. Assistance Features

Once again, relative frequencies of 50% of respondents, or better, were used to determine which Learning Dimensions were associated with each of the Assistance Features. Table 9, following shows the Learning Dimensions which were associated with the Assistance Features, with the relative frequencies.

Table 9

LEARNING DIMENSIONS associated with ASSISTANCE FEATURES

Features	Learning Dimensions Supported	Relative Frequencies			
		S	M	E	C
F1 - Manuals	MEC	33	63	72	65
F2 - System Messages	SC	57	47	47	60
F3 - Tutorials	EC	24	46	70	68
F4 - Prompts	SEC	51	49	72	60
F5 - Templates	ME	44	68	52	42
F6 - Defaults	SE	60	44	62	33
F7 - Examples	EC	17	44	62	67
F8 - Index	MEC	33	52	56	52
F9 - Glossary	M	31	59	43	39
F10 - Unsolicited Help	EC	30	28	57	74
F11 - Cautions	C	33	41	43	64
F12 - Checklists/Memory Jogs	M	26	60	36	48
F13 - Navigational Aids	E	47	39	65	45
F14 - Instructive Feedback	EC	24	31	53	65
F15 - Context Help	MEC	34	53	62	66
F16 - Expertise Accommodation	E	48	29	73	42
F17 - Restricted Options	0	27	43	47	47
F18 - Subject Matter Aid	EC	29	43	55	55
F19 - Conceptual Models	EC	26	41	61	55

Legend S - Speed, M Memory, E - Effort, C - Comfort

Interestingly, it seems that **Comfort** and **Effort** are the learning dimensions most often associated with the Assistance Features. Twelve and fourteen features out of nineteen, respectively, were identified by the panel with these dimensions. We had expected it to be Comfort and Memory. We had expected **Speed** to be the dimension least associated with the assistance features and this, indeed, was the case. The Learning Dimensions which we had expected to be associated with the Assistance features are given in Table 11, in the Section 5.A.5.b. following. The features are differentiated below according to each of the learning dimensions.

The features associated with the **Speed** dimension were:

F2 - System Messages
F4 - Prompts
F6 - Defaults

Those features associated with **Memory** were:

F1 - Manuals - on- and offline
F5 - Templates
F8 - Index
F9 - Glossary
F12 - Checklists/Memory Jogs
F15 - Context Help

Effort was associated with:

F1 - Manuals
F3 - Tutorials
F4 - Prompts
F5 - Templates
F6 - Defaults
F7 - Examples
F8 - Index
F10 - Unsolicited Help
F13 - Navigational Aids
F14 - Instructive Feedback
F15 - Context Help

F16 - Expertise Accommodation
F18 - Subject Matter Aid
F19 - Conceptual Models

The features associated with the **Comfort** dimension were:

F1 - Manuals
F2 - System Messages
F3 - Tutorials
F4 - Prompts
F7 - Examples
F8 - Index
F10 - Unsolicited Help
F11 - Cautions
F14 - Instructive Feedback
F15 - Context Help
F18 - Subject Matter Aid
F19 - Conceptual Models

As with the Design Features, the expert panel differentiated the Assistance Features according to the four Learning Dimensions we identified. They mostly associated **Effort** and **Comfort** with them, and **Speed** least of all.

5.A.4.c. Conclusions

The relative frequencies indicate that the expert panel differentiated the Design and Assistance Features according to the four Learning Dimensions. In particular, Design Features were most often associated with **Effort** and least often with **Memory**. Assistant Features were most often associated with **Effort** and **Comfort** and least often with **Speed**. Again, there was no evidence on which to reject Proposition 1.c.

5.A.5. **Analyses of consensus correspondence with theoretical expectations**

The fourth proposition stated:

Proposition 1.d. - The agreements found in Proposition 1.c. will match with expectations derived from the literature.

To the extent that our expert panel agreed on which Learning Dimensions were supported by each of the Design and Assistance Features, we were interested in knowing whether their perceptions matched the associations which we had constructed from the Human Factors and Learning literature. We will discuss these possibilities under the following sections:

5.A.5.a. Design Features;
5.A.5.b. Assistance Features;
5.A.5.c. Conclusions.

5.A.5.a. Design Features

As the summary shown in Table 10 indicates, the expectations were not completely supported. Notwithstanding, there was reasonable agreement with the perceptions of our expert panel.

Table 10				
COMPARISON of PANEL RESULTS on DESIGN FEATURES with EXPECTATIONS from the literature				
Features	**Expected**	**Learning Dimensions Supported**	**Dimensions Agreed On**	**Dimensions Disagreed On**
F1 - Command Structure	SME	SMEC	SME	C
F2 - Depth	SME	SE	SE	M
F3 - Logic	MEC	SME	ME	SC
F4 - Consistency	ME	MEC	ME	C
F5 - Screen Design	SMEC	MEC	MEC	S
F6 - Flexibility	ME	E	E	M
F7 - Error Trapping/Recovery	C	C	C	-
F8 - System Response	S	S	S	-
F9 - Data Input	SME	SEC	SE	MC
F10 - Macros	SME	SE	SE	M
F11 - Autosave/Backup	MC	C	C	M
F12 - Autoadjustment	SE	E	E	S
F13 - Autoformatting	SME	SE	SE	M
Legend S - Speed, M - Memory, E - Effort, C - Comfort				

As Table 9 indicates, only **F7 - Error Trapping/Recovery** and **F8 - System Response** which were associated with Comfort and Speed only, respectively, showed total agreement between our expectations and the perceptions of the panel. There were varying levels of agreement on the other features. The discrepancies, with respect to each of the Learning Dimensions, are listed and discussed below.

Speed

Features which we had expected to support **Speed**, and which were not found to be so perceived were:

F5 - Screen Design and F12 - Autoadjustment.

One feature that the panel perceived to support **Speed** which was not expected, was:

F3 - Logic of the Structure.

Memory

Features expected to support **Memory** which were not perceived to do so were:

F2 - Depth, F6 - Flexibility, F9 - Data Input, F10 - Macros, F11 - Autosave/Backup, and F13 - Autoformatting.

Effort

Features expected to support **Effort** were all as we had expected.

Comfort

One feature expected to support **Comfort**, but which was not perceived to do so, was:

F3 - Logic of the Structure.

On the other hand, those features that the panel perceived to support **Comfort** which were not anticipated, were:

F1 - Command Structure; F4 - Consistency; and F9 - Data Input.

Support for Memory showed the least agreement between our expectations and the panel's perceptions, whereby there were discrepancies in six out of the thirteen Design Features. This may have been due to the fact that we suggested this component as "offering support for memory or not requiring it" which may have lead to confused understanding of this dimension.

We had expected Autoadjustment to alleviate the user of one step, thereby decreasing the time spent on a task. It is not clear why the panel did not make this association. The same is true of Screen Design. We had expected that clear screen in terms of comprehensibility and readability would improve the time to perform a task. On the other hand, we should have anticipated the association between Logic of the Structure and **Speed**. If the package logic matches that of the user, this should have a favourable impact on speed of performance.

It is not clear why Depth of the Structure, Flexibility, Data Input, Macros, Autosave/backup and Autoformatting were not perceived to support **Memory**. One would have expected Depth, the number of levels required to execute a command or to traverse through menus and Flexibility, the number of ways to perform an operation would impose different requirements on Memory. One would also have expected line entry to impose greater memory requirements than full-screen editing, since line editors require that commands be remembered. Macros, which are the automating of repeated tasks into one, or a few, keystrokes, apart from speeding up the process, have the objective of alleviating the user from having to remember the detailed steps for performing an operation. Autosave/backup also alleviates the user from having to remember to perform this important task. The same applies to Autoformatting, whereby users need not know or remember the usual format required for the application, changing it only when necessary.

There were no discrepancies between expectations and the panel results, with respect to **Effort.**

We should have perhaps predicted **Comfort** to be associated with Command Structure, since there has been some speculation that different classes of users prefer different interface command structures. As such, one would expect that users matched with their preferred interface would be more at ease and comfortable.

We should also have foreseen **Comfort** as being associated with Consistency. 'Peace of mind' is likely to be affected by the number of different ways that exist for accomplishing operations. A user will quite possibly become irritated when an operation to perform a certain function is not the same in all parts of the software, for example, to exit from different modules in the package.

Mode of Data Input should also perhaps have been expected to support **Comfort,** given the popularity of the spreadsheet. It is not clear why Comfort was not associated with Logic of the Structure. The structure of a package which is clear and makes sense to the user should contribute to a reduction in stress and, therefore, to his psychological comfort.

In summary, the results indicate reasonable agreement, if not total agreement, between expectations derived from the literature and expert users' perceptions of which

Learning Dimensions are supported by each of the Design Features.

5.A.5.b. Assistance features

There is again reasonable agreement between the expectations derived from the literature and users' perceptions of the Learning Dimensions supported by the features. There was less agreement found for Assistance Features, however, than was found for Design Features. Table 11, following, shows these agreements.

Table 11 COMPARISON of PANEL RESULTS on ASSISTANCE FEATURES with EXPECTATIONS from the literature				
Features	Expected	Learning Dimensions Supported	Dimensions Agreed On	Dimensions Disagreed On
F1 - Manuals	MEC	MEC	MEC	-
F2 - System Messages	MC	SC	C	SM
F3 - Tutorials	MC	EC	C	ME
F4 - Prompts	MC	SEC	C	SE
F5 - Templates	MC	ME	M	EC
F6 - Defaults	SMEC	SE	SE	MC
F7 - Examples	MC	EC	C	ME
F8 Index	MC	MEC	MC	E
F9 - Glossary	MC	M	M	C
F10 - Unsolicited Help	MC	EC	C	ME
F11 - Cautions	MC	C	C	M
F12 Checklists/Memory Jogs	MC	M	M	C
F13 - Navigational Aids	SMC	E	-	SMEC
F14 - Instructive Feedback	MEC	EC	EC	M
F15 Context Help	MEC	MEC	MEC	-
F16 - Expertise Accommodation	SME	E	E	SM
F17 - Restricted Options	MEC	0	-	MEC
F18 - Subject Matter Aid	MC	EC	C	ME
F19 - Conceptual Models	EC	EC	EC	-
Legend S - Speed, M - Memory, E - Effort, C - Comfort				

Referring to Table 10, above, only F1 - Manuals, F15 - Context Help and F19 - Conceptual Models were perceived by our panel to support the same learning dimensions we had expected. On the other hand, there was no agreement at all on the learning dimensions supported by F13 - Navigational Aids, for which we expected Speed, Memory and Comfort to be supported. Instead the panel perceived **Effort** only. There was also no agreement on F17 - Restricted Options, for which we expected Memory, Effort and Comfort to be supported. The panel did not strongly perceive any dimensions to be supported by it. On the rest of the features, there were varying levels of agreement. Discrepancies are discussed in the context of the individual learning dimensions.

Speed

In terms of **Speed**, those features which we had not expected to support this dimension, but identified by our panel, were:

F2 - System Messages; F4 - Prompts; F5 - Templates.

The features we had expected to support **Speed** which were not identified by the panel were:

F13 - Navigational Aids; F16 - Expertise Accommodation.

Memory

Those features which we expected to support **Memory** which were not supported by our panel, were quite a few:

F2 - System Messages; F3 - Tutorials; F4 - Prompts; F6 - Defaults; F7 - Examples; F10 - Unsolicited Help; F11 - Cautions; F13 - Navigational Aids; F14 - Instructive Feedback; F16 - Expertise Accommodation; F17 - Restricted Options; and F18 - Subject Matter Aid.

Effort

One feature expected to support **Effort**, but not corroborated by our panel, was:

F17 - Restricted Options.

Those features perceived by the panel to support **Effort**, but not expected by us, were:

F3 - Tutorials; F4 - Prompts; F5 - Templates; F7 - Examples; F8 - Index; F10 - Unsolicited Help; F13 - Navigational Aids and F18 - Subject Matter Aid.

Comfort

Those features expected to support **Comfort** which were not so perceived by our panel, were:

F5 - Templates; F6 - Defaults; F8 - Index; F9 - Glossary; F12 - Checklists/Memory Jogs; F13 - Navigational Aids; and F17 - Restricted Options.

Those features which were expected to support **Memory** showed the least agreement with the panel's perceptions, with discrepancies in eleven of the nineteen Assistant Features. We suggested earlier that this may have been due to confusion arising from the definition we proposed which was "offering support for memory or not requiring it". This is followed by **Effort**, with discrepancies in nine of the nineteen features.

The most interesting discrepancy was for Restricted Options, which was not strongly perceived to support any of the dimensions, though we had expected Memory, Effort and Comfort. On reflection, this feature may not have been described with enough precision. We defined it as "Restricted to certain operations at certain junctures ". An example might have been more informative, such as when restrictions are used to assist the transition from novice to expert levels. Novices may find a more restricted environment beneficial. However, in a situation where certain classes of users have restricted access to certain options, there may, indeed, be no basis for assuming any of the learning dimensions to be associated with the feature.

Other significant discrepancies were for System Messages, Tutorials, Prompts, Templates, Defaults, Examples and Navigational Aids. We did not foresee System Messages supporting **Speed**; however, they were perceived to do so by the panel. We also associated

Memory with System Messages, since they provide information and may give reminders of actions to be taken, but the panel did not make this association. We are not able to account for these perceptions.

We had not anticipated Speed and Effort to be supported by Prompts, defined as, advising of errors, how to correct them and why they were made. While we can see where Effort might be a reasonable assignment to make, Speed is less evident, unless, in-so-far as time is reduced as a consequence of not having to search for the errors and the procedures to correct them. It is not clear why **Memory** was not associated with it, however, since Prompts not only remind one of current actions needed, but may also assist longterm memory by clearing up problems previously experienced.

There was no evidence of **Memory** and **Comfort** being perceived to support Defaults, by our panel, which is surprising. One would have thought that having the usual responses entered automatically for you, which you can then modify, in whole or in part, would foster peace of mind and psychological comfort for the user and alleviate memory load.

We expected Memory to be one of the dimensions associated with Examples. However, the panel did not perceive this to be so, and instead, identified Effort. This should probably have been expected since the provision of examples might, indeed, alleviate mental strain.

The panel did not perceive Speed, Memory or Comfort to be associated with Navigational Aids, but associated **Effort**, which we had not. Effort is probably reasonable, if one considers the facility provided by function keys to ease movement from screen to screen in a package. However, it is not clear, for the same reasons, why Speed, Memory and Comfort were not perceived.

Other discrepancies included:

- Templates were viewed as being associated with Speed, while we had not expected this dimension, but rather Memory and Comfort. We should have forseen Templates giving support to Speed, given that ready access to commands are likely to reduce search time in locating the proper commands.
- Effort associated with Index was not anticipated. It is not clear how a list of commands with explanation reduces the effort required to perform a task.
- Comfort was not found to be associated with Glossary, and Memory was not found to be associated with Unsolicited Help, though **Effort** was, according to the panel. On reflection, it does seem likely that having the computer provide unsolicited help at moments of difficulty detected by idle computer time, or after a number of incorrect actions, would alleviate **Effort**, since the user can take advantage of the assistance, that is, force it by non-action, rather than seek help from manuals, et cetera. For this same reason, it is not clear why Memory was not perceived to be supported by this feature.
- The panel did not attribute **Memory** to Instructive Feedback. It is baffling why this occurred. Instructive Feedback acts as a memory jog and minimizes effort since procedures on proper operation are provided, without the user having to search for them.
- We expected **Speed** and **Memory** to be associated with Expertise Accommodation;

however, the panel did not make this association. We expected the memory load on users to be reduced if the package permits them to have access to operations and functions appropriate to their level of expertise. We also expected that if experts can bypass certain operations which novices would require, the speed with which operations are performed should be affected.

- We had expected Cautions, which act as jogs to memory, should support **Memory**, but this was not identified by our panel, the reason for which is unclear.
- We expected Checklists/memory jogs to offer peace of mind, since they ensure that the user does not forget anything, but the panel did not associate Comfort with this feature, but only **Memory**. It is not clear why.

While there were discrepancies on the Assistance features, the Learning Dimensions reported by the panel suggest some support for our expectations for those features.

5.A.5.c. **Conclusions**

There was not complete support for Proposition 1.d.; nevertheless, there was reasonable agreement between what was expected from a review of the literature, in terms of the Design and Assistance Features and the Learning Dimensions they support, and the perceptions of the panel. In particular, full agreement between our panel's perceptions and our expectations was found for Design Features: F7 - Error Trapping/Recovery and F8 - Systems Response; and for Assistance Features: F1 - Manuals; F15 - Context Help, and F19 - Conceptual Models. There was no agreement on the Assistance Features, F13 - Navigational Aids; F17 - Restricted Options. There were varying levels of agreement on the other features. In particular, those Design Features expected to support Effort showed more agreement with our panel than those supporting Speed, Memory and Comfort. The Assistance Features showed less agreement for those features supporting Memory than the other dimensions, but there were considerable differences on the other dimensions as well. There was, in addition, less agreement on the Assistance Features than on the Design Features.

Discrepancies which exist may be a function of differences in understanding of the definition of the features, notwithstanding the glossary supplied to the panelists. Alternatively, they may be a function of the inadequacy of the sample size for the number of features being assessed. As well, it is possible that in some instances, our deductions may have been faulty. Another factor may be the bias of our sample to expert users. Of necessity, we targeted experts since novices cannot be expected to rank and assess what they do not know. As such, the responses of the experts are with respect to their requirement which are not necessarily the same for the average user.

It should be pointed out that the dimensions are not mutually exclusive, they may overlap each other. A feature that reduces effort, for example, probably leads to less memory load. It may also lead to reduced performance speed. Also, if memory and effort are minimized, the feature is likely to promote psychological comfort and peace of mind. Nonetheless, some learning dimensions are likely to be more frequently associated with some features than with others, and the results of the panel survey lend some support to this.

5.A.6. **Summary of findings in Study 1**

The results provided no evidence on which to reject Propositions 1.b., and Proposition 1.c., and gave evidence of partial support for Propositions 1.a. and 1.d. The propositions are restated below for reference.

Proposition 1.a. - PS - There will tend to be agreement on the relative rankings assigned to the Design and Assistance Features identified as contributing to **ease of use.**

Proposition 1.b. - S - There will tend to be agreement on the relative weights assigned to the Learning Dimensions identified as contributing to **ease of use.**

Proposition 1.c. - S - There will tend to be agreement on the links made between Design and Assistance Features and the Learning Dimensions they support.

Proposition1.d. -PS- The agreements found in Proposition 1.c. will match with theoretical expectations.

Legend: S - supported; PS - partially supported

The expert panel perceived the same general level of importance for each of the Design and Assistance Features individually but the degree of concensus on the rankings assigned to the features overall was low. The most important Design Features were identified as Command Structure, Consistency and Logic of the Structure. For Assistance Features, these were Manuals, System Messages and Prompts. Least important Design Features were Macros, Autosave/backup, Autoadjustment and Autoformatting. Least important Assistance Features were Checklists and Memory Jogs, Subject Matter Aid, Unsolicited Help and Restricted Options.

There were no significant differences between the weights assigned to the learning dimensions by the panel, both for Design and Assistance Features. Each of the dimensions was considered equally important. There was, however, a lot of variability in the weights reported, as the histograms reveal. There seems to be different priorities for the support of Speed by Design Features, as indicated by the high variability in the weights assigned to it, but there was more agreement, generally, on the support for Memory, Effort and Comfort, with most of the panel assigning weights of either 20, 25, or 30%. In terms of Assistance Features, the panel agreed more on the support given to Effort, with most assigning weights of 20, 25, or 30%, than they did on Speed, Memory and Comfort, to which they assigned a range of weights.

There was reasonable agreement among the panel on certain learning dimensions supporting certain of the Design and Assistance Features. Design Features were felt to support mostly the **Effort** dimension and, least of all, **Memory**. Assistance Features, on the other hand, support mostly **Effort** and **Comfort**, and **Speed**, least of all. There was less agreement on the Assistance Features than there was on the Design Features.

While there was not complete agreement between the panel results and our expectations from the Human Factors and Learning literature regarding which learning dimensions were supported by each of the Design and Assistance Features, there was reasonable agreement. Notwithstanding certain discrepancies, some learning dimensions seem to be more frequently associated with certain features than others. There was less agreement on Design Features' support of Memory and Assistance Features' support of Memory and Effort than on other dimensions. There was also less agreement on the Assistance Features than on the Design Features. We concluded Study 1 by examining a number of explanations for the discrepancies, ranging from differences in interpretation of the terminology, inadequacy of sample size and its expert sample bias, possible inappropriateness of theoretical conception, to the fact that the dimensions are not mutually exclusive. There is also the possibility that our sample was not truly experts and that our classification of them as such was inaccurate.

The results of Study 1 were reasonably confirming of our expectations, bearing in mind that the study is an exploratory one and that our view of **ease of use**, which offers a new perspective on the construct, is in the developmental stages and is not yet complete nor fully developed. We were able to identify certain important Design and Assistance Features for experts and to get some confirmation for the relevance of our proposed Learning Dimensions for the **ease of use** of business software. There is need, however, for further definition and operationalizing of **ease of use**, and the components we suggest comprise it, before a suggestion for using them as a basis for evaluating the **ease of use** of business software can be made. The potential for such assessment is not self-evident from the results but neither do they preclude this possibility. Further research is, therefore, indicated. Study 2 adds to this investigation in some measure.

The section following will present and discuss the results and findings from Study 2, in which we looked particularily at the most important Design and Assistance Features identified in Study 1, namely different Command Structures and Online Assistance strategies.

5.B. Study 2 - Quasi-experimental evaluation of effect of package and experience level differences, and other factors, on ease of use

In Study 1, we found that our expert panel identified certain design and assistance characteristics as being more important than others, namely Command Structure and on- and offline Manuals, respectively. Study 2 was concerned with the impact of differing package design and assistance features on **ease of use**, with respect to those which were identified. We were also interested in whether this may be moderated by differences in experience levels with packages of similar and dissimilar type in function and/or operation, as well as by other factors. The research questions posed were:

Research Question 2:

Is there a difference in perceived comfort and performance measures relating to **ease of use**, as a consequence of differing Design and Assistance Features?

Research Question 3:

Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to differences in experience levels?

Research Question 4:

Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to other factors, such as computer anxiety, gender, and quantitative competence?

In this section, we present the profile of the sample and the statistical analyses for this study. The analyses are discussed in three main parts:

a) univariate investigations of one explanatory variable at a time;

b) multiple regression to investigate relationships of the variables in the context of an overall model for predicting and explaining **ease of use**;

c) analyses of sub-tasks.

5.B.1. <u>Characteristics of the sample</u>

In Study 2, we sought individuals with a variety of experience with various packages. The general demographic profile of these individuals is shown in Table 12, following. In total, two hundred and forty three people took part in this study, of which nine were Statistics and MIS professors, and the rest were students from introductory courses in Statistics and Management Information Systems. These subjects were assigned to one of three package treatments, based on the convenience to them of the available time slots for the experiments. There were 99 subjects, or 40.7%, assigned to Treatment 1 - Merlin package; 93 subjects, or 38.3%, to Treatment 2 - Minitab; and 51 subjects, or 21.0%, to Treatment 3 - Extended Merlin package, which was the original Merlin plus a Hypertext-based online help index.

There were slightly more females than males in the sample, the breakdown being 132 females, or 54.5% and 110 males, or 45.5%. Except for the professors, the sample was taken from the student population; hence, most were less than 25 years of age, that is, 210, or 87.1%.

In terms of their perceived competence in quantitative courses, most subjects, that is 148, or 78.7%, reported Average competence; while thirteen, or 6.9%, reported Excellent competence; and twenty-seven, or 14.4%, reported Poor competence.

Table 12 General Demographics									
N - 243		Total		Merlin,		Minitab,		Extended Merlin,	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
Age	Total	241	100	97	100	93	100	51	100
	1- <25	210	87 1	78	80 4	86	92 4	46	90 2
	2 25-34	19	7 9	9	9 3	5	5 4	5	9 8
	3 35-44	10	4 2	8	8 2	2	2 2	0	0 0
	4 >44	2	0 8	2	2 1	0	0 0	0	0.0
Sex	Total	242	100	99	100	92	100	51	100.
	1- Male	110	45 5	51	51 5	37	40 2	22	43 1
	2 Female	132	54 5	48	48 5	55	59 8	29	56 9
Course	Total	243	100	99	100	93	100	51	100.
	1 Intro MIS	72	29 6	27	27 3	20	21 5	25	49 0
	2 Intro Stats	116	47 7	37	37 4	53	57 0	26	51 0
	3 Intro Stats	46	18 9	26	26 2	20	21 5	0	0 0
	4 Stats Profs	9	3 7	9	9 1	0	0 0	0	0 0
Quantitative Competence	Total	188	100	76	100	73	100	39	100
	1 Excellent	13	6 9	4	5 3	6	8 2	3	7 7
	2 Average	148	78 7	58	76 3	59	80 8	31	79 5
	3 Poor	27	14 4	14	18 4	8	11 0	5	12 8

The experience levels of the subjects could not be ascertained beforehand. As a consequence of this, some of the experimental levels were under-represented. The levels, as shown in Table 13 following, were represented as follows:

Table 13
Distribution of Experience Levels

Statistics Package Experience	Microcomputer Experience: Novice	Intermediate	Expert	Total
L1 - None, minimal	26	-	-	26
L2 - Varied, non-Stat	89	37	1	127
L3 - Varied, + 1 Stat	55	24	0	79
L4 - Varied, + >1 Stat	2	3	6	11
Total	172	64	7	243

Note The level categories are designated as follows
L1N - no computer or package experience, or minimal experience
L2N - novice with many packages, but not with statistical packages
L2I - intermediate with many packages, but not with statistical packages
L2E - expert with many packages, but not with statistical packages
L3N - novice with one statistical package, and other packages
L3I - intermediate with one statistical package, and other packages
L3E - expert with one statistical package, and other packages
L4N - novice with many statistical packages, and other packages
L4I - intermediate with many statistical packages, and other packages
L4E - expert with many statistical packages, and other packages

The levels were relatively evenly represented across each of the package treatments. The nine professors were assigned only to Merlin, as we were interested in the effect of experience with many different statistical packages including Minitab, on a package with which there was no previous experience. These individuals are excluded from analyses which compare package differences. Level 3 individuals mostly cited Minitab as the one package they already knew. None of the Levels had ever used Merlin. (See Chapter 4.B.4. for a full explanation of the establishment of these levels).

The data collected concerning general computer experience is summarized in Table 14, following. Most of the sample had no mainframe experience, that is, 136, or 58.4%. There were 97, or 41.6%, who had mainframe experience. A majority, however, had microcomputer experience - 205, or 85.8%, of which 141, or 79.2%, came from an IBM environment. Most considered themselves to have novice computer experience - 147, or 62.3%. There were 83 intermediates, or 35.2%, and 6 experts, or 2.5%. The frequency of use of the computer was varied: 26 never used the computer; 75 used it less than once per month; 27 used it once per month; 64, a few times per month; 23, a few times per week; 7, once per day; and 14 used it several times per day. The majority, or 81.3%, used it less than a few times per month.

Most people rated themselves as novice in the use of DOS, numbering 183, or 77.9%, and, of these, 83.7% used it less than once per month.

Table 14
General Computer Experience

		Total		Merlin,		Minitab,		Extended Merlin,	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
Mainframe Experience	Total	233	100	94	100	89	100	50	100
	1 Yes	97	41 6	45	47 9	36	40 4	16	32 0
	2 No	13o	58 4	49	52 1	53	59 6	34	68 0
Micro-computer Experience	Total	239	100	98	100	90	100	51	100
	1- Yes	205	85 8	93	94 9	72	80 0	40	78 4
	2 No	34	14 2	5	5 1	18	20 0	11	21 6
Type of Micro-Computer Used	Total	178	100	81	100	61	100	36	100
	1 IBM	141	79 2	597	72 8	53	86 9	29	80 5
	2 Macintosh	11	6 2	5	8 6	3	4 9	1	2 8
	3 Both	18	10 1	10	6 2	2	3 3	1	2 8
	4 Other	8	4 5		12 4	3	4 9	5	13 9
General Computer Experience Level	Total	236	100	95	100	91	100	50	100
	1 Novice	147	62 3	563	59 0	58	63 7	33	66 0
	2 Intermediate	83	35 2	5	36 8	31	34 1	17	34 0
	3 Expert	6	2 5	4	4 2	2	2 2	0	0 0
Frequency of Use of Computer	Total	236	100	94	100	91	100	51	100
	1 Never	26	11 0	7	7 4	11	12 1	8	15 6
	2 < Once/mth	75	31 8	30	31 9	27	29 6	18	35 3
	3 Once/mth	27	11 4	9	9 6	14	15 4	4	7 8
	4 Few times/mth	64	27 1	29	30 9	22	24 2	13	25 5
	5 Few times/wk	23	9 7	10	10 6	7	7 7	6	11 8
	6 Once/dy	7	3 0	1	1 1	5	5 5	1	2 0
	7 Several times/dy	14	5 9	8	8 5	5	5 5	1	2 0
DOS Experience	Total	235	100	95	100	89	100	51	100
	1 Novice	183	77 9	72	7^ 8	70	78 7	41	80 4
	2 Intermediate	51	21 7	22	23 2	19	21 3	10	19 6
	3 Expert	1	0 4	1	1 0	0	0 0	0	0 0
Frequency of Use of DOS	Total	233	100	94	100	88	100	51	100
	1 Never	109	46 8	32	34 0	49	55 7	28	54 9
	2 < Once/mth	66	28 3	31	33 0	22	25 0	13	25 5
	3 Once mth	20	8 6	10	10 6	8	9 1	2	3 9
	4 Few times/mth	23	9 9	12	12 8	5	5 7	6	11 8
	5 Few times/wk	10	4 3	5	5 3	3	3 4	2	3 9
	6 Once/dy	0	0 0	0	0 0	0	0 0	0	0 0
	7 Several times/dy	5	2 1	4	4 3	1	1 1	0	0 0

The results pertaining to general package experience are summarized in Tables 15-18 found in Appendix M, and Table 19, following. In terms of spreadsheets, of the 172 who use these, 161, or 93.6%, reported experience with Lotus. There was an even split among those reporting novice and intermediate experience, 49% each. Most, 54.1%, use a spreadsheet less than once per month. Only 11.2% use it a few times per week or more. There were 97, or 59.9%, who reported the spreadsheet they used as being a mixed command structure; 58, or 35.8% reported a menu structure; 5, or 3.1%, reported a command structure; and 2 reported using a mouse. The majority used the spreadsheet for school - 123, or 73.7%, and most learnt it in a course - 118, or 71.5%.

There were 155 who reported using a wordprocessor, and the most widely used one was Wordperfect, by 140, or 90.3%. Again, there was almost an even split of novice and intermediate experience levels reported - 116 novices, or 49.2%, and 101 intermediates, or 42.8%. Most, 83.3%, use it less than a few times per month. Most reported using a mixed command structure - 74, or 50.3%. There were 44, or 29.9%, who reported menu structures; 23, or 15.7%, reported command structures; and 6, or 4.1% reported a mouse. Again, the majority used it for school - 80, or 51.6%, followed by those who employ it for both school and personal use - 26, or 16.8%. There was an even split between learning it in a course and being self-taught, each about 45% of the sample.

Of the 61 using a database, DBase III+ and IV were the databases most frequently reported used - 58, or 95.2%. The majority were novices - 203 or 84.6%, and 88.7% used a database less than a few times per month. Most reported a mixed system - 28, or 49.1%; 17, or 29.8%, reported a menu system; 12, or 21.1%, reported a command system. Once again, not surprisingly, the majority - 47, or 77.1%, used it for school; 7, or 11.5%, for personal use; and only 5, or 8.2%, used it for work. Most learnt it in a course - 46, or 74.2%.

There were 122 who used programming languages, with Basic being the most often used, by 106, or 86.9%. There were 161, or 67.3%, who classified themselves as novices; 68, or 28.5%; as intermediates; and 10, or 4.2%, as experts. Most, 90.5%, used it less than once per month. The command structures which they reported they used with these programming languages were: Command - 47, or 44.8%; Mixed - 42, or 40.0%; Menu - 15, or 14.3%; Mouse - 1, or 0.9%. The purpose for use was mostly for school - 91, or 82%, and it was reported to be learnt mostly in a course - 92, or 83.6%.

Only 99 reported experience with statistical packages, of whom 96, or 97%, reported it to be with Minitab. (See Table 22). Of the 99, 53, or 53.5%, considered themselves intermediate; 42, or 42.5%, as novices; and 4, or 4%, as experts. It should be noted that the Novice category in the table includes those with no experience with any statistical packages. Frequency of use reported was mainly less than once per month, by 87.8%. The command structures reported were mainly Mixed - 40, or 44.5%; or Command - 38, or 42.2%. School was the purpose for which the packages were used in 91.6% of the cases, and they were learnt in a course 71% of the time.

Table 19
General Statistical Package Experience

		Total		Package Treatment					
				Merlin,		Minitab,		Extended Merlin,	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
Statistical Package	Total	99	100	42	100	37	100	20	100
	1- Minitab	96	97 0	41	97 6	35	94 6	20	100
	2- Merlin	0	0 0	0	0 0	0	0 0	0	0.0
	3- SPSS	2	2 0	0	0 0	2	5 4	0	0 0
	4- SAS	0	0 0	0	0 0	0	0 0	0	0 0
	5 Other	1	1 0	1	2 4	0	0 0	0	0 0
Statistical Package Experience	Total	240	100	96	100	93	100	51	100
	1- Novice	183	76 3	69	71 9	73	78 5	41	80 4
	2 Intermediate	53	22 1	23	24 0	20	21 5	10	19 6
	3- Expert	4	1 7	4	4 1	0	0 0	0	0 0
Frequency of Use of Statistical Package	Total	98	100	42	100	37	100	19	100
	1- Never	18	18 4	8	19 1	6	16 2	4	21.0
	2 < Once/mth	53	54 1	21	50 0	20	54 1	12	63 2
	3- Once/mth	15	15 3	4	9 5	9	24 3	2	10 5
	4- Few times/mth	7	7 1	5	11 9	1	2 7	1	5 3
	5- Few times/wk	4	4 1	3	7 1	1	2 7	0	0 0
	6 Once a day	1	1 0	1	2 4	0	0.0	0	0 0
	7- Several times/dy	0	0 0	0	0 0	0	0 0	0	0 0
Perceived Command Structure of Statistical Package	Total	90	100	38	100	37	100	15	100
	1- Menu	11	12 2	6	15 8	3	8 1	2	13 3
	2 Command	38	42 2	16	42 1	15	40 5	7	46 7
	3 Mixed	40	44 5	15	39 5	19	51 4	6	40 0
	4- Mouse	1	1 1	1	2 6	0	0 0	0	0 0
Purpose for which Statistical Package Used	Total	95	100	41	100	36	100	18	100
	1- Work	6	6 3	5	12 2	1	2 8	0	0 0
	2 School	87	91 6	35	85 4	34	94 4	18	100
	3- Personal	1	1 1	0	0 0	1	2 8	0	0 0
	4- Work/School	0	0 0	0	0 0	0	0 0	0	0 0
	5 School/Personal	1	1 0	1	2 4	0	0 0	0	0 0
	6- Work/Personal	0	0 0	0	0 0	0	0 0	0	0 0
	7- All of the above	0	0 0	0	0 0	0	0 0	0	0 0
Where Statistical Package Learnt	Total	93	100	39	100	36	100	18	100
	1- Course	66	71 0	25	64 1	29	80 6	12	66 7
	2- Self-taught	25	26 9	12	30 8	7	19 4	6	33 3
	3 Both	2	2 1	2	5 1	0	0 0	0	0 0

There is some evidence to suggest that users' anxiety with computer technology may have adverse effects on their performance. (Howard and Smith, 1986; Gilroy and Desai, 1986). We, therefore, asked a few questions related to this anxiety, which were borrowed from Raub's (1981) instrument and used in studies conducted by Igbaria (1990) and Howard (1986).

The responses to these questions about computer anxiety are summarized in Table 20 following. Using Cronbach's coefficient alpha (Cronbach, 1990), we found the reliability measure for the items to be .8457 which was in line with those reported in the studies mentioned above. The majority of the sample thought they could acquire computer skills, numbering 126, or 53%. There was an equal distribution - 65, or 27.3%, representing those who either Strongly Disagreed or Disagreed to Some Extent with the statement "I am unsure of my ability to learn a computer programming language". Those who were Uncertain made up 23.5%; those who Agreed to Some Extent made up 18.1%. To the statement "I will be able to keep up with important technological advances in computers", 88, or 37%, were Uncertain or Agreed to Some Extent, while 37, or 15.5%, Strongly Agreed. There were approximately even amounts who Strongly Disagreed, Disagreed to Some Extent, were Uncertain, or Agreed to Some Extent with "I feel apprehensive about using a computer terminal". The majority, 69.8%, Strongly Disagreed with the statement that they are afraid they might damage the computer. The majority, 46.6% and 43.5%, respectively, also reported that they did not avoid computers because they are unfamiliar, nor because they were afraid of making mistakes. Equal numbers Strongly Disagreed, Disagreed to Some Extent, were Uncertain, or Agreed to Some Extent with the statements "I am unsure of my ability to interpret a computer printout", "I have difficulty understanding most technological matters", and "Computer terminology sounds like confusing jargon to me".

Generally speaking, the subjects in the sample did not seem to be highly anxious with respect to computer usage.

Table 20
Responses to Anxiety Questions

		Total		Merlin,		Minitab,		Extended Merlin,	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
I am confident I could learn computer skills	Total	238	100	96	100	92	100	50	100
	1- Strongly disagree	1	0 4	0	0 0	1	1 1	0	0 0
	2- Disagree to some extent	1	0 4	1	1 1	0	0 0	0	0 0
	3- Uncertain	21	8 8	12	12 5	3	3 3	6	12 0
	4- Agree to some extent	89	37 4	32	33 3	38	41 3	19	38 0
	5- Strongly agree	126	53 0	51	53 1	50	54 3	25	50 0
I am unsure of my ability to learn a computer programming language	Total	238	100	96	100	92	100	50	100
	1- Strongly disagree	65	27 3	27	28 1	26	28 3	12	24 0
	2- Disagree to some extent	65	27 3	23	24 0	26	28 3	16	32 0
	3- Uncertain	56	23 5	26	27 1	19	20 6	11	22 0
	4- Agree to some extent	43	18 1	17	17 7	16	17 4	10	20 0
	5- Strongly agree	9	3 8	3	3 1	5	5 4	1	2 0
I will be able to keep up with important technological advances in computers	Total	238	100	96	100	92	100	50	100
	1- Strongly disagree	4	1 7	1	1 0	2	2 1	1	2 0
	2- Disagree to some extent	22	9 2	10	10 4	8	8 7	4	8 0
	3- Uncertain	88	37 0	30	31 3	38	41 3	20	40 0
	4- Agree to some extent	87	36 6	38	39 6	34	37 0	15	30 0
	5- Strongly agree	37	15 5	17	17 7	10	10 9	10	20 0
I feel apprehensive about using a computer terminal	Total	234	100	94	100	91	100	49	100
	1- Strongly disagree	62	26 4	28	29 8	23	25 2	11	22 5
	2- Disagree to some extent	50	21 4	16	17 0	21	23 1	13	26 5
	3- Uncertain	57	24 4	29	30 8	19	20 9	9	18 4
	4- Agree to some extent	48	20 5	15	16 0	20	22 0	13	26 5
	5 Strongly agree	17	7 3	6	6 4	8	8 8	3	6 1
If given the opportunity to use a computer, I am afraid I might damage it	Total	238	100	96	100	92	100	50	100
	1- Strongly disagree	166	69 8	64	66 7	69	75 0	33	66 0
	2- Disagree to some extent	33	13 9	13	13 5	13	14 1	7	14 0
	3- Uncertain	10	4 2	4	4 2	2	2 2	4	8 0
	4- Agree to some extent	22	9 2	11	11 4	5	5 4	6	12 0
	5- Strongly agree	7	2 9	4	4 2	3	3 3	0	0 0

Table 20
Responses to Anxiety Questions

		Total		Merlin,		Minitab,		Extended Merlin,	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
I have avoided	Total	238	100	96	100	92	100	50	100
computers	1- Strongly disagree	111	46 6	46	47 9	43	46 7	22	44 0
because they	2- Disagree to some extent	36	15 1	21	21 9	11	12 0	4	8 0
are un-familiar	3- Uncertain	21	8 8	4	4 2	12	13 0	5	10 0
to me	4- Agree to some extent	41	17 2	13	13 5	18	19 6	10	20 0
	5- Strongly agree	29	12 2	12	12 5	8	8 7	9	18 0
I hesitate to	Total	237	100	96	100	92	100	49	100
use a	1- Strongly disagree	103	43 5	41	42 7	42	45 7	20	40 8
computer for	2- Disagree to some extent	42	17 7	22	22 9	16	17 4	4	8 2
fear of making	3- Uncertain	26	11 0	14	14 6	6	6 5	6	12 2
mistakes I	4- Agree to some extent	44	18 6	9	9 4	23	25 0	12	24 5
cannot correct	5- Strongly agree	22	9 3	10	10 4	5	5 4	7	14 3
I am unsure of	Total	238	100	96	100	92	100	50	100
my ability to	1- Strongly disagree	61	25 6	27	28 1	23	25 0	11	22 0
interpret a	2- Disagree to some extent	61	25 6	25	26 0	27	29 3	9	18 0
computer	3- Uncertain	58	24 4	20	20 8	24	26 1	14	28 0
printout	4- Agree to some extent	49	20 6	21	21 9	17	18 5	11	22 0
	5- Strongly agree	9	3 8	3	3 1	1	1 1	5	10 0
I have	Total	236	100	95	100	91	100	50	100
difficulty	1- Strongly disagree	52	22 0	25	26 3	17	18 7	10	20 0
understand-ing	2- Disagree to some extent	71	30 1	32	33 7	27	29 6	12	24 0
most	3- Uncertain	52	22 0	17	17 9	23	25 3	12	24 0
technological	4- Agree to some extent	55	23 3	21	22 1	23	25 3	11	22 0
matters	5- Strongly agree	6	2 6	0	0 0	1	1 1	5	10 0
Computer	Total	238	100	96	100	92	100	50	100
terminology	1- Strongly disagree	50	21 0	23	24 0	18	19 6	9	18 0
sounds like	2- Disagree to some extent	63	26 5	23	24 0	26	28 3	14	28 0
confusing	3- Uncertain	41	17 2	17	17 7	16	17 4	8	16 0
jargon to me	4- Agree to some extent	73	30 7	30	31 3	28	30 4	15	30 0
	5- Strongly agree	11	4 6	3	3 1	4	4 3	4	8 0

5.B.2. Overall Performance and Perceived Comfort Results

In terms of performance, there was quite a bit of variability on all the measures. These are summarized in Table 21 following, and in the histograms in Figure 18. Once again, the histograms depict density so no scales are shown, however, the table shows the distributions. The explanations for the measures used are given in Chapter 4.B.8. The measures established were:

TIDX - score obtained for the sub-tasks completed*100/time to complete the tasks

EIDX - number of raw errors*100/scores obtained for the sub-tasks completed

HIDX - number of raw help calls*100/scores obtained for the sub-tasks completed

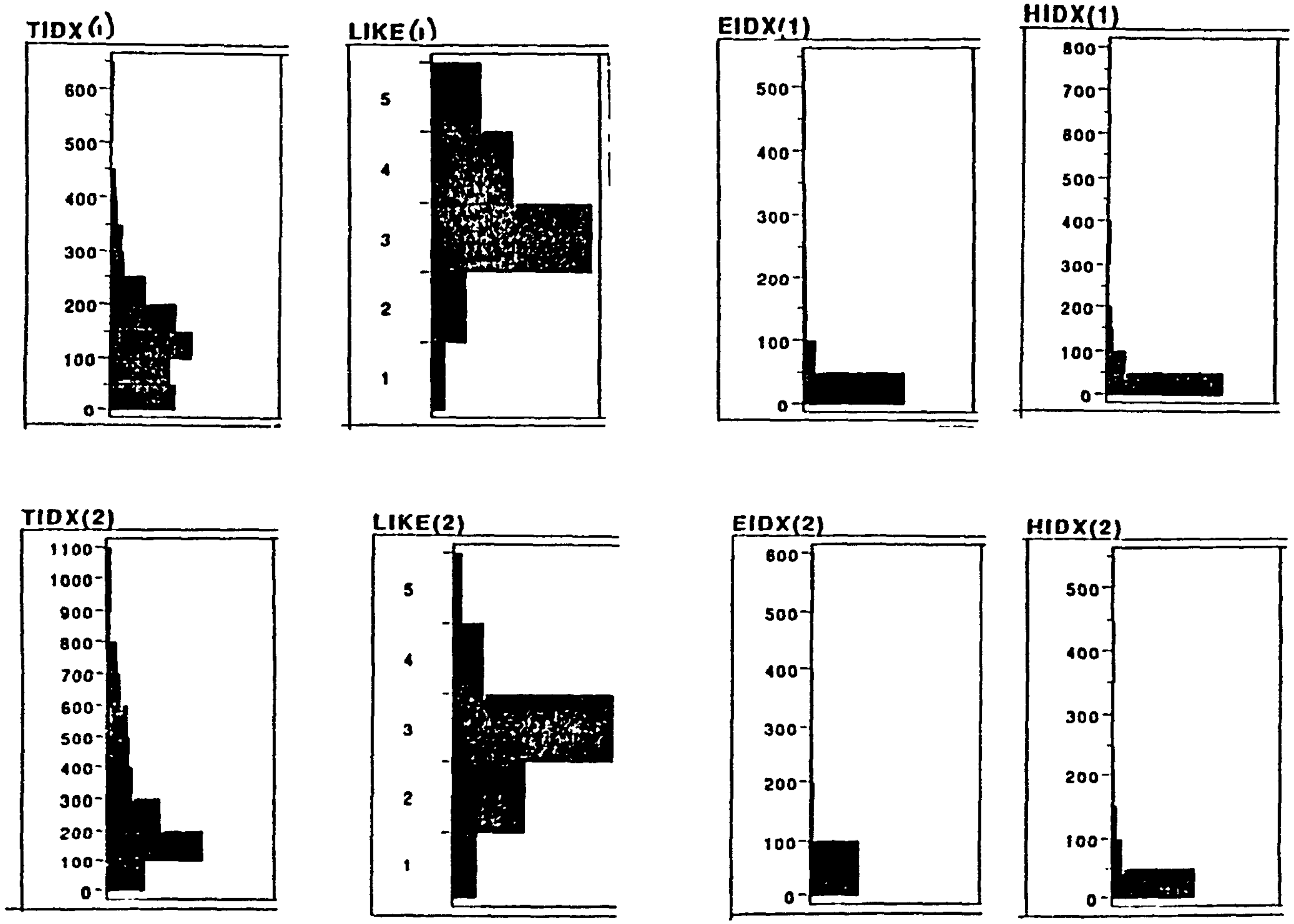
LIKE - 1-Very much; 2-Above average; 3-Average; 4-Below average; 5-Not at all

Table 21
Summary of Overall Distributions of Performance and Perceived Comfort Indices

Percent Distri-bution	TIDX(1)	TIDX(2)	EIDX(1)	EIDX(2)	HIDX(1)*	HIDX(2)*	LIKE(1)	LIKE(2)
N	227	128	208	124	162	79	232	117
Mean	143 7	300 8	42 8	21 6	68 6	24 3	3 3	2 7
sd	112 2	233 7	76 1	49 3	10 3	21 0	0 9	0 9
Lower25%	0-72	0-132	0-9	0-4	0-19	0-10	1 - 3 9%	1 - 8 5%
50%	72-192	132-236	9-41	4-21	19-62	10-33	2 - 10 3%	2 - 23 9%
15%	192-267	236-661	41-78	21-49	62-169	[illegible]	3 - 47 4%	3 - 54 7%
7 5%	267-444	661-971	78-328	49-128	169-496	58-88	4 - 23 7%	4 - 10 3%
Upper2 5%	444-645	971-1100	328-520	128-500	496-760	88-100	5 - 14.7%	5 - 2 6%

Legend * Excludes Extended Merlin
(1) Indicates Initial Use
(2) Indicates Retest
LIKE - 1 - Very much, 2 - Above Average, 3 - Average, 4 - Below Average, 5 - Not at all
Note A higher LIKE mean indicates a lower rating

Figure 18
Histograms of Distributions
of Overall Performance and
Perceived Comfort Scores



5.B.2.a. **Results - Performance (TIDX)**

Of the 243 subjects, there was data missing on 16 subjects. The mean Time Index for the remaining 227 was 143.7, with a standard deviation of 112.2. The lower 25% had a Time Index below 72, while at the high end, 10% had an index between 267 and 645. On the retest of 128 subjects, there was considerable improvement in this index. The mean was 300.8, but the standard deviation was still large, at 233.7. The lower 25% had an index less than 132, while the upper 10% now had an index between 661 and 1100.

5.B.2.b. **Results - Perceived Comfort (LIKE)**

In terms of perception of the packages, the mean rating given was 3.3 on initial use and 2.7 on the retest. On initial use, the majority - 47.4%, rated the packages as Average which increased to 54.7% on the retest. Those rating the packages Below average or Not at all made up 38.4% initially compared to 12.9% on the retest.

5.B.2.c. **Results - Error Analyses (EIDX)**

The mean of the Error Index was 42.8, with a standard deviation of 76.1. The lower 25% had an index less than 9, while the upper 10% had an index between 78 and 520. The retest resulted in a mean of 21.6 and a standard deviation of 49.3. The lower 25% had an index less than 4, and the upper 10% had an index between 49 and 500, about the same as on the initial use.

5.B.2.d. **Results - Help Analyses (HIDX)**

On the Help Index, the mean was 68.6, with a standard deviation of 10.3. The lower 25% had an Help Index less than 19, while the upper 10% had an index between 169 and 760. On subsequent use, the Help Index decreased to a mean of 24.3 with a standard deviation of 21.0. Twenty five percent had an index less than 10, while 10% had scores at the top of the range, of between 58 and 100, substantially below that found on initial use.

5.B.3. **<u>Results - Package and Experience Differences</u>**

It was postulated that differences in package Design and Assistance Features would have an impact on user performance with, and perceptions of, the packages. (Whiteside, et.al., 1985; Hauptmann and Green, 1983; Ogden, et.al., reported in Paap and Roske-Hofstrand, 1988; Carroll, 1985; Relles, 1979; Burns, et.al., 1986). We also noted in Study 1 that Command Structure and on- and offline Manuals were identified, by our expert panel, as being the most important Design and Assistant features compared to other features. We, therefore, assigned our subjects to three different package treatments. Treatment 1 - Merlin$_t$ has a mixed menu/command structure, with a concise online listing of commands and brief description. Treatment 2 - Minitab$_t$ has a command structure, with a full offline manual available online. Treatment 3 - Extended Merlin$_t$ is the original Merlin$_t$, with additional assistance from a Hypertext-based online index of commands, with explanations and examples. (See Appendix B for the examples of the package differences).

Although prior research has not been conclusive, a mixed menu/command structure with concise online help should conceptually produce better scores on measures relating to **ease of use** than a command structure with full online manual. One might expect Minitab$_t$ to outperform Merlin$_t$ on Speed, since it is usually faster to execute commands with a command structure than with an undirected menu structure, though this is dependent on the number of levels which have to be navigated in order to execute an operation. However, the originator of Merlin$_t$ designed the package with a directed menu structure and to maximize speed, so on that basis, we expected Merlin$_t$ to outperform Minitab$_t$ on the Speed dimension as well.

It also seems reasonable to assume that as experience is gained in the use of various packages, the facility with which new packages are learnt should be accelerated and enhanced. Much of the prior research which has examined the moderating effect of experience levels, has done so in terms of the expertise of the participants with the particular package being used in the research, or with packages of similar function and operation. (Karat, et.al., 1986; Ledgard, et.al., 1980; Gilfoil, 1982; Whiteside, et.al., 1985). Less attention has been given to the effect of experience gained with packages of dissimilar function and operation. To examine these possibilities, we differentiated our subjects based on their varying experience levels, Novice, Intermediate and Expert, with package types of similar and dissimilar function and operation. We examined the effect of experience with: no packages; various packages, other than statistical packages; various packages, which included one statistical package, the most cited being Minitab; various packages, which included several statistical packages, of which Minitab, was one. (See Chapter 4.B.4. for a full explanation of how these levels were established). Table 3 following, is reproduced from Chapter 4 and shows the level designations which were derived, and are to be used for reference throughout this section.

Table 3
Level Classification Scheme

Statistics Package Experience	Microcomputer Package Experience		
	Novice	Intermediate	Expert
L1 - No Pkgs	L1N	-	-
L2 - Varied, No Stats	L2N	L2I	L2E
L3 - Varied + 1 Stat	L3N	L3I	L3E
L4 - Varied + > 1 Stat	L4N	L4I	L4E

Legend
L1N - minimal, or no, computer or package experience
L2N - novice with various packages, not including statistics packages
L2I - intermediate with various packages, not including statistics packages
L2E - expert with various packages, not including statistics packages
L3N - novice with various packages, including one statistics package
L3I - intermediate with various packages, including one statistics package
L3E - expert with various packages, including one statistics package
L4N - novice with various packages, including more than one statistics package
L4I - intermediate with various packages, including more than one statistics package
L4E - expert with various packages, including more than one statistics package

The sample sizes in the following levels are small: L2E - 1; L3E - 0; L4N - 2; L4I - 3; L4E - 6; and therefore, their distributions cannot be considered representative. The sample sizes in the other levels were more reasonable: L1N - 25; L2N - 84; L2I - 36; L3N - 48; L3I - 22. As the sample sizes are small in some categories, we chose to look at the effect of statistical package experience separately from that of general microcomputer package experience. Tables 22-25, below, summarize the performance and perceived comfort results accordingly, that is, by package by experience. Discussion of these results follow the tables. Differences between the package treatments and between experience levels were assessed for statistical significance using the Tukey-Kramer Honestly Significant Differences comparison for all pairs. This method is suggested when the analyses are exploratory in nature and is a conservative measure when the sample sizes are unequal as is the case in this study. (Neter, Wasserman and Kutner, 1985).

Table 22
Performance (TIDX) by Package by Experience

	1-Merlin,			2-Minitab,			3-Extended Merlin,			Package Differences
	N	Mean	SD	N	Mean	SD	N	Mean	SD	
Statistics Package Experience										
L1-None, minimal	8	137 9	54 0	10	83 0	64 4	7	81 4	54 4	Mer > Min p < .10
L2-Varied, non Stat	50	154 4	130 7	47	88 3	68 1	24	126.4	68 1	Mer > Min p < 05
L3-Varied, + 1 Stat	27	178 8	110 0	24	160 0	72 3	19	162 5	95.8	N S
Subtotal L1, L2, L3	**85**	**160.6**	**118.9**	**81**	**108.9**	**75.8**	**50**	**133.8**	**81.5**	**Mer > Min p < .05**
L4-Varied, + > 1 Stat	10	332 2	208 2	1	140 0	-	-	-	-	N A
All	95	178 7	140 0	82	109 3	75 4	50	133 8	81 5	N A
Stat Pkg Differences	L4 > L3, L2, L1 p < 05			L3 > L2, L1 p < 05			L3 > L1 p < .05			
Microcomputer Package Experience										
Novice	61	158 4	118 3	57	102 2	62 2	41	129 8	84.2	Mer > Min p < 05
Intermediate	28	163 8	119 0	24	123 3	100 2	9	152 3	69 1	N S.
Expert	6	453 7	163 8	1	178 0	-	-	-	-	N A
All	95	178 7	140 0	82	109 3	75 4	50	133 8	81 5	N A
Microcomp Differences	Experts > Novices, Intermediates p < 05			N S			N.S			

Legend N A - not applicable
N S - no statistically significant differences

Table 23
Perceived Comfort (LIKE) by Package by Experience

	1-Merlin			2-Minitab			3-Extended Merlin			Package Differences
	N	Mean	SD	N	Mean	SD	N	Mean	SD	
Statistics Package Experience										
L1-None, minimal	7	3 6	1 0	10	3 9	0 7	6	3 7	1 0	N S
L2-Varied, non-Stat.	48	3 4	1 1	50	3.2	1 0	24	3 2	1 0	N S
L3-Varied, + 1 Stat	28	3 6	0 9	30	3 1	0 9	18	3 2	0 9	Min < Mer p < 10
Subtotal L1, L2, L3	**83**	**3.5**	**1.0**	**90**	**3.2**	**1.0**	**48**	**3.3**	**0.9**	**N.S.**
L4-Varied, + > 1 Stat	10	3 7	0 9	1	4 0	-	-	-	-	N A
All	93	3 5	1 0	91	3 2	1 0	48	3 3	0 9	N A
Stat Pkg Differences	N S.			L3 > L1 p < 10			N S			
Microcomputer Package Experience										
Novice	57	3 5	1 1	66	3 3	0 9	40	3 3	1 0	N S
Intermediate	30	3 6	1 0	24	3 2	1 0	8	3 0	0 5	N S
Expert	6	3 5	0 8	1	2 0	-	-		-	N A
All	*93	3 5	1 0	*91	3 2	1 0	*48	3 3	0 9	N A
Microcomp Differences	N S			N S			N S			

Legend N A - not applicable
N S - no statistically significant differences
LIKE rating 1- Very much, 2- Above average, 3- Average, 4- Below average, 5- Not at all

Table 24
Error Analyses (EIDX) by Package by Experience

	1-Merlin,			2-Minitab,			3-Extended Merlin,			Package Differences
	N	Mean	SD	N	Mean	SD	N	Mean	SD	
Statistics Package Experience										
L1-None, minimal	8	21 7	15 4	8	28 6	14 6	6	37 1	49 7	N.S.
L2-Varied, non-Stat	47	82 0	127 8	38	27 8	18 7	21	28 0	39 2	Mer > Min $p < .05$
L3-Varied, + 1 Stat	26	42 8	69 4	24	28 5	28.3	19	39 3	89.6	N.S.
Subtotal L1, L2, L3	**81**	**63.5**	**106.9**	**70**	**28.1**	**21.8**	**46**	**33.8**	**64.8**	**Mer > Min $p < .05$ Mer > Mer+ $p < .10$**
L4-Varied, + > 1 Stat	10	23 3	21 5	1	1 3	-	-	-	-	N.A
All	91	59 1	101 8	71	27 8	21 9	46	33 8	64 8	N A.
Stat Pkg Differences	N S			N S			N S			
Microcomputer Package Experience										
Novice	60	72 2	118 2	51	28 1	22 5	37	31 6	66 8	Mer > Min $p < 05$ Mer > Mer+ $p < 10$
Intermediate	25	39 5	55 3	19	27 7	21 0	9	43 2	58 5	N.S.
Expert	6	8 7	4 9	1	10 0	-	-	-	-	N A
All	*91	59 1	101 8	*71	27 8	21 9	*46	33 8	64 8	N A
Microcomp Differences	N S			N S			N S			

Legend N A - not applicable
N S - no statistically si ificant differences

Table 25
Help Analyses (HIDX) by Package by Experience

	1-Merlin,			2-Minitab,			Package Differences
	N	Mean	SD	N	Mean	SD	
Statistics Package Experience							
L1-None, minimal	8	29 3	21.2	8	41.6	23 0	N.S
L2-Varied, non-Stat.	47	111 0	180 3	38	65 1	42 2	N S
L3-Varied, + 1 Stat.	26	45 6	70 8	24	53 1	67 1	N S
Subtotal L1, L2, L3	**81**	**81.9**	**146.6**	**70**	**58.3**	**50.8**	**N.S.**
L4-Varied, + > 1 Stat	10	37 9	55 1	1	17.3	-	N A
All	91	77 1	140 0	71	57 7	50 6	N A
Stat Pkg Differences	N S			N.S			
Microcomputer Package Experience							
Novice	60	86.5	153 9	51	61 4	55 2	N S
Intermediate	25	69 6	118 3	19	48 5	37 0	N S
Expert	6	14 4	4 9	1	45 0	-	N A
All	91	77 1	140 0	71	57 7	50 6	N A
Microcomp Differences	N.S			N S			

Legend N.A - not applicable
N S - no statistically significant differences
Note It was not possible to track the help calls made in Extended Merlin.

5.B.3.a. **Results - Package Differences**

$Minitab_t$ had a lower average number of tasks completed than either of the other two treatments, that is, 3.2, with a mean score of 54.3. On $Merlin_t$, there were an average of 4.1 tasks completed, for a mean score of 64.6, while on the Extended $Merlin_t$, there were an average of 3.9 tasks completed, for a mean score of 65.8. The mean time for completion of the tasks on $Minitab_t$ was 54.1 minutes. This was higher than $Merlin_t$'s 43.1 minutes and the same as Extended $Merlin_t$'s 54.5 minutes. As the Subtotal row in Table 22 above indicates, $Merlin_t$'s mean TIDX, 160.6, was significantly better than $Minitab_t$'s, 108.9, but was not different from the Extended $Merlin_t$'s, 133.8. This was true for Novices, regardless of experience level with statistics packages, and L2's, those with experience with a variety of packages, not including statistics packages. There was no difference between the package treatments in the TIDX for L3's, those with experience with a variety of packages, including one statistics package, nor in the TIDX for Intermediates. This is surprising since the statistics package, with which the L3 individuals were already familiar, was $Minitab_t$, so those on $Minitab_t$ were revisiting $Minitab_t$ but those on $Merlin_t$ were seeing it for the first time. In fact, $Merlin_t$'s TIDX score, 178.8, was higher than $Minitab_t$'s, 160.0.

This better performance on $Merlin_t$ was earned in spite of worse Error Indices. (See Table 24, above). $Merlin_t$ had a significantly higher mean Error Index, 63.5, than both $Minitab_t$, at $p < .05$, and the Extended $Merlin_t$, at $p < .10$. This difference is the result of greater errors/score, on average, on $Merlin_t$ than on $Minitab_t$ experienced by those with Novice microcomputer experience and by L2's, those with experience with a variety of packages, not including statistics packages. There were no significant differences, from one package treatment to another, in the indices of L1's, minimal, or no, experience and L3's, those with experience with one statistics package. There were also no differences in the mean Help Indices between Merlin$_t$ and Minitab$_t$. The Extended Merlin$_t$'s scores could not be tracked. (See Table 25, above).

The better mean TIDX but lower EIDX did not seem to have an impact on perceived comfort. (See Table 23, above). There was no significant differences in the LIKE ratings between the package treatments overall, however, L3, those with experience with one statistics package, which was Minitab$_t$, gave a higher rating to Minitab$_t$ than to Merlin$_t$, at $p < .10$. Although the level of significance is low, it was what we would expect since these individuals who were on Minitab$_t$ already knew Minitab$_t$.

5.B.3.b. **Results - Experience Differences**

Looking at experience levels globally as defined in Table 3 reproduced above, there was not much difference in the number of tasks successfully accomplished in each of the levels, except for L4E, as expected. These subjects are the most experienced and completed almost all the tasks, an average of 5.7, to obtain an average score of 93.3. The other levels completed between 3.3 and 4.4 tasks. The differentiating factors were the types of tasks completed and the time it took to complete these tasks.

The lowest score on the tasks was obtained by L4I - 50.0. Although they completed the same average number of tasks as L4I, L1N members were able to complete sub-tasks worth higher scores to get an average score of 61.7, which was also higher than that obtained by L2N and L2I. L2N had an average score of 54.6, and L2I's score was 52.6. The one L2E individual had a score of 65.0, L3N got 69.5, L4N got 70.0, and L3I got 72.5.

L4E completed the tasks in the fastest time, in an average of 22.6 minutes. L1N, L2N, and L2I completed them in an average of 53.5, 53.0, and 51.0 minutes, respectively. The next fastest time was taken by L3N, in an average of 48.6 minutes. This was followed by L4I, with an average of 43.2 minutes, then L3I, in 42.5 minutes, and L4N, in 38.9 minutes. There was only one L2E individual, completing the tasks in 36.0 minutes.

Looking at experience separately according to statistics package experience and microcomputer package experience, we see that on the Merlin$_t$ package, only L4's, those with experience with a variety of packages, including more than one statistics package, had signficantly higher mean TIDX, 332.2, than the others, L1 - 137.9, L2 - 154.4, L3 - 178.8. (See Table 22, above). On Minitab$_t$ and Extended Merlin$_t$, L3, those with experience with only one statistics package, also outperformed L2's, those with experience with a variety of packages, not including statistics packages. The better performance of L3's on Minitab$_t$ was expected since these individuals already knew Minitab$_t$, however, it is interesting that on Extended Merlin$_t$, they also outperformed L1's, those with minimal, or no, package experience, which was not the case on Merlin$_t$. As there was no statistically significant difference between L3's on Minitab$_t$ and L3's on Merlin$_t$, we combined their results for all further analyses involving the L3 category.

Also as expected, Experts, 453.7, performed better than Novices, 158.4, and Intermediates, 163.8, on Merlin$_t$. There were no Experts on the Minitab$_t$ nor Extended Merlin$_t$ treatments. There were no differences in the mean TIDX between Novices and Intermediates on any of the treatments.

There were no differences in the Error nor Help Indices, regardless of statistics or microcomputer package experience. (See Tables 24 and 25, above).

In terms of perceived comfort, the only significant difference found was a better mean LIKE rating given to $Minitab_t$ by L3's compared to L1's, at $p < .10$. (See Table 23, above). Not surprisingly, having prior knowledge of $Minitab_t$ promoted a more favourable opinion to working on it again than having minimal, or no, prior experience on any package, but not compared to those with experience with a variety of packages, even if these did not include statistics packages.

5.B.4. <u>Retest Results - Package and Experience Differences</u>

Extending our investigation of the impact of experience on **ease of use**, we asked our volunteer subjects to redo the same task, on the same package, a week later. About half of the subjects took part. There was only one Expert and no individuals with experience with more than one statistics package.

5.B.4.a. Retest Results - Package Differences

Tables 26-29, below, compare the performance and perceived comfort results on the various package treatments, of only those who took part in the retest. It shows that each of the package treatments had substantially improved indices from initial use to retest a week later.

Table 26
Performance (TIDX) of those Retested by Package

	Initial			Retest		
	N	Mean	SD	N	Mean	SD
MERLIN	35	122 5	121.2	34	376 2	222 5
MINITAB	45	79 2	66 3	47	201 3	169 3
MERLIN +	46	133 7	83 9	47	345.9	264.9
ALL	126	111 1	93 1	128	300.8	233.7
	Minitab is significantly lower than Extended Merlin at p < 05 and lower than Merlin at p < .10			Minitab is significantly lower than Merlin and Extended Merlin at p < 05		
	Improvement on all treatments statistically significant at p < 05 Improvement on Minitab was significantly lower than Merlin at p < 05 and Extended Merlin at p < .10					

Table 27
Perceived Comfort (LIKE) - of those Retested by Package

LIKE	Initial				Retest			
	Total	Merlin,	Minitab,	Merlin,+	Total	Merlin,	Minitab,	Merlin,+
N	125	33	47	45	117	33	41	43
Mean	3 3	3 5	3 3	3 2	2 7	2 6	2 8	2 9
s d	1 0	1 1	1 0	1 0	0 9	0 9	0 8	0 9
1 - Very much %	4 8	6 0	6 4	2 2	8 5	15 2	7 3	4 7
2 - Above Average %	11 2	12 1	8 5	13.3	23.9	24 2	22.0	25 5
3 - Average %	44 8	27 3	44 7	57.8	54 7	48 5	58 5	55 8
4 - Below Average %	24 6	26 4	31 9	11.1	10.3	12.5	12.2	7.0
5 - Not at all %	13 6	18 2	8 5	15 6	2 6	-	-	7 0
	No statistically significant differences				No statistically significant differences			
	Improvements on Merlin and Minitab were significant at p < .05 and Extended Merlin at p < 10 Improvements between treatments were not statistically significant							
Note A higher LIKE mean indicates a lower rating								

Table 28
Error Scores (EIDX) of those Retested by Package

	Initial			Retest		
	N	Mean	SD	N	Mean	SD
MERLIN	32	104 0	146 6	34	10 3	11 6
MINITAB	35	28 3	18 7	45	26 0	26 7
MERLIN +	42	35 6	67 5	45	25 9	76 4
ALL	109	53 3	95 4	124	21 6	49 3
	Merlin had significantly higher EIDX than Minitab or Extended Merlin at p < 05			No statistically significant differences		
	Improvement on Merlin was significant at p < .05 Improvement on Minitab and Extended Merlin was not statistically significant Improvements on Merlin and Extended Merlin were significantly higher than on Minitab at p < 05					

Table 29
Help Scores (HIDX) of those Retested by Package

	Initial			Retest		
	N	Mean	SD	N	Mean	SD
MERLIN	32	139 8	195 6	34	13 6	9 9
MINITAB	35	63 9	37 8	45	32 4	23 6
ALL	67	100 2	142 0	79	24 3	21 0
	No statistically significant differences			Merlin had a significantly lower HIDX than Minitab at p < 05		
	Improvement on Merlin was significant at p < 05 Improvement on Minitab and Extended Merlin was not statistically significant Improvement on Merlin significantly greater than on Minitab at p < 05					

Of the 128 subjects who volunteered to be retested, those on $Minitab_t$ had a significantly lower mean Time Index, both initially and on the retest, than both the stand alone $Merlin_t$ at $p < .10$ and the Extended $Merlin_t$ at $p < .05$. The increase in performance from initial to subsequent use was also lower for $Minitab_t$ compared to $Merlin_t$ and to the Extended $Merlin_t$

While of those retested, those on $Merlin_t$ had a significantly higher mean error/score, 104.0, than either of the other treatments, 28.3 and 35.6, on the first try, this was reduced considerably on the retest, to 10.3, with the result that the differences between the treatments were no longer statistically significant. On the retest, both $Merlin_t$'s had a significantly greater improvement than $Minitab_t$, at $p < .05$.

Although $Merlin_t$'s mean Help Index was higher, 139.8, than $Minitab_t$'s, 63.9, the differences were not statistically significant. On the retest, however, $Merlin_t$'s index improves substantially to 13.6, thereby producing a significant difference at $p < .05$. In conjunction with the significantly greater improvement in $Merlin_t$'s Time and Error Indices, this suggests that considerably greater learning had taken place in the $Merlin_t$ users than $Minitab_t$ users.

Each of the package treatments was rated Average by the majority of the participants, initially and subsequently, and the differences between the treatments were not statistically significant. From initial use to the retest, the mean score on $Merlin_t$ went from 3.4 to 2.7, $Minitab_t$ from 3.3 to 2.8, and the Extended $Merlin_t$ from 3.2 to 2.9, but these improvements were not statistically significant between the treatments.

5.B.4.b. **Retest Results - Experience Differences**

Tables 30-33, below, summarize the performance and perceived comfort results of only those retested according to statistics package and microcomputer package experience. Their discussion follows the tables.

Table 30
Performance (TIDX) by Experience of those retested

	Initial			Retest		
	N	Mean	SD	N	Mean	SD
Statistics Package Experience						
L1-None, minimal	21	80 9	45 1	21	170 9	138 6
L2 Varied, non Stat	89	108 8	96 4	89	321 8	234 5
L3 Varied, + 1 Stat	16	163 8	103 5	18	348 9	274 0
L4-Varied, + > 1 Stat	-	-	-	0	-	-
All	126	111 1	93 1	128	300 8	233 7
	L3 greater than L2, L1 at p < 05			L3 greater than L1 at p < 05		
	All improved significantly at p < 05 L2 had a greater improvement than L1 at p < 05					
Microcomputer Package Experience						
Novice	93	109 6	93 7	95	288 8	224 4
Intermediate	32	113 3	93 4	32	317 8	242 1
Expert	1	178 0	-	1	581 0	-
All	126	111 1	93 1	128	300 8	233 7
	No statistically significant differences			No statistically significant differences		
	All improved significantly at p < 05 Differences in improvements not statistically significant					

Table 31
Perceived Comfort (LIKE) by Experience
of those retested

	Stat. Pkg. Experience					Micro Pkg. Experience			
	Average	L1	L2	L3	L4	Average	Novice	Intermediate	Expert
Initial									
N	125	20	87	18	0	125	93	31	1
Mean	3 3	3 9	3 2	3 2	-	3 3	3 4	3 3	2 0
s d	1 0	0 9	1 0	0 9	-	1 0	1 0	1 0	-
1- Very much %	4 8	-	6 9	-	-	4 8	4 3	6 5	
2- Above average %	11 2	5 0	11 5	16 7	-	11 2	11 8	6 5	100 0
3- Average %	44 8	30 0	46 0	55 5	-	44 8	43 0	51 6	-
4- Below average %	25 6	40 0	24 1	16 7	-	25 6	25 8	25 8	-
5- Not at all %	13 6	25 0	11 5	11 1	-	13 6	15 1	9 6	-
*Sum of sample sizes	L2 gave a better mean rating than L1 at p < 05					No statistically significant differences			
Retest									
N	117	17	83	17	0	117	90	32	1
Mean	2 7	3 1	2 7	2 6	-	2 7	2 7	3 1	1 0
s d	0 9	0 9	0 9	0 6	-	0 9	0 8	0 9	
1- Very much	8 5%	5 9	9 6	5 9		8 5%	6 7	11 5	
2- Above average	23 9	5 9	26 5	29 4	-	23 9	28 9	7 7	
3- Average	54 7	64 7	50 6	64 7		54 7	52 2	65 4	100 0
4- Below average	10 3	17 6	10 9	-	-	10 3	10 0	11 5	
5- Not at all	2 6	5 9	2 4	-	-	2 6	2 2	3 9	
	No statistically significant differences					No statistically significant differences			
	All improved significantly at p < 05 Differences in improvements not statistically significant					Improvement in Novices significant at p < 05 Improvement in Intermediates not statistically significant Difference in improvements not statistically significant			

Table 32 Error Analyses (EIDX) by Experience of those retested						
	Initial			Retest		
	N	Mean	SD	N	Mean	SD
Statistics Package Experience						
L1-None, minimal	18	30 3	29 8	18	27.7	41.0
L2-Varied, non-Stat	75	60 7	104 7	74	43 0	93 5
L3-Varied, + 1 Stat	16	44 9	97 1	16	20 4	33 7
L4-Varied, + > 1 Stat	-	-	-	0	-	-
All	109	53 3	95 4	108	37 1	80.4
	No statistically significant differences			No statistically significant differences		
	Improvement in L2 significant at p < 05 Improvement in L1 and L2 not statistically significant Differences in improvements not statistically significant					
Microcomputer Package Experience						
Novice	83	55 9	103 9	81	41 5	91 2
Intermediate	25	46 5	62 7	26	24 5	28 4
Expert	1	10 0	-	1	1 0	-
All	109	53 3	95 4	108	37 1	80 4
	No statistically significant differences			No statistically significant differences		
	All improved significantly at p < 05 Differences in improvements not statistically significant					

Table 33
Help Analyses (HIDX) by Experience
of those retested

	Initial			Retest		
	N	Mean	SD	N	Mean	SD
Statistics Package Experience						
L1-None, minimal	12	40 9	22.3	13	24 0	14 3
L2-Varied, non-Stat	54	113 8	155 0	63	21 7	20 3
L3-Varied, + 1 Stat	1	76 0	-	3	38 7	53 7
L4-Varied, + > 1 Stat	0	-	-	0	-	
All	67	100 2	142 0	79	24 3	21 0
	No statistically significant differences			No statistically significant differences		
	Improvement in L1 and L2 significant at p < 05 Improvement in L3 not significantly different Differences in improvement not statistically significant					
Microcomputer Package Experience						
Novice	50	104 3	145 3	56	27 1	23 4
Intermediate	16	90 5	139 4	22	18 1	10 7
Expert	1	45 0	-	1	2 0	
All	67	100 2	142 0	79	24 3	21 0
	No statistically significant differences			No statistically significant differences		
	All improved significantly at p < 05 Differences in improvement not statistically significant					

All the levels of statistics and microcomputer package experience taking part in the retest improved considerably from initial to subsequent use a week later, at $p < .05$, however, only L2 - those with experience with packages other than statistics packages had a greater improvement, statistically, over L1 - minimal, or no, experience with any package, at $p < .05$. (See Table 30, above). It would also seem that experience even with packages other than statistics packages facilitates retention, since, on the retest, this category, L2, is no longer statistically different from L3, who have experience with 1 statistics package, but is still significantly different from L1 who have minimal, or no, experience. On the retest, there were no L4's - experience with more than one statistics package, and only one Expert, whose Index improved from 178.0 to 581.0. There was no statistical difference in the improvements made between Novices and Intermediates.

On average, all experience levels had improved Error Indices, that is, lower indices, however, only L2's improvement was statistically significant. There were no statistically significant differences in improvements between statistics package experience levels or microcomputer package experience levels. (See Table 32, above).

As Table 33 above indicates, the number of help calls/score decreased considerably on the retest, but, again, the differences in improvements between levels were not statistically significant.

The percentage of those rating the packages as Average and better increased on the retest for all experience levels, at $p < .05$, the most notable being L1 - minimal, or no, experience, from 35.0% to 87.1%, and Novices, from 59.1% to 87.8%. The percentage of

L2's went from 64.4% to 86.7% and L3's went from 72.2% to 100.0%. Intermediates went from 64.6% to 84.6%. There was only one Expert on the retest. These improvements were not statistically different between the experience levels, as shown above in Table 31.

5.B.5. Results - Propositions

In Study 2, three major research questions were posed, around which a number of propositions were developed. The questions revolved around package design differences, experience differences and other factors. They were:

Research Question 2:

2.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, as a consequence of differing Design and Assistance Features?

Research Question 3:

3.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to differing Experience levels?

Research Question 4:

4.1 Is there a difference in perceived comfort and performance measures relating to **ease of use**, attributable to other factors, such as computer anxiety, gender, quantitative competence?

In the previous section, we used the Tukey-Kramer comparison for all pairs as it is a more conservative test which is appropriate for exploration of the data. In this section, however, as we are evaluating particular comparisons between treatments, the Student's t pair-wise comparison is used to assess statistically significant differences. (Neter, Wasserman and Kutner, 1985).

5.B.5.a. **Results - Research Question 2 - Propositions 2.a.-2.g.**

Notwithstanding the lack of consistent results in the research done to date on the advisability of different command structures and help aids, we suggested that $Merlin_t$ would produce better performance and perceived comfort measures than $Minitab_t$ because of its mixed menu/command structure and concise online help, which were expected to give greater support to the le-rning dimensions of **memory** and **effort**. This, in turn, should assist the learning and execution of tasks with the package, that is, affect performance **speed**, and affect the perceived **comfort** of the user. The Extended Hypertext-based $Merlin_t$ help system, with examples and expanded explanations, was expected to further assist these measures relating to **ease of use**. The assistance to memory provided by both of the $Merlin_t$ systems should also facilitate retention.

Propositions - Initial Use

Table 34, below, is to be used as reference for the propositions which follow.

Table 34
Performance and Perceived Comfort by Package

	$Merlin_t$		$Minitab_t$		Extended $Merlin_t$		Differences
	N	Mean	N	Mean	N	Mean	
TIDX	85	118 9	81	108 9	50	133 8	Mer > Min, p < 05
LIKE	83	3 5	90	3 2	48	3 3	N S
EIDX	81	63 5	70	28 1	46	33 8	Mer > Min, p < 05 Mer > Mer+ p < 10
HIDX*	81	81 9	70	58 3	N A	N A	N S

*Excludes Extended $Merlin_t$
N A - Not applicable
N S - Not statistically significant

The first proposition was stated as:

Proposition 2.a. - Merlin$_t$ will produce better performance measures than Minitab$_t$.

The Time Index (TIDX) was found to be significantly lower, that is, worse, on Treatment 2 - Minitab$_t$ than on Treatment 1 - Merlin$_t$, at the $p < .05$ level. There was no difference between the Help Indices (HIDX), but Merlin$_t$'s Error Index (EIDX) was significantly higher than Minitab$_t$'s, at $p < .05$.

The better results of Merlin$_t$'s Time Index over Minitab$_t$'s, even with a higher Error Index and the same Help Index, indicate strong support for Proposition 2.a.

The second proposition was stated as:

Proposition 2.b. - Merlin$_t$ will be given better perceived comfort ratings than Minitab$_t$.

There were no statistically significant differences between the mean ratings of the package treatments. Proposition 2.b. was not supported.

The third proposition was stated as:

Proposition 2.c. - Users of the Extended Hypertext-based Merlin$_t$ will have better performance measures than either the stand-alone Merlin$_t$ or Minitab$_t$, and will have better perceived comfort ratings than either of them.

There was no statistically significant differences in the Time Indices between the Extended Merlin$_t$ and Minitab$_t$ treatments, nor the Merlin$_t$ stand alone, though the Extended Merlin$_t$'s Index was higher than both. There were no differences in the Error Indices between the

Extended Merlin$_t$ and Minitab$_t$, but Minitab$_t$ was significantly lower than the Merlin$_t$ stand alone. The Help Index was not useful here since this could not be tracked in the Extended Merlin$_t$.

There was no statistically significant difference in the 'liking' rating for the Extended Merlin$_t$ versus either Minitab$_t$ or Merlin$_t$. While the means were comparable, being 3.5, 3.2, and 3.3, respectively for Merlin$_t$, Minitab$_t$, and Extended Merlin$_t$, a greater percentage did rate the Extended Merlin$_t$ higher, 133.8, than the other two, 118.9 and 108.9. The Extended Merlin$_t$ was rated Average and above by 72.9%, compared to Merlin$_t$'s 52.7% and Minitab$_t$'s 64.8%.

It would seem that the addition of the online Hypertext-based Index did nothing to improve the performance results over either Merlin$_t$ or Minitab$_t$. Also, there were no statistically significant differences in perceived comfort between the treatments, though a higher percentage rated the Extended Merlin$_t$ Average and above compared to the other two treatments. Though the raw data gives some indication for support of Proposition 2.c., statistically speaking it cannot be considered to be supported.

Propositions - Retest

Merlin$_t$'s mixed menu/command structure and concise online help system were expected to impose less strain on users' memory requirements than the command structure of Minitab$_t$ with its full online manual. Subsequent use should, therefore, permit users to accelerate their performance time and reduce the errors and help calls made. This should, in turn, lead to a greater improvement in the liking for the package. In fact, each of the packages should have improved 'liking' ratings because of our assumption that familiarity, in this case, does not breed contempt, but rather tolerance. Table 35, below, can be used for reference for the propositions which follow.

Table 35
Performance and Perceived Comfort by Package of those retested

		Initial		Retest	
		N	Mean	N	Mean
TIDX	Mer	35	122 5	34	376 2
	Min	45	79 2	47	201 3
	Mer+	46	133 7	47	345 8
		Min < Mer, Mer+ at p < 05		Min < Mer, Mer+ at p < 05	
		Mer > improvement than Min at p < 05 Mer+ > improvement than Min at p < 10			
LIKE	Mer	33	3 5	33	2 6
	Min	47	3 3	41	2 8
	Mer+	45	3 2	43	2 9
		No statistically significant differences		No statistically significant differences	
		Differences in improvements not statistically significant			
EIDX	Mer	32	104 0	34	10 3
	Min	35	28 3	45	26 0
	Mer+	42	35 6	45	25 9
		Mer > Min, Mer+, p < 05		No statistically significant differences	
		Mer, Mer+ > improvement than Min at p < 05			
HIDX*	Mer	32	139 8	34	13 6
	Min	35	63 9	45	12 4
		Mer > Min, p < 05		Mer < Min, p < 05	
		Mer > improvement than Min at p < 05			

*Excludes Extended Merlin

The fourth proposition was stated as:

Proposition 2.d. - Users of Merlin$_t$ will show a greater improvement in performance, on subsequent use of the package, than will Minitab$_t$ users.

There were significant improvements in the Time Indices, on all the packages on subsequent use, at $p < .05$. The Merlin$_t$ stand alone was, however, found to have a statistically greater improvement, at $p < .05$, and Extended Merlin$_t$, at $p < .10$, than Minitab$_t$. Both Merlin$_t$'s had a statistically greater improvement in errors/score than did Minitab$_t$, at $p < .05$, as was Merlin$_t$'s improvement in help calls/score over Minitab$_t$'s. One reason for Merlin$_t$'s substantial improvement compared to Minitab$_t$'s may be thought to be the fact that L3 individuals on Merlin$_t$ had not previously used Merlin$_t$ whereas L3 individuals on Minitab$_t$ had previously used Minitab$_t$, therefore, there was more room for improvement on Merlin$_t$ than on Minitab$_t$. As we see in Table 41 following, however, there was no statistically significant difference in the performance measures on these packages between these individuals. Add to this the fact that Merlin$_t$'s performance was nevertheless superior to Minitab$_t$'s, except for error/score, then it would seem that the superior initial and retest results are due to Merlin$_t$'s package design.

Proposition 2.d. was supported. Merlin$_t$ showed greater improvement on all performance indices compared to Minitab$_t$.

The fifth proposition was stated as:

Proposition 2.e. - There will be a general improvement in the perceived comfort rating for each of the packages, on subsequent use.

The perceived comfort of each of the packages improved on subsequent use, at $p = .05$.

Merlin$_t$'s mean 'liking' rating went from 3.5 to 2.6, Minitab$_t$'s from 3.3 to 2.8, and Extended Merlin$_t$'s from 3.2 to 2.9. The percentage of those rating Merlin$_t$ Average or above rose from 45.4% to 87.9, for Minitab$_t$ it rose from 59.6% to 87.8%, and for Extended Merlin$_t$ it went from 73.3% to 86.0%. Support for Proposition 2.e. was found. There were no differences in the changes in the ratings, between the packages.

Propositions - Learning Dimensions

One of the premises put forward in this thesis, is the belief in the importance of support offered by the various package features for certain learning dimensions which we identified as Speed, Memory, Effort and Comfort. It follows, then, that if users identify these dimensions as being supported by the package treatment to which they were assigned, they should express a better perception of the package than if support is not perceived to be offered. We also suggested that these learning dimensions are interdependent, that is, when memory and effort are alleviated, peace of mind and psychological comfort are promoted, which, in turn, will contribute to a reduction in performance time. This interdependence is seen in our measure of performance time, as discussed in Chapter 4.B.8., which includes time to make and correct errors and to read help screens. It does not account for the Comfort factor. If this interdependence holds, it is reasonable to assume that the greater the number of attributes or features in the package that are perceived to support these dimensions, the better will be the performance and perceived comfort with the package.

The sixth proposition was stated as:

Proposition 2.f. - Perceived comfort, or 'liking' ratings and performance measures will be better when the dimensions, Speed, Memory, Effort and Comfort, are perceived to be supported by more attributes in the packages than when less are perceived to be supported.

Table 36, below, shows that there is a statistically significant positive correlation between both perceived comfort and the performance TIDX measures and the number of attributes in the packages identified as supporting Speed, Memory, Effort and Comfort. Although the correlation between these dimensions and the performance index is low, the relationship is quite strong with respect to the 'liking' rating. Speed and Comfort have the strongest relationships with 'liking', with coefficients of 0.566 and 0.642, respectively. The associated coefficients for Memory and Effort are 0.376 and 0.487, respectively. Proposition 2.f., therefore, cannot be rejected.

Table 36
Performance and Perceived Comfort by Learning Dimensions

Learning Dimensions	TIDX		LIKE	
	Correlation Coefficient	p-value	Correlation Coefficient	p-value
Speed	+0 292	000	+0 566	000
Memory	+0 174	045	+0 376	000
Effort	+0 224	001	+0 487	000
Comfort	+0 214	019	+0 642	000

LIKE rating 1- Very much, 2- Above average, 3- Average, 4- Below average, 5- Not at all

The seventh proposition was stated as:

<u>Proposition 2.g.</u> - Better perceived comfort, or 'liking' ratings, will result in <u>better</u> performance indices.

Table 37, below, compares the average performance indices with the 'liking' ratings given by our sample. It indicates quite clearly a positive relationship between the ratings given to the packages and the performance scores obtained. Better 'liking' ratings were associated with better performance indices, that is, higher Time Indices and lower Error and Help Indices. Statistically significant differences were found in the Time Indices between those rating the packages 'Very much' and all other ratings, at $p < .05$. The performance of those rating the packages 'Above average' and 'Average' was also significantly higher than those rating them 'Below Average'. Those rating the packages 'Not at all' or 'Below average' had higher Error and Help Indices than those rating them 'Average' or better. We may say that Proposition 2.g. is supported.

Table 37
Performance versus Perceived Comfort for all packages

LIKE	N	TIDX	N	EIDX	N	HIDX
1- Very much	9	221 0	8	9 7	7	20 8
2- Above average	22	143 8	20	16 7	14	31 2
3- Average	100	150 2	95	36 5	69	53 1
4- Below average	47	99 6	42	57 4	37	[illegible]5 1
5- Not al all	28	95 6	23	78 7	18	140 6
	1 < all, p < 05, 4,5 < 3, p < .05 4,5 < 2, p < .10		1,2,3 < 5, p < 05, 2 < 4, p < 05 1 < 4, p < 10		1,2,3 < 5, p < 05, 1,2,3 < 4, p < 10	

5.B.5.b. **Results - Research Question 3 - Propositions 3.a.-3.i.**

It seems reasonable to suggest that as experience is gained with a package, performance and perceived comfort should improve. It also seems reasonable to assume that prior experience with one package of a similar type should cause some difficulty in adapting to another, both in terms of performance and perceived comfort. On the other hand, varied experience with package should facilitate operation of a new package, even if perceptions are negatively affected. Since there was no statistically significant differences found, as discussed in Section 5.B.2, between L3's on Minitab$_t$ and L3's on Merlin$_v$, their results were henceforth combined for all analyses. Once again, we use the Student's t pair-wise comparisons to evaluate our findings.

Propositions - Initial Use

The first proposition was stated as:

<u>Proposition 3.a.</u> - L3 users of various packages, including only one statistical package, will <u>perform worse on Merlin$_v$</u>, <u>and give it lower perceived comfort ratings</u>, than L2 users, who have had experience with various packages, not including statistics packages.

Believing that exposure to only one package, of similar function, may tend to make the user become entrenched in that package and, therefore, resistant to a new package of similar function, we expected L3 subjects, most of whom had had exposure to Minitab$_t$ but not to Merlin$_v$, to perform worse on Merlin$_v$, and to give it a lower rating, than L2 subjects, who have had experience with a variety of packages. This did not prove to be the case. There was no statistically significant difference between L2's TIDX, 154.4 and L3's, 178.8, of those on the Merlin$_t$ package. There were also no significant differences in the EIDX, HIDX or LIKE measures. (See Table 38, below). Proposition 3.a. was, therefore, not supported.

Table 38 Comparison of L2 and L3 on Merlin,					
	L2		L3		L3 vs. L2 on Merlin,
	N	Mean	N	Mean	
TIDX	50	154 4	27	178 8	N S.
LIKE	48	3 4	28	3 6	N.S.
EIDX	47	82 0	26	42.8	N.S.
HIDX*	47	111.0	26	45.6	N.S.

Legend N S - not significant
L2 - experience with various packages, not including statistics packages
L3 - experience with various packages, including one statistics package
*Excludes Extended Merlin,

The second proposition was stated as:

Proposition 3.b. - L2 - experience with various packages, not including any statistics packages, and L3 - experience with various packages, but only one statistics package, will outperform L1 - minimal, or no experience, but give lower perceived comfort ratings, regardless of package treatment.

It seems reasonable to expect that those who have had experience with packages will have better performance than those who have had none. It also seems reasonable to assume that they will give a lower rating to an unfamiliar package than would complete novices who have had no prior experience of any kind, and so have no basis for formulating prior expectations which are likely to influence perceptions. There were no differences in performance or perceived comfort between L1's and L2's. However, L3s Time Index was significantly better than L1's, at $p < .05$, as hypothesized. Contrary to what was hypothesized, however, L2 and L3 rated the packages higher than L1. L2 rated them higher than L1, at $p < .05$, and L3 rated them higher, at $p < .10$. L2 also had a higher HIDX than L1, at $P < .10$. (See Table 39, below). Proposition 3.b. was partially supported.

Table 39
Comparison of L2 and L3 with L1 regardless of package

	L1		L2		L3		L3 vs. L1	L2 vs. L1
	N	Mean	N	Mean	N	Mean		
TIDX	25	100.1	121	123.2	70	167.9	N S	p < .05
LIKE	23	3.7	122	3.3	76	3.3	p < .05	p < .10
EIDX	22	28.4	106	51.9	69	36.8	N S	N S
HIDX*	16	35.4	85	90.5	50	49.2	p < .10	N S

Legend N S - no statistically significant differences
L1 - minimal, or no, computer or package experience
L2 - experience with various packages, not including statistics packages
L3 - experience with various packages, including one statistics package
*Excluding Extended Merlin$_t$

The third proposition was stated as:

Proposition 3.c. - L2 users with various packages, not including statistics packages, will perform better on Merlin$_t$ than on Minitab$_{t'}$, and give it better perceived comfort ratings.

Based on the fact that Merlin$_{t'}$, with its mixed command structure and concise online help, outperformed Minitab$_t$ in terms of the Time Index, it was assumed that, as both the Minitab$_t$ and Merlin$_t$ packages were new to L2 subjects, who have had experience with packages other than statistical packages, those using Merlin$_t$ should perform better, and rate it higher, than those using Minitab$_t$. This did prove to be the case for performance but not for perceived comfort. (See Table 40, below). Merlin$_t$ had a higher mean TIDX, 154.4, compared to Minitab$_{t'}$, 88.3, at $p < .05$, but there was no statistical difference in the liking rating. This better performance was earned at the cost of a higher EIDX, 82.0 compared to Minitab$_{t'}$, 27.8, at $p < .05$. There was no difference in the Help Indices. Proposition 3.b. was, therefore, partially supported for L22, and L23 is still to be tested, there being only one subject in L23.

Table 40 Comparison of L2 on Merlin, vs. Minitab,									
	Merlin,		Minitab,		Extended Merlin,		Mer vs Min	Mer vs Mer+	Mer+ vs. Min
	N	Mean	N	Mean	N	Mean			
TIDX	50	154.4	47	88.3	24	126 4	p < 05	N.S.	N S.
LIKE	48	3.4	50	3 2	24	3.2	N S	N.S	N S.
EIDX	47	82 0	38	27.8	21	28 0	p < .05	N.S.	N.S.
HIDX*	47	111.0	38	65 1	-	N.A.	N.S.	N S.	N.S

Legend N.S - not significant
N.A - not applicable
L2 - experience with various packages, not including statistics packages
*Excludes Extended Merlin,

The fourth proposition was stated as:

Proposition 3.d. - L3 users with one statistics packages, which was Minitab,, as well as with other packages, will give lower perceived comfort ratings to Merlin, than to Minitab, and perform better.

Familiarity with only one statistics package was expected to cause resistance to a new package and, therefore, adversely impact liking for the new package. Table 41, below, indeed, shows this to be true. The mean Liking rating on Minitab, was 3.2, compared to 3.6 on Merlin,. Minitab, was rated Above Average and better, by 16.7%, compared to only 3.9% for Merlin,. Those rating Minitab, as Average and below made up 83.3%, while the comparable figure for Merlin, was 96.1%. The differences were found to be statistically significant at $p < .10$. Since L3 had prior experience with one statistics package, which was Minitab,, we had expected that these individuals using Minitab,, should outperform Merlin,.

This was not the case. The differences were not statistically significant. There were also no differences in errors or help. Proposition 3.d. was supported for perceived comfort but not for performance.

Table 41
Comparison of L3
on $Merlin_t$ vs. $Minitab_t$

	$Merlin_t$		$Minitab_t$		Extended $Merlin_t$		Differences
	N	Mean	N	Mean	N	Mean	
TIDX	27	178 8	24	160 0	19	162 5	N S
LIKF	28	3 6	30	3 1	18	3 2	Min > Mer p < 10
EIDX	26	42 8	24	28 5	19	39 3	N S
HIDX	26	45 6	24	53 1	-	N A	N S

The fifth proposition was stated as:

Proposition 3.e. - L4 users of various packages, including more than one statistics package, of which $Minitab_t$ is one, will outperform all other levels, on $Merlin_t$.

We made the assumption that those with experience on a variety of statistics and other packages would have more flexibility in adapting to new packages. The analyses indicate that, at $p < .05$, L4, those expert with several packages including many statistics packages, outperformed all other levels on the Time Index. (See Table 42, below). There were no significant differences between any of the levels on the Error or Help Indices nor on the Liking measure. Proposition 3.e. was, therefore, supported.

Table 42 Comparison of L4 with all other levels, on Merlin,									
	L1		L2		L3		L4		L4 vs. all other levels
	N	Mean	N	Mean	N	Mean	N	Mean	
TIDX	8	137 9	50	154 4	27	178 8	10	332.2	p < 05
LIKE	7	3 6	48	3 4	28	3 6	10	3 7	N S
EIDX	8	21 7	47	82 0	26	42 8	10	23 3	N.S
HIDX*	8	29 3	47	111 0	26	45 6	10	37 9	N S

Legend N S - no statistically significant differences
L1 - minimal, or no, experience
L2 - experience with several packages, excluding statistics packages
L3 - experience with several packages, including one statistics package
L4 - experience with several packages, including more than one statistics packages
*Excludes Extended Merlin,

The sixth proposition was stated as:

<u>Proposition 3.f.</u> - Novices, regardless of level, will have the <u>same level of perceived comfort</u>, on the same package.

With minimal or no exposure to packages, it can be assumed that novices in all level categories, working on the same package, should rate the packages the same. These users should not, as yet, be entrenched in a package and so be resistant to a new one. This was the case on the Merlin$_t$ and Extended Merlin$_t$ treatments, but not on Minitab$_t$. There were no significant differences between any of the novice groups on either of the Merlin$_t$'s. On Minitab$_t$, however, both L2N and L3N gave ratings significantly better than L1N, at $p < .05$. It seems experience of any kind, even at a novice level, promotes a greater sense of comfort on a command structure package than having no experience with packages, which was not seen on the mixed menu/command structure package. (See Table 43, below). There was no evidence to refute Proposition 3.f. for Merlin$_t$ nor Extended Merlin$_{ι}$, but it was not supported for Minitab$_t$. There was only one subject each on Merlin$_t$ and on Minitab$_t$ in L41, and there were none on the Extended Merlin$_t$.

Table 43
Comparison of all Novice Levels within each Package

		Merlin$_t$			Minitab$_t$			Extended Merlin$_t$		
		N	Mean	Differences	N	Mean	Differences	N	Mean	Differences
LIKE	L1N	7	3 6		10	3 9	L2N, L3N > L1N	6	3 7	
	L2N	32	3 3	N S	34	3 2	$p < .05$	19	3 2	N S
	L3N	17	3 7		21	3 0		15	3 3	
	L4N	1	3 0		1	4 0		-	-	

Legend p - significance probability level
N S - not significant
L1N - minimal, or no, computer or package experience
L2N - novice with various packages, not includings statistics packages
L3N - novice with various packages, including one statistics package
L4N - novice with various packages, including more than one statistics package
Note There was only 1 L4N individual each, on Merlin$_t$ and Minitab$_t$

The seventh proposition was stated as:

Proposition 3.g. - L1 - novices with minimal, or no, computer or package experience, will rate and perform better with Merlin$_t$ than with Minitab$_t$.

Using the same argument as previously, for Proposition 3.b., it was assumed that L1 novices would perform better with Merlin$_t$ and Extended Merlin$_t$ than with Minitab$_t$ and rate them higher. Table 44, below, summarizes the findings. Merlin$_t$ did, indeed, have better mean time indices than Minitab$_{tt}$ at $p < .10$, but not better error or help indices. The Extended Merlin$_t$ did not produce performance time differences significantly different from Minitab$_t$, nor was its Error Index different from Minitab$_t$'s. The Help Indices on the Extended Merlin$_t$ were not useful, since they could not be traced. Although there were no significant differences found in the Error and Help Indices, there were differences in the Time Index, so Proposition 3.g. was supported for performance. There were no significant differences among the novices, in terms of liking for any of the packages, therefore Proposition 3.g. was not supported for perceived comfort. These results should be viewed with caution, however, owing to the small sample sizes.

Table 44
Comparison of L1
by Package

		N	Mean	Differences
TIDX	Mer Min Mer+	8 10 7	137 9 83.0 81 4	Mer > Min $p < .10$
LIKE	Mer Min Mer+	7 10 6	3 6 3 9 3 7	N S
EIDX	Mer Min Mer+	8 8 6	21 7 28 6 37 1	N S
HIDX*	Mer Min	8 8	29 3 41 6	N S

Legend N S - no statistically significant differences
L1N - minimal, or no computer or package experience
*Excludes Extended Merlin

The eighth proposition was stated as:

Proposition 3.h. - Novices will make less help calls than Experts on Merlin.

In order to use help, in some sense, one needs to know what to look for. Users tend to prefer an exploratory approach to learning packages. (Carroll and Mack, 1985; Hiltz and Kerr, 1986). With these in mind, it was reasonable to assume that the various novice categories would make less help calls than the expert categories. There was no statistically significant differences found, however, between Novices and Experts. (See Table 45, below). Proposition 3.h. was, therefore, not supported. It should be noted, however, that the sample size of Experts is small, only six.

Table 45
Comparison of Microcomputer Experience on Merlin

	Novice		Intermediate		Expert		Differences
	N	Mean	N	Mean	N	Mean	
TIDX	61	158.4	28	163.8	6	453.7	E > N, I p < .05
LIKE	57	3.5	30	3.6	6	3.5	N.S.
EIDX	60	72.2	25	39.5	6	8.7	N.S.
HIDX*	60	86.5	25	69.6	6	14.4	N.S.

Legend N.S. - no statistically significant differences
*Excludes Extended Merlin

The ninth proposition was stated as:

Proposition 3.i. - Experts and Intermediates will have better performance time and error indices and perceived comfort ratings than Novices, on Merlin.

Table 45, above, seems to support conventional wisdom and to contradict Hauptmann and Green's (1983) findings, with respect to Experts who outperformed both Novices and Intermediates, at $p < .05$, in terms of the Time Index. Intermediates, however, performed at the same level as Novices. There were no statistically significant differences in EIDX or the LIKE rating. Although the sample size of Experts is small, the direction of the Indices is consistently better from Novice to Intermediate to Expert, nonetheless, Proposition 3.i. was only partially supported.

5.B.5.c. **Results - Research Question 4 - Propositions 4.a.-4.c.**

In this section, we will examine the propositions which were developed relating to the effect of computer anxiety, gender and quantitative competence on perceived comfort and performance measures relating to **ease of use**. As with the analyses of package treatment differences, the L4 and expert individuals were excluded from these analyses because of their small sample size and non-randomness.

5.B.5.c.1. **Results - Anxiety**

The proposition was stated as:

Proposition 4.a. - Those with higher anxiety scores will have worse performance and perceived comfort scores than those with lower anxiety scores, regardless of package treatment.

Assuming that more experience is likely to lead to less anxiety with respect to using a new package and also to better performance scores, we suggested that those with higher performance scores would have lower anxiety scores. This, indeed, proved to be the case. As the correlation coefficients in Table 46, below, indicate, anxiety was negatively related to performance (TIDX), which was significant at $p = .007$. This significance remains on the retest with a p-value of .003. The p-values on the other Indices were not statistically significant. Proposition 4.a. is, therefore, partially supported and supports Gilroy and Desai's (1986) findings, at least with respect to performance, if not perceptions.

Table 46
Performance and Perceived Comfort by Anxiety

	N	Correlation Coefficient	p -value
Initial Use			
TIDX(1)	210	-0 223	.007
LIKE(1)	214	+0 003	780
EIDX(1)	191	+0 060	469
HIDX*(1)	148	+0 059	554
Retest			
TIDX(2)	101	-0 293	003
LIKE(2)	94	+0.022	.771
EIDX(2)	98	+0.153	133
HIDX*(2)	75	+0 165	157

*Excludes Extended Merlin,

5.B.5.c.2. **Results - Gender**

Speculating that males may use different parts of the brain and may, as a consequence, be more mathematically and machine-oriented, we suggested that their performance may be better than females. The proposition was stated as:

Proposition 4.b. - Males will exhibit higher performance scores than females.

The results, both on initial and retest use of the packages, seemed to confirm this. Males had a statistically significantly higher mean TIDX than females. (See Table 47, below). This should, however, be viewed with caution. Underlying causes other than gender per se are most probably confounding this result, such as income and educational level of parents, exposure to video arcade and home computer games, to name a few. Our finding is in contrast to that of Murhpy (1992) who found that gender did not have a significant impact on performance.

Table 47 Performance and Perceived Comfort by Gender							
	Gender	Initial Use			**Retest		
		N	Mean	Differences	N	Mean	Differences
TIDX	1-Males 2-Females	95 120	155.2 119 6	p < 05	55 72	357.2 260 4	p < .05
LIKE	1-Males 2-Females	99 121	3.4 3.3	N S.	47 69	2.7 2.7	N.S.
EIDX	1-Males 2-Females	91 105	41 7 46 1	N.S.	53 70	18 6 24 0	N.S.
HIDX*	1-Males 2-Females	70 80	67 3 74 0	N S.	34 44	23.3 25 0	N.S.

Legend N S - no statistically significant differences
*Excludes Extended Merlin
**No L4's or Experts on the retest

5.B.5.c.3. **Results - Quantitative Competence**

We suggested a proposition for examination concerning quantitative competence. It stated:

Proposition 4.c. - Higher reported quantitative competence will result in better performance and perceived comfort scores.

Our results indicate that on the initial trial, reported competence had an impact on performance, though this disappeared on the retest. (See Table 48, below). Those reporting excellent quantitative competence had statistically significantly higher performance (TIDX) and 'liking' ratings than those reporting Poor or Average competence, at $p < .05$. These former individuals also made significantly less errors/score than those reporting Poor competence, at $p < .05$. The help calls/score were not statistically different. Proposition 4.c. was supported which negates Evans and Simkin's (1989) findings.

Table 48
Performance and Perceived Comfort by Quantitative Competence

	Quantitative Competence	Initial Use			**Retest		
		N	Mean	Differences	N	Mean	Differences
TIDX	1-Excellent	23	174 6	1 > 2, 3	8	205 9	NS
	2-Average	137	123 7	p < 05	97	291 2	
	3-Poor	12	97 1		11	340 8	
LIKE	1-Excellent	23	2 6	1 < 2, 3	8	2 3	1 < 2
	2-Average	139	3 4	p < 05	87	2 8	p < 10
	3-Poor	13	3 5		10	2 7	
EIDX	1-Excellent	22	24 5	1 < 3	8	29 4	NS
	2-Average	121	50 6	p < 10	95	22 8	
	3-Poor	11	87 1		9	7 8	
HIDX*	1-Excellent	17	27.9	N S	5	22 0	NS
	2-Average	94	88 7		67	24 8	
	3-Poor	8	91 9		5	20 1	

Legend N S - no statistically significant differences
*Excludes Extended Merlin
Like 1- Very much, 2- Above average, 3- Average, 4- Below average, 5- Not at all

5.B.6. Summary - Package and Experience Differences and Other Factors

The findings on performance and perceived comfort for package and experience differences are summarized in tables following. The findings with respect to the propositions are also presented.

5.B.6.a. Summary - Package Differences

Tables 49 and 50, below, summarize the statistically significant differences which were found between the package treatments, on initial and subsequent use. These are based on the Tukey-Kramer comparison for all pairs.

Table 49
Summary of Comparison of Package Differences

	Merlin$_t$ vs. Minitab$_t$	Extended Merlin$_t$ vs. Minitab$_t$	Merlin$_t$ vs. Extended Merlin$_t$
TIDX	(>) $p < .05$	N S	N.S.
LIKE	N.S.	N.S	N.S.
EIDX	(>) $p < .05$	N S.	(>) $p < .10$
HIDX	N.S.	N.A.	N.A.

Legend p - probability level of significance
N S - not significant
N.A - not applicable
(>) - direction of difference, comparing left to right, higher score or higher rating
(<) - direction of difference, comparing left to right, lower score or lower rating

There was more accomplished on Merlin$_t$, as indicated by a better performance Time Index, than on either of the other two treatments, despite the worst Error Indices. Minitab$_t$ had the lowest mean performance Time Index and Error and Help Indices. As Minitab$_t$'s online help is a full manual, its lower Help Index may be indicative of the volume of help screens that must be read before another help command can be issued. In contrast, Merlin$_t$'s online help is arranged around two screens of concise command listings, which users may easily access repeatedly. The length of time needed to read Minitab$_t$'s help would also adversely affect its performance time.

We found that the online Hypertext-based index does not boost performance as such, that is, the Time Index, but does help to reduce the Error Index. The fact of having to move from one keyboard to another undoubtedly contributed to reduced speed in this treatment, thereby affecting the TIDX score.

The perceptions about the packages were about even, with all three packages rated as Average. There was a higher percentage rating any one of the packages Below average and Not at all than rating them Above average and Very much. Nonetheless, the Extended Merlin$_t$ tended to have a higher liking rating than the other two treatments, though it was not statistically significant.

Retest

Table 50
Summary of Comparison of Retest vs. Initial Performance and Perceived Comfort Results of those Retested

	Change(2-1)			Change Comparing Packages		
	Merlin$_t$	Minitab$_t$	Extended Merlin$_t$	Merlin$_t$ vs Minitab$_t$	Extended Merlin$_t$ vs Minitab$_t$	Merlin$_t$ vs Extended Merlin$_t$
TIDX(2-1)	+253.7 $p < .05$	+122.1 $p < .05$	+212.1 $p < .05$	$p < .05$	N.S	$p < 05$
LIKE(2-1)	+0.9 $p < .05$	+0.5 $p < 05$	+0.3 $p < .10$	N.S.	N.S.	N S
EIDX(2-1)	-93.7 $p < .05$	-2.3 N.S	-9.7 N.S.	$p < .05$	$p < .05$	N.S.
HIDX(2-1)	-126.2 $p < .05$	-31.5 $p < 05$	N A.	$p < 05$	N.A.	N.A.

Legend: p - probability level of significance
N.S. - not significant
N.A. - not applicable
(2-1) - difference between Retest and Initial results

In all three treatments, all the indices showed marked improvement for those redoing the same task on the same package a week later, except for the Error Indices on Minitab$_t$ and Extended Merlin$_t$. These remained about the same which does not seem to support our earlier suggestion that increased performance leads to more errors. We proposed that the lower error index on Minitab$_t$, on initial use, when compared to either of the Merlin$_t$'s, was due to less being attempted, as indicated by its lower TIDX. However, on the retest, while $Minitab_t$'s TIDX increased substantially, its EIDX did not change. The same was true for the Extended $Merlin_t$. Both the $Merlin_t$'s EIDX's had statistically greater improvements on the retest than $Minitab_t$, at $p < .05$. $Merlin_t$'s TIDX and HIDX also showed significantly greater improvements than Minitab's, at $p < .05$.

We also found that of those retested, the advantage seen initially, of having the hypertext-based online index, was removed. The Extended $Merlin_t$'s TIDX went lower than that of the other two treatments on the retest, where it had been the highest initially. Its liking rating became marginally worse than $Merlin_t$'s, whose rating became marginally better than $Minitab_t$'s. None of these differences in liking ratings were statistically significant. The Extended $Merlin_t$'s EIDX is also no longer the lowest, that place now taken by the $Merlin_t$ stand alone.

In summary, it would seem that subsequent use promoted increased performance and better perceived comfort of the packages, accompanied by a reduction, or no change, in the errors and help calls made. This improvement was greatest on the $Merlin_t$ treatment, the mixed command/menu structure with concise online assistance.

Propositions

The propositions which were put forward are restated with their findings for reference. They were:

Proposition 2.a. - S - Merlin$_t$ will produce better performance measures than Minitab$_t$.

Proposition 2.b. - NS- Merlin$_t$ will be given better perceived comfort ratings than Minitab$_t$, (except by users who already have a certain level of experience with Minitab$_t$, to be discussed later).

Proposition 2.c. - NS- Users of the Extended Hypertext-based Merlin$_t$ will have better performance measures than either the stand-alone Merlin$_t$ or Minitab$_t$, and will have better perceived comfort ratings than either of them.

Proposition 2.d. - S - Users of Merlin$_t$ will show a greater improvement in performance, on subsequent use of the package, than will Minitab$_t$ users.

Proposition 2.e. - S - There will be a general improvement in the perceived comfort rating for each of the packages, on subsequent use.

Proposition 2.f. - S - Performance measures and perceived comfort ratings will be better when the dimensions, Speed, Memory, Effort and Comfort, are perceived to be supported by more features in the packages than when fewer are perceived to be supported.

Proposition 2.g. - S - Better perceived comfort, or 'liking' ratings, will result in better performance indices.

Legend: S - Supported; NS - Not supported

Full support was found for Propositions 2.a., 2.d., 2.e., 2.f. and 2.g., while no support was found for Propositions 2.b. and 2.c.

Our theoretical assumption that a mixed menu/command structure, with concise online help, should promote better performance and perceived comfort measures than a strict command structure, with verbose online manual, was supported only for performance.

However, our belief that a Hypertext-based online index should further improve the measures, when using Merlin$_t$, was not substantiated.

Interestingly, although the initial 'liking' rating of Merlin$_t$ was lower, though not statistically so, than that of Minitab$_t$, notwithstanding the better performance indicators, there was a marked, significant improvement on subsequent use. This improvement, however, was not different from that made by any of the other treatments. Our findings, therefore, do not seem to agree with those in the studies by Lee, et. al. (1986), who found that a combined menu and keyword command structure was preferred by their subjects, though they agree with their findings of better performance using this type of structure. One of the reasons for the discrepancies in the research findings on menu versus command structures may be that, in some instances, the menu structure being compared is simply another representation of the command structure, such as that seen in Lotus$_t$. Their menu structure does not direct or focus operation of the package along a pre-specified path as, for example, the menu structure found in Merlin$_t$. Menu structures of this type give the system greater control of the flow of activies which users may find restrictive rather than helpful. This could possibly account for the lower perceived comfort ratings of Merlin$_t$ which we found compared to those of Minitab$_t$.

Our findings also support Carroll (1985), who found that a reduced manual improved performance on initial and subsequent exposure. The emerging interest in the application of hypertext technology to online assistance and other information and learning aids, such as the applications proposed by Shneiderman and Kearsley (1989) and Bieber and Kimbrough (1992), with the anticipated benefits to users, does not seem to be supported by our findings for hypertext as applied here. These showed that there was a lack of improvement in performance and perceived comfort, despite having this assistance provided. Little has been done so far to evaluate the benefits of this approach; research has been mostly developmental.

We also found some evidence for the importance of support for the learning dimensions we identified, Speed, Memory, Effort and Comfort, and for these possibly being important components of **ease of use**. We found that there was a strong relationship between support of these dimensions and the 'liking' rating, though it was low with respect to performance, in that the higher the number of features or attributes in the packages which were perceived to support these dimensions, the better was the 'liking' rating and the performance Time Index. We also found that better 'liking' ratings were associated with better performance.

5.B.6.b. **Summary - Experience Differences**

The findings with respect to experience are summarized in Tables 51-53, following.

Initial Use

Table 51
Summary of Comparison of Experience Differences

	L4 vs. L3, L2, L1	L3 vs. L2, L1	L2 vs. L1	Intermediate vs. Novice	Expert vs. Novice	Expert vs. Intermediate
TIDX	(>) p < .05	N.S.	(>) p < 05	N S.	(>) p < .05	(>) p < .05
LIKE	N.S	(>) p < 05	(>) p < 10	N S	N.S.	N.S
EIDX	N S	N.S.	N S	N S	N S.	N S.
HIDX*	N S	(>) p < 10	N.S	N.S.	N S	N.S

Legend L1 - minimal, or no, experience
L2 - experience with various packages, not including statistics packages
L3 - experience with various packages, including 1 statistics package
L4 - experience with various packages, including more than 1 statistics package
*Excludes Extended Merlin

In summary, the results indicate that the extent of experience has an influence on performance, if not on the errors made and the help required to attain it. The extent of experience on packages of similar function also has an impact. Experts, on $Merlin_t$, had a significantly higher mean TIDX than did Novices or Intermediates, at $p < .05$. Likewise, L4 individuals, those experienced with more than 1 statistics package had a statistically higher performance (TIDX) than all other levels. L3 individuals, those with experience with only 1 statistics package, likewise had significantly higher performance, but required more help to attain it, than L1 individuals who had no experience with statistics package. Having experience with packages, even if not with packages of similar function, L2's, also assisted performance over those with no experience, L1's. L3's and L2's tended to perceive the packages more favourably than those with minimal, or no, experience, L1's.

Retest

Table 52 Summary of Comparison of Experience Differences of those retested				
	L3 vs. L2	L3 vs. L1	L2 vs. L1	Intermediate vs. Novice
Initial				
TIDX	(>) $p < .05$	(>) $p < 05$	NS	NS
LIKE	NS	NS	(>) $p < 05$	NS
EIDX	NS	NS	NS	NS
HIDX*	NS	NS	NS	NS
Retest				
TIDX	NS	(>) $p < 05$	NS	NS
LIKE	NS	NS	NS	NS
EIDX	NS	NS	NS	NS
HIDX*	NS	NS	NS	NS

Legend. (>) - direction of difference, comparing left to right, higher score or better rating
L1 - minimal, or no, experience
L2 - experience with various packages, not including statistics packages
L3 - experience with various packages, including 1 statistics package
L4 - experience with various packages, including more than 1 statistics package
Note. There were no L4's and only 1 Expert on the retest
*Excludes Extended Merlin

On the retest, only L3, those with experience with 1 statistics package remained significantly higher than L1, those with none, or minimal, experience. There were no differences between the other categories. It should be remembered, however, that there were no L4's and only 1 Expert who took part in the retest. Table 53, below, shows the extent of the improvements which took place between initial and subsequent use. These were all significant at $p < .05$. Only L2's those with experience with packages other than statistics packages had a significantly greater increase in performance (TIDX) over L1's those with none, or minimal, experience. All other differences were not statistically significant.

Table 53
Summary of Comparison of Retest vs. Initial Use by Experience Differences of those retested

	Change(2-1)			Change Comparison		
	L1	L2	L3	L1 vs. L2	L1 vs. L3	L2 vs. L3
Statistics Package Experience						
TIDX(2-1)	+90 0 p < .05	+213 0 p < 05	+185.1 p < 05	p < .05	N S	N.S.
LIKE(2-1)	+0 8 p < .05	+0.5 p < .05	+0 6 p < .05	N.S.	N.S.	N.S.
EIDX(2-1)	-1 3 N.S	-38 2 p < 05	-36.2 N S.	N S.	N.S.	N.S.
HIDX*(2-1)	-16 9 p < 05	-90.1 p < .05	-37.3 N.S.	N.S.	N.S.	N.S.
Microcomputer Package Experience						
	Novice	Intermediate	Expert	Novice vs. Intermediate	Expert vs. Novice, Intermediate	
TIDX(2-1)	+ 179.2 p < .05	+204 5 p < 05	+406 7	N.S.	N A.	
LIKE(2-1)	+0 7 p < 05	+0.4 N S	+50.0	N.S.	N.A.	
EIDX(2-1)	-33 2 p < 05	-27 3 p < .05	-90 0	N.S.	N.A.	
HIDX*(2-1)	-77.2 p < 05	-72 4 p < 05	-95.6	N.S	N A.	

Legend N A - not applicable
N S - no statistically significant differences
(2-1) - Retest vs Initial use
L1 - minimal, or no, experience
L2 - experience with various packages, not including statistics packages
L3 - experience with various packages, including 1 statistics package
L4 - experience with various packages, including more than 1 statistics package
Note There were no L4's and only 1 Expert on the retest
*Excludes Extended Merlin

Propositions

The propositions related to experience and their findings are summarized below for reference. They were:

Proposition 3.a. - NS - L3 users of various packages, including one statistics package, will perform worse on Merlin$_t$, and give it lower perceived comfort ratings, than L2 users, who have had experience with various packages, not including any statistics packages.

Proposition 3.b. - PS - L2 users with experience with various packages, not including any statistics packages, and L3 users of various packages, but only one statistics package, will outperform L1 - minimal, or no, experience, but give lower perceived comfort ratings, regardless of package treatment.

Proposition 3.c. - PS - L2 users with various packages, not including statistics packages, will perform better on Merlin$_t$ than on Minitab$_t$, and give it better perceived comfort ratings.

Proposition 3.d. - PS - L3 users with one statistics packages, which was Minitab$_t$, as well as other packages, will give lower perceived comfort ratings to Merlin$_t$ than to Minitab$_t$, and perform better.

Proposition 3.e. - S - L4 users of various packages, including more than one statistics package, of which Minitab$_t$ is one, will outperform all other levels, on Merlin$_t$.

Proposition 3.f. - PS - Novices, regardless of level, will have the same level of perceived comfort on the same package.

Proposition 3.g. - PS - L1 - minimal or no computer or package experience, will rate and perform better with Merlin$_t$ than with Minitab$_t$.

Proposition 3.h. - NS - Novices will make less help calls than Experts, regardless of package treatment.

Proposition 3.i. - PS - Experts and Intermediates will have better performance time and error indices and perceived comfort ratings than Novices, on Merlin$_t$.

Legend: S - Supported; NS - Not supported; PS - Partially supported

Full support was found only for Proposition 3.e. Partial support was found for Propositions 3.b., 3.c., 3.d., 3.f. and 3.g. There was no support found for Propositions 3.a. and 3.h.

The results found on Proposition 3.d., with respect to 'liking', support our belief that familiarity with a package does create resistance to a new package of similar function, indicated by a lesser 'liking' rating for the unfamiliar package. Although the sample size was small in L4, being only eleven subjects, the findings on Proposition 3.e. suggest that this resistance is overcome, in terms of performance, if not perceptions, by experience gained in many different packages of similar function. L4 was seen to outperform all other levels on the TIDX, on an unfamiliar package, at $p < .05$.

The advantage of prior exposure to packages of similar and dissimilar function compared to no experience was also seen in the additional findings reported under Propositions 3.a. and 3.b. L3 users with experience with one statistics and other packages, as well as L2 users, those with experience with packages other than statistics packages, were seen to have better time performance scores than L1, who had had no prior exposure to packages. We also saw this pattern in the findings on package design, whereby liking for the packages increased on subsequent use, even if it was not statistically significant.

Proposition 3.a., which examined individuals with experience with one statistics package on $Merlin_t$ versus those with no statistics package experience, found no significant differences in either performance or in perceived comfort, for L3 versus L2. No significant differences in perceived comfort was found on Proposition 3.c. between $Minitab_t$ and $Merlin_t$

for those subjects targeted, but these individuals had a significantly higher TIDX on Merlin$_{t}$ than on Minitab$_{t'}$. Prior exposure, then, does seem to have a positive effect on performance with, and a negative effect on perceived comfort with particular package designs, as also indicated from the findings on the other propositions to be discussed below. Foss and DeRidder (1989) likewise found a positive transfer from other text-editors to DEC text-editors for inexperienced computer text-editors. Karat, et.al. (1986), on the other hand, found that prior experience with wordprocessors hindered performance with an unknown wordprocessor. Hauptmann and Green (1983), in their study, found that there were no differences in performance time or accuracy for experienced programmers, using unknown menu, command and natural language systems. It seems that computer experience alone is not a determining factor, but rather, package experience is. Our findings suggest that both are contributing factors.

It would also seem that for initial use of a package by novices, a mixed menu/command system, with concise online help, is more advantageous for performance and perceived comfort than a command driven system, with full online manual, or a mixed menu/command system, with a Hypertext-based online help index, with explanations and examples. The findings on Proposition 3.f. indicated that there was no difference in performance and perceived comfort between any of the novice categories on Merlin$_{t'}$; however, there were differences on Minitab$_{t}$ and Extended Merlin$_{t'}$, wherein L1N was significantly lower in performance, at $p < .05$, than L3N. On Minitab$_{t'}$, L1N had also lower perceptions of the package than the other novice categories. This was seen as well from Proposition 3.g., in which Merlin$_{t}$ produced a better TIDX for L11 first-time novices than either Minitab$_{t}$ or Extended Merlin$_{t'}$, at $p < .10$. As we know, this debate is still open. Highly contradictory results have been found from the research in this area. Whiteside, et.al. (1985) found that commands were better for all user categories compared to menus, whereas Hauptmann and Green (1983) did not find any differences in performance for their users, on either menu, command, or natural language systems. Ogden, et.al., as reported in Paap and Roske-Hofstrand (1988), found that not all users preferred to use the menus in their system; rather, they found that novices preferred commands, while expert users favoured menus, which was contrary to common belief. Khalifa (1990) suggested and found that the simpler interface was preferred by novices, and that the more complex interface, though more difficult to learn initially, was easier to use once learned and preferred by experienced users. We did not find this to be the case. $Merlin_t$, the simpler interface, outperformed $Minitab_t$, the more complex interface, both initially and on reuse. Also, there were no significant differences in perceived comfort with the packages nor between novices and experts either in performance or perceived comfort.

The additional assistance provided by the Hypertext-based online help index to the $Merlin_t$ package did not improve performance, but it did reduce the number of errors. This is congruent with the findings obtained by Burns, et.al. (1986) in their study of improvements made to screen displays used by flight controllers. Performance time remained the same, but errors were reduced. This is an important consideration in evaluating ease of use. Reduction in the number of errors may be as important, or even more so, than speed. The effect on peace of mind of the user is not easily measured, but is certainly an important factor to support, as we have suggested throughout this thesis.

Proposition 3.h. showed no difference statistically in help calls/score made between Expert and Novices, though Experts did have a lower index, 18.7 versus 75.0. Experts were also able to accomplish more, 414.3 versus Novices, 130.9, and Intermediates, 146.2. Whereas experts seem to be able to overcome hurdles encountered, novices cannot do so readily, or certainly, not as quickly. Perhaps novices would benefit more from offline documentation. Relles (1979) found that novices performed better with a printed manual, while experts performed better with online assistance only. Dunsmore (1980) found that novices performed worse when given only a brief summary of the system to support the online assistance, compared to having the summary plus an offline manual. Carroll's study (1985) goes further to suggest that the manual should be tailored to the novice. He found that this category of user performed better with a reduced manual than a full manual.

5.B.6.c. **Summary - Other Factors**

The propositions related to anxiety, gender and quantitative competence, with their results, are summarized here. They are:

Proposition 4.a. - PS - Those with higher anxiety scores will have worse performance and perceived comfort scores than those with lower anxiety scores, regardless of package treatment.

Proposition 4.b. - S - Males will exhibit higher performance scores than females.

Proposition 4.c. - S - Higher reported quantitative competence will result in better performance and perceived comfort scores.

Legend: PS - partially supported; S - supported

Full support was found for Propositions 4.b. and 4.c., but only partial support for Proposition 4.a. Anxiety with computers was found to be negatively related to performance on initial use and retest, but its relationship with perceived comfort was not statistically significant. Males were found, both on initial use and the retest, to have a higher performance TIDX than females. This finding may be confounded by other factors, however, such as parent's income and educational level and exposure to home computer and arcade games. It should, therefore, be viewed with caution, especially as this was not detected in previous studies. The fact that, to date, software is predominantly designed by the male population may also be a factor. This being the case, it may suggest that an area of research should address the possible special requirements females may have in learning to use a package and make accommodation for this in design or assistance provided. Confirming conventional wisdom, those rating themselves as having Excellent competence in quantitative courses had significantly better performance and perceived comfort ratings, on initial use, than those rating themselves as Poor or Average which was contrary to Evan and Simkin's (1989) study. These differences disappear on the retest except that those rating themselves Excellent gave better 'liking' ratings than those rating themselves Average.

5.B.7. Summary of Findings in Study 2

We suggested that the learning dimensions, **Speed, Memory, Effort and Comfort**, constitute components of **ease of use**, and that these are likely to be impacted by package design, experience level, and other factors.

With respect to the learning dimensions, we found that when more attributes of the packages were perceived to support these learning dimension, both performance and perceived comfort were better. Understandably, the relationship between perceived support of the dimensions and perceived comfort was stronger than performance. Nonetheless, the results lend some evidence to the importance of support of these dimensions for **ease of use**, bearing in mind as always that the study is an exploratory one.

Despite the fact that subjects made more errors and initiated somewhat more help calls (**memory, effort**) and liked the package somewhat less (**comfort**), performance time (**speed**) on Package 1 (Merlin$_t$) was better than on Package 2 (Minitab$_t$) by 64%. (See Table 24 in Section 5.B.3.a.). The magnitude of this difference in performance was consistent over all experience levels, with the exception of those having previous experience with Package 2, in which case the performance advantage was reduced to 16%. (See Table 38 in Section 5.B.4.a.).

The reason(s) for this higher performance result could not be deduced by analyzing user's perceived comfort with the package. However, we believe it was related to command structure differences, Package 1 having a mixed directed menu/command structure and Package 2, a pure command structure. In addition, the HELP messages in Merlin$_t$ were more concise and context dependent. This conclusion is reached based on the comparison of the two packages as shown in Appendix B. The two packages were quite similar. The differences were in the command structure and online help format, including the fact that Merlin$_t$ had additional assistance features, such as Input Error Correction, Error Recovery, Default values, Memory Jogs, Expertise Accommodation, Question or Prompt assistance, Navigational Aids and partial Context Help. The requirements for the task were very similar, the major difference being in the editing conventions, and the fact that Merlin$_t$ allowed the use of several commands for exiting.

It is also unclear why, with such superior performance measures, Package 1 had lower perceived comfort ratings than Package 2, though the univariate analyses indicate they were not statistically significantly different. One possible explanation may again be a consequence of the command structure. We noted previously that Merlin$_t$'s menu structure links customary sequences of activities to achieve desired operations. For instance, the menus guide the user through the choice of data entry type - keyboard, read a file, generate random nos. in col. 1, or create stepped or const. data through edit sub-system, to the adding of labels and the printing of data onto the screen. From here, the user must invoke commands to accomplish the tasks of editing, saving and exiting. We noted in our analysis of the sub-tasks that a greater percentage of users had difficulty with data entry on Merlin$_t$ than on Minitab$_t$. It may be that users feel constrained by having the system direct the interaction, preferring the freedom offered by a command structure. The user may feel more in control with this latter structure, notwithstanding the superior performance the menu system can offer. Other than the manner in which the data operation is invoked, menu in Merlin$_t$ and command in Minitab$_t$, the rest of the data entry should have been comparable, users enter one row at a time with a carriage return. In fact, Merlin$_t$ should be clearer, since users are prompted with the row number for each line of entry, whereas Minitab$_t$ simply displays, 'DATA>'. To end data entry, Merlin$_t$ required a carriage return and Minitab$_t$ required the command 'END'. (See Appendices C and D).

The adage 'first impressions are lasting' may also be a factor in the perceptions reported. Data entry was the first encounter with the package, and as this was problematic for more users on Merlin$_t$ than on Minitab$_t$, it may have influenced their overall impressions and, hence, ratings.

In terms of data editing, Merlin$_t$ perhaps has more conceptual hurdles to overcome than Minitab$_t$. Minitab$_t$ provides only one way to edit, that is, by element, and not by row, and is issued at the same command level as all commands. The syntax is also the same as that used in the popular Lotus$_t$ package, **A15**, column then row, which on Minitab$_t$ is **LET c1(15)=55**. Merlin$_u$, on the other hand, requires a change of command mode to edit mode before the edit command can be issued. It also allows editing by row or element and uses matrix notation convention. The sequence required to perform this operations is: **EDIT, R EL(15,1) <u>or</u> R ROW15,** enter no.(s) changed on next line, press return to return to exit edit mode. In addition to this difference between Minitab$_t$ and Merlin$_u$, the example of this operation is also presented differently in the two packages. If users do find the correct help screen in Minitab$_u$, the example of changing a data element is very clear, with only one option identified and spacing is not important. In Merlin$_u$, a list of various editing options is provided, of deleting, adding, changing, printing, et cetera. This may have been a source of confusion for users. The process traces on Merlin$_t$ indicate that most tried to change the whole row rather than the element, which was not evident on Minitab$_u$. Mainly, they had difficulty with the syntax of the command, since spacing is important in Merlin$_t$'s syntax. Spacing seems to give users difficulty, as noted in our analysis of sub-tasks to follow, wherein several users inserted unnecessary spacing between data entries. This suggests the benefit of allowing format-free data entry.

Another possible explanation could be the nature of online assistance provided with the two packages. Package 1's assistance is composed of a concise listing of commands on two screens, with a brief explanation of each command. The assistance on Package 2 is a full online manual. We noted that with the addition of the Hypertext Index to Package 1, although performance was not improved, the liking rating was somewhat higher than for either Package 1 or Package 2. It may be that having more verbose explanations provides the user with some level of 'comfort', which, evidently, will impact their subjective assessments. Whereas the concise assistance permits speedier performance, perhaps users feel unsure of how well they understand the underlying concepts. The expanded explanations provide this deeper understanding which is necessary to promote confidence.

The impact of the Hypertext HELP in Package 3 (Package 1 + Help Index) was relatively small, as subjects performed at a lower TIDX level, 133.8 than on Package 1, 178.7, despite having a lower error level, 33.8 compared to 59.1. (See Tables 24 and 28). Complete novices, L1's, were particularly hampered by this addition to the package, resulting in performance being substantially below Package 1, 81.4 versus 137.9. (See Table 46). Nevertheless, as noted above, the perceived comfort rating of this package treatment was somewhat higher than it was on the other two treatments. While the somewhat lower overall TIDX score may be explained by the slower Olivetti machines which were used on this treatment, and the fact of having to move between keyboards, this does not account for the large difference found among complete novices.

Regardless of the package, those with intermediate and expert experience had somewhat better 'ease of use' performance measures than did novices. This latter group had the greatest difficulty with the packages, while those experienced with various packages, including more than one statistical package, had the least difficulty. It was also found that 'ease of use' performance measures were better for L3 and L4 individuals who had had prior exposure to at least one package of similar function compared to L2's with prior exposure to dissimilar packages. The former had performances which were 36% and 156% better, respectively. We believe that prior exposure to the distinct 'look and feel' of line-editor statistical packages, which is very different from that of package such as Lotus$_t$ or database packages, may account for this finding.

Whereas prior exposure to packages of similar function was able to assist performance, it resulted in resistance to a new package of similar function, as seen in the somewhat lower initial perception ratings which these users gave to Merlin$_t$ compared to Minitab$_t$. These perceptions improved, however, with subsequent use of the package.

We also found that anxiety, gender and quantitative competence had an impact on performance, the latter also impacting perceived comfort. We have already noted some possible reasons for the finding with respect to gender and suggest caution in interpreting this finding.

There were marked improvements on subsequent use, in terms of both performance and perceived comfort ratings on all packages and for all experience levels. It is interesting to note, however, that although those taking part in the retest one week later initially had 54.7% better performance on Merlin$_t$ than those on Minitab$_t$, this margin widened to 86.9% on the retest. (See Table 25).

The greatest performance improvement on Merlin$_t$ was seen in intermediates with experience on many packages, including one statistical package which was Minitab$_t$, who improved 313%, suggesting that the conceptual hurdles of Minitab$_t$ were overcome by these individuals. On this package, complete novices showed the greatest improvement in perceived comfort of the package, from 3.8 to 2.8.

Anxiety and gender continued to be a contributing factor to performance on the retest, whereas quantitative competence did not.

In this section, we discussed the effect of different package designs, different types of experience, microcomputer package experience and statistics package experience, and factors, such as computer anxiety, gender and perceived quantitative competence, on perceived comfort and performance measures relating to **ease of use**. In the next section, we fit these variables to multiple regression models in an attempt to explain and predict **ease of use**.

5.B.8. **Results - Regression models**

The model adopted in the thesis was based on the assumption that the controlled, independent variables - package design and assistance features, would have an impact on perceived comfort and performance measures relating to **Ease of Use**. This was further assumed to be mediated by the controlled, constant variables - Package Class, Task, Instructional Strategy and User Role, and by the uncontrolled variables - User Characteristics. In Study 2, we examined the effect of package differences on perceived comfort and performance, and the effect of the user characteristics, experience level, gender, perceived quantitative competence and computer anxiety. We fitted these variables to a regression model to determine the extent to which they could be said to predict and explain our measures relating to **ease of use**.

Our dependent variables were comprised of three performance measures relating to **ease of use** and one measure of perceived comfort:

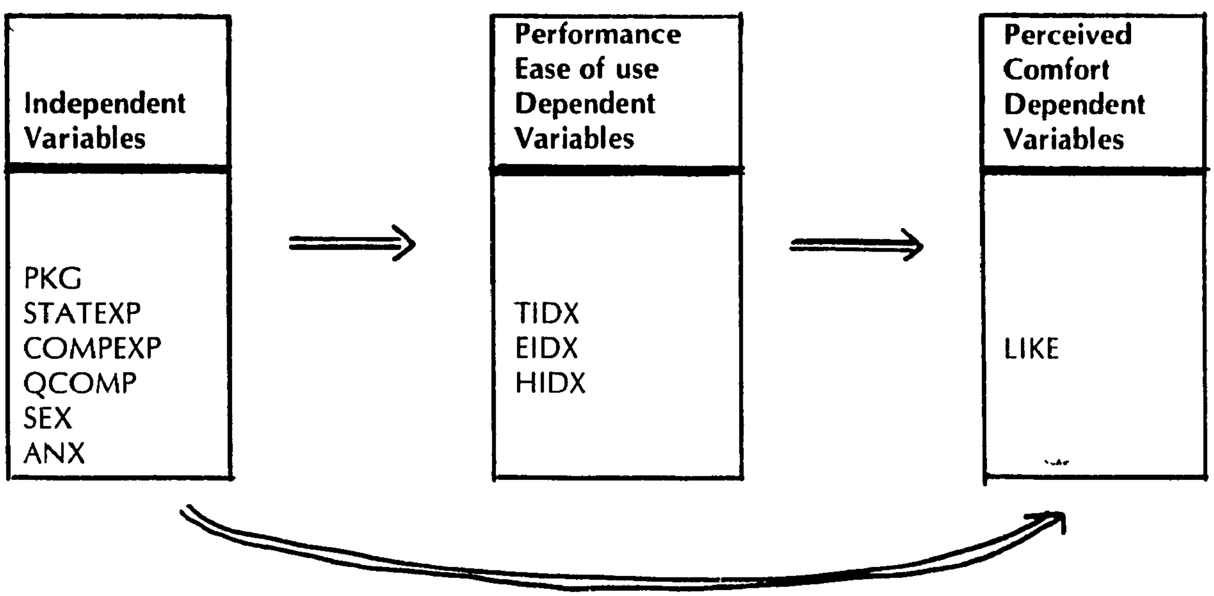
Time Index (TIDX) - score received on the task divided by the time taken on the task; a high value indicates a high score, or better performance;

Error Index (EIDX) - number of raw errors divided by the score on the task; a high value indicates high number of errors;

Help Index (HIDX) - number of raw help calls divided by the score on the task; a high value indicates high number of help calls;

Comfort (LIKE) - measured on a 5-point likert scale where, 1 - Liked very much, 2 - Liked above average, 3 - Liked average, 4 - Liked below average, 5 - Liked not at all; a high value indicates low liking or perceived comfort of the package.

The data collected permitted us to model these dependent variables in relation to the independent variables - package treatments (PKG), statistics package and general microcomputer package experience levels (STATEXP) and (COMPEXP), gender (SEX), perceived competence in quantitative courses (QCOMP), and a general computer anxiety score (ANX). The models we tested were based on the following:



We were first interested in the effect of PKG, STATEXP, COMPEXP, QCOMP, SEX and ANX on the performance measure TIDX (time), and how this might be explained by the effect on EIDX (errors) and HIDX (help). We were then interested in the effect these variables may have had on perceived comfort, or LIKE, and how this might be moderated by performance.

The dependent variables TIDX, EIDX and HIDX are interval variables, while the LIKE variable is ordinal. The independent variables PKG, STATEXP, and SEX were put into the model as categorical variables. COMPEXP and QCOMP were input as ordinal variables. ANX was input as an interval variable. Regression models were run using step-wise regression to establish the fit of the independent variables to the dependent variables. In the case of LIKE, as the dependent variable, logistic regression was used to account for the ordinal nature of the variable. The resulting final regression models can be seen in Table 54, which follows. Because of the small sample sizes of the following categories, and the fact that they were not represented in all the package treatments, the models do not include the nine Level 4 experts who were assigned only to $Merlin_u$, nor one remaining expert and one Level 4 individual on $Minitab_t$. As a consequence, we were not able to model the effect of experience with more than one statistics package nor the effect of expert experience. Our univariate analyses do indicate, however, that these may have a significant impact on performance and perceived comfort. Also, as our univariate tests showed no statistically significant differences between the L3 individual on either $Minitab_t$ or $Merlin_u$, they were not separated in the model. We did test whether there was an interaction effect between STATEXP and PKG but none was found. There was also no interaction effect found between STATEXP and COMPEXP.

The results of our analysis of the effect of our dependent variables on the performance measures relating to **ease of use** are shown in Table 54, below, and on perceived comfort in Table 55, following. Because of the skewed nature of the data visible in the analysis of the residuals, it was necessary to perform transformations on the dependent performance variables. A square root transformation was performed on the Time Index and cube root transformations on the Error and Help Indices. (Neter, Wasserman and Kutner, 1985). None were performed on the Comfort Index.

The small R^2's of our models suggest that much of the variance is still unexplained. The variables we identified, while they are significant factors, do not tell the whole story. Other User Characteristics, such as psychostructure makeup and general microcomputer experience, are perhaps more important for explaining **ease of use.** The low predictive power of the models may also be due to the fact that the study is an exploratory one and this tends to increase the amount of error. Our constructs are also still at the developmental stage and in need of further refinement.

It is interesting to note that in Murphy's (1992) study, gender, age and experience on other systems were not found to have significant explanatory power of performance, while the Learning Style Inventory measure of cognitive style and attitude to computers were found to be covariate with performance. In his study, experience was assessed in terms of general computer experience and not differentiated based on extent of experience with particular packages of similar and dissimilar function. He was, of course, testing a contrived interface and not a commercial package which perhaps makes this unnecessary. Attitude was assessed based on six general questions on perceptions of ease of use of computers, usefulness, enjoy using, unconcern about pressures to learn, the necessity of everyone learning something about computers, and expecting to use computers frequently during career.

It should be noted that as the models include interval, categorical and ordinal data, the beta coefficients are only comparable for the categories within each variable, which are shown in the top portion of Table 54, below. The lower portion of the table shows the p-values resulting from the F-tests on the grouped categories, which was the basis used for determining which variables remained in the model. The effect of these variables on the measures relating to **ease of use** are presented below, bearing in mind that while the variables are statistically significant, they have low predictive value.

TIDX - Those with experience with a variety of packages, but not with statistics packages had significantly lower performance, with respect to the TIDX Index, compared to the others. There were also statistically significant differences in package treatments. Minitab$_t$ had significantly lower performance compared to the others, while Merlin$_t$ had significantly higher performance. Anxiety was negatively related to performance, that is, those with higher anxiety scores had statistically lower performance than those with lower anxiety scores. Males had marginally higher performance than females.

EIDX - Both package and perceived quantitative competence had a statistically significant effect on errors made/score. Merlin$_t$ had significantly more errors/score than the other treatments, while those rating themselves as Average had less errors per score than those rating themselves as having Poor competence.

HIDX - Only perceived quantitative competence had a statistically significant effect on the help calls made/score. Those who rated themselves as having Excellent competence made significantly less help calls/score than those rating themselves as having Poor competence.

LIKE - Perceived quantitative competence had the most statistically significant effect on our measure of perceived comfort, with those rating themselves as having Excellent competence giving significantly better liking ratings to the packages than those rating themselves as having Poor competence. Package differences was the only other variable having a statistically significant effect on the Comfort Index. Merlin$_t$, the mixed menu/command structure with concise online help, was given significantly lower ratings than the other package treatments. When the performance Indices are added to the model to determine their moderating effect, the Time Index is found to be an even more significant factor affecting perceived comfort than either package or quantitative competence, better perceived comfort ratings being related to higher or better performance. The Time Index does include time spent making and correcting errors, and accessing help.

Of the variables we considered, only self-assessed quantitative competence did not have a significant effect on the performance Time Index, though it most certainly had on the Error and Help Indices and on the Comfort measure. As we noted previously, we could not test the effect of general microcomputer experience because of the small sample sizes, though we believe this would add to the explanatory and predictive power of the model.

Table 54
Fitted Regression Models
with respect to
Performance

SQRTIDX			CUBEEIDX			CUBEHIDX		
R^2 .16 N 208 p < .01			R^2 .10 N 154 p < .01			R^2 .06 N 119 p < 05		
Factors	**Coeff-icient**	**p-value**	**Factors**	**Coeff-icient**	**p-value**	**Factors**	**Coeff-icient**	**p-value**
*PKG[merlin]	+1 03	021	*PKG[merlin]	+ 0 43	.004	QCOMP[2]	-0.21	.703
*PKG[minitab]	-1 18	.009	*PKG[minitab]	-0 23	.129	QCOMP[3]	-1 00	.013
*SEX[male]	+0 64	.057	QCOMP[2]	-0.94	.029			
ANX	-0 10	029	QCOMP[3]	-0 50	.110			
*STATEXP[1]	-0 73	304						
*STATEXP[2]	-1 00	041						
PKG		010	PKG		.014	QCOMP		.037
SEX		057	QCOMP		.017			
ANX		029						
STATEXP		001						

Note
*indicates relative to average
Lower half of the table indicates the p-values of the grouped categories which were used to determine significant variables to retain in the model
STATEXP[1] - individuals with minimal, or no, package experience
STATEXP[2] - individuals with experience with various packages, but no statistical packages
STATEXP[3] - individuals with experience with various packages, including one statistical package
QCOMP[1] - individuals with Low perceived quantitative competence
QCOMP[2] - individuals with Average perceived quantitative competence
QCOMP[1] - indiviudals with Excellent perceived quantitative competence

Table 55
Fitted Regression Models
with respect to
Perceived Comfort

LIKE R^2 04 N 175 $p <$ 01			LIKE (TIDX added) R^2 08 N 166 $p <$ 01		
Factors	**Coeff-icient**	**p-value**	**Factors**	**Coeff-icient**	**p-value**
*PKG[merlin]	-0.51	010	TIDX	+0 01	000
*PKG[minitab]	+0 24	.217	*PKG[merlin]	-0 67	002
QCOMP[2]	+0 07	891	*PKG[minitab]	+0 33	114
QCOMP[3]	+1 68	000	QCOMP[2]	+0 20	727
			QCOMP[3]	+1 40	002
PKG		.033	TIDX		000
QCOMP		001	PKG		006
			QCOMP		006

Note:
*indicates relative to average
Lower half of the table indicates the p-values of the grouped categories which were used to determine significant variables to retain in the model
QCOMP[1] - individuals with Low perceived quantitative competence
QCOMP[2] - individuals with Average perceived quantitative competence
QCOMP[3] - indiviudals with Excellent perceived quantitative competence

We discussed the impact of the variables we identified as potentially having an effect on **ease of use** in the context of overall regression models in this section. To gain further insights into what may facilitate and detract from **ease of use**, we examine the performance of our sample on the sub-tasks in the next section.

5.B.9. **Results - Performance on Sub-tasks**

In Table 56 following, is a summary, by package treatment, of the sub-tasks correctly executed; those attempted, but not achieved; and those not attempted.

5.B.9.a. Correctly executed - Initial Use

Data Entry

The same percentage, 74%, on Minitab$_I$ and Merlin$_I$ were able to accomplish Data Entry. On Extended Merlin$_I$, this figure was 82%. In the Minitab$_I$ treatment, 48% of the subjects were able to successfully add labels to the data entered. In the Merlin$_I$ treatment, 62.2% were able to do this, and in the Extended Merlin$_I$, 66% accomplished it.

Editing

Changing, or editing, was the most difficult sub-task for all the package treatments. Only 33.3% were able to successfully accomplish this on the stand alone Merlin$_I$, 25% on Minitab$_I$. There was more success on Extended Merlin$_I$ with 44% successfully completing this task.

Viewing

Minitab$_I$ and Extended Merlin$_I$ seemed to provide greater difficulty in Viewing or printing the input on the screen than did the stand alone Merlin$_I$. Here, 66.7% were able to successfully complete the task, while only 51.1% were able to do so on Minitab$_I$, 52% on Extended Merlin$_I$.

Saving

Saving was more problematic in Minitab$_t$ than it was in the other package treatments. Only 27.3% were able to successfully complete this task, while 71.1% were able to do so on the stand-alone Merlin$_t$ and 70% on the Extended Merlin$_t$.

Exiting

Minitab$_t$ had the lowest percentage of those successfully able to exit the program, 68.2%. On Merlin$_t$, this figure was 82.2% and on the Extended Merlin$_t$, 80.0%. Perhaps the fact that Merlin$_t$ allows many options for exiting accounts for this difference.

Table 56
Performance on sub-tasks by Package

	Merlin, N = 90						Minitab, N = 88						Extended Merlin, N = 50					
	Correct		Wrong		Not Attempted		Correct		Wrong		Not Attempted		Correct		Wrong		Not Attempted	
TASKS	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%
INITIAL																		
ENTER	67	74.4	21	23.3	2	2.2	65	73.9	17	19.3	6	6.8	41	82.0	9	18.0	0	0.0
NAME	56	62.2	16	17.8	18	20.0	42	47.7	20	22.7	26	29.5	33	66.0	8	16.0	9	18.0
EDIT	30	33.3	16	17.8	44	48.9	22	25.0	12	13.6	54	61.4	22	44.0	7	14.0	21	42.0
VIEW	60	66.7	7	7.8	23	26.6	45	51.1	20	22.7	23	26.1	26	52.0	5	10.0	19	38.0
SAVE	64	71.1	5	5.6	21	23.3	24	27.3	35	39.9	29	33.0	35	70.0	3	6.0	12	24.0
EXIT	74	62.2	4	4.4	12	13.3	60	68.2	0	0.0	22	25.0	40	80.0	1	2.0	9	18.0
RETEST																		
ENTER	32	91.4	3	8.6	0	0.0	46	95.8	2	4.2	0	0.0	42	89.4	5	10.6	0	0.0
NAME	31	88.6	2	5.7	2	5.7	36	75.0	9	18.8	3	6.3	39	83.0	5	10.6	3	6.4
EDIT	17	48.6	9	25.7	9	25.7	21	43.8	13	27.1	14	29.2	27	57.4	8	17.0	12	25.5
VIEW	30	85.7	1	2.9	4	11.4	39	81.3	6	12.5	3	6.3	28	59.6	3	6.4	16	34.0
SAVE	33	94.3	1	2.9	1	2.9	9	18.8	35	72.9	4	8.3	40	85.1	4	8.5	3	6.4
EXIT	34	97.1	0	0.0	1	2.9	44	91.7	0	0.0	4	8.3	38	80.9	3	6.4	6	12.8

5.B.9.b. **Correctly executed - Retest**

The problems with Changing were still evident on the retest, even though there were improvements. Still, less than 50% were able to accomplish this sub-task on either Minitab$_t$ - 43.8%, or Merlin$_t$ - 48.6%. On the Extended Merlin$_t$, 57.4% completed the task. On the retest, Saving was still a problem in Minitab$_t$, with only 18.8% completing the task, compared to 94.3% on Merlin$_t$ and 85.1% on Extended Merlin$_t$. For Extended Merlin$_t$, the problem area seemed to be in Viewing the data entered. Initially, 46% successfully completed this task; on the retest, the number rose to only 59.6%. A substantially greater percentage were now able to successfully enter data. On Merlin$_t$, the figure was 91.4%, on Minitab$_t$, 95.8%, and on the Extended Merlin$_t$, 89.4%. Merlin$_t$ showed the greatest improvement.

5.B.9.c. **Attempted, not successful - Initial Use**

On Merlin$_t$, the tasks which had the highest unsuccessfully completed figures were Entering - 29.2%, and Changing - 34.8%. Changing was also a problem on the Extended Merlin$_t$, with 34% trying but not succeeding. On the other hand, Saving was the major problem area for Minitab$_t$, with 42.7% not succeeding, followed by Data Entry, with 23.2% not completing the task.

5.B.9.d. **Attempted, not successful - Retest**

On the retest, Saving difficulties became even more exaggerated on Minitab$_t$; 72.9% could not accomplish it. Data Entry improved, however, with only 4.2% not achieving it. Changing became the next major area of difficulty, with 27.1% not being able to achieve it. Merlin$_t$'s problem area remained in Changing, with 25.7% not accomplishing the task, while Data

Entry improved, with only 8.6% not being able to do the task. Both Entering and Changing improved on Extended Merlin$_t$. Only 10.6% and 17%, respectively, could not successfully complete those tasks.

5.B.9.e. **Not attempted - Initial Use**

Changing was evidently the most difficult task on all the packages, with the highest percent never attempting it at all, at the initial testing. On Merlin$_t$, 42.7% never attempted this task; on Minitab$_t$, 64.6%; and on Extended Merlin$_t$, 42%. Viewing was also difficult in both of the Merlin$_t$ treatments. On the stand alone Merlin$_t$, 31.5% never attempted it, and neither did 46% on Extended Merlin$_t$. On Minitab$_t$, 46.3% never reached the stage of being able to attempt Exiting.

5.B.9.f. **Not attempted - Retest**

On the retest, on Merlin$_t$ and Minitab$_t$, Changing was still the task which had the highest percent of non-attempts, 25.7% and 29.2%, respectively. On Extended Merlin$_t$, this figure was 25.5%; however, Viewing was even more problematic, with 34% not attempting it.

5.B.9.g. **Errors by sub-task**

The average number of errors found in Treatment 1 and Treatment 2, on each of the sub-tasks successfully completed, was used to assess the differences in the difficulty with which it was possible to achieve each of the sub-tasks and, therefore, whether the scoring which was developed in Chapter 4.B.8 and used for our analyses was appropriate. The relative number of errors of those successfully completing the sub-tasks during testing were found to be:

	Merlin$_t$	**Minitab$_t$**
Average errors Overall	13.1	14.8
	———	———
Entry	22.2%	47.1%
Naming	16.0	17.3
Change	38.1	12.1
Viewing	15.1	12.7
Saving	3.0	9.0
Exiting	5.6	1.9

It would seem that entry and saving under Treatment 1 - Merlin$_t$ is easier than under Treatment 2 - Minitab$_t$, but changing and exiting gave rise to more errors than Minitab$_t$. The errors tended to revolve around not understanding what was required in assigning a filename. Taking the average of the two packages, in order to even out these differences, produces 34.9% in entry, 16.6% in naming, 13.9% in viewing, 6% in saving, 25% in changing, 3.7% in exiting. These figures are relatively close to those established conceptually. Therefore, the ones so established, as explained previously in Chapter 4.B.8., were considered appropriate for computing the scores for the sub-tasks.

Task	Scoring based on: Relative no. of errors during testing*	Conceptual difficulty of task**
Entry	34.9%	35%
Naming	16.6	15
Change	25.0	25
Viewing	13.9	10
Saving	6.0	10
Exiting	3.7	5

* Treatments 1 and 2, combined

** Scoring used in this study

5.B.9.h. **Conceptual hurdles**

A number of conceptual difficulties in performing the tasks were noted in the online traces collected from some of the subjects, who were almost exclusively in the Novice experience categories. These difficulties were noted, in varying degrees, for 80 of the subjects on Merlin$_I$, for 59 on Minitab$_I$, and for 48 on Extended Merlin$_I$. Although some of the conceptual problems were manifested in different ways in each of the three package treatments, because of the particular operational requirements of the packages, they pointed to the same underlying conceptual hurdles, all of which are pertinent in any package design, regardless of its state-of-the-art. A few of the students added comments to their questionnaires which support our observations of these conceptual hurdles garnered from the online protocols. A summary of these is included in Appendix L. Surprisingly few subjects took advantage of this opportunity which we offered. In fact, only one person made comments on Minitab$_I$; fifteen made comments on Merlin$_I$.

5.B.9.g.i. **Initial Use**

Saving

In Minitab$_I$, as noted previously, Saving proved to be a problem. The reasons for this were: 1) not recognizing the need to assign a filename to the data being saved; 2) not knowing the syntax for the command, which required **single** quotation marks around the name; and 3) trying to save one column at a time, but not knowing how to get it into the same file. Fortunately, Minitab$_I$ automatically assigns the filename 'Minitab' when one has not been furnished. However, this created problems for those who then wanted to see the file, but did not recognize that it had been saved under this name.

In Merlin$_I$, although there were fewer problems with executing the Save sub-task, a number of subjects, 24 to be exact, did question what the notion of "Save 'filename'" meant. Here again the issue of the quotation marks was evident. In Merlin$_I$, the quotations are not part of the syntax, but merely used to highlight the need for a filename.

For novice users especially, it is necessary to delineate when quotations form part of the syntax and when they are only used for highlighting. If they are there, users tend to assume that they are part of the syntax. Novices do not readily exclude them from the syntax, even after experiencing repeated failures using them. They assume there is something else wrong. Novices need an explanation of the concept of a file, and the syntax related to all commands, but in particular, to filename assignment, saving and retrieving, and also guidelines on what happens when it is saved using the same name twice. They also may need elementary computer literacy instruction, in order to understand the difference between changes made to the file in random access memory versus to secondary storage.

Command Names

A handful of subjects on Minitab$_t$ investigated the commands FILE and STORE as potential commands for Saving. There were two individuals who tried to use the INSERT command rather than READ or SET to enter the data. A sizeable number, 37 in all, sought help with the command TABLE for the same purpose. Compounding the tendency of users to conceptualize a dataset as a tableau, which is exploited in the design of spreadsheets, the Minitab$_t$ help screen, immediately following the command for help - HELP HELP, gives an example of HELP TABLE in the text presented to the user. In a statistical package, the TABLE command has a particular function, usually referring to contingency table analyses. This obviously represents a conceptual hurdle for first time users of software and users unfamiliar with Statistics.

In Merlin$_t$, there was also confusion regarding command names which seemed synonymous, FILE versus TERM versus SAVE, and NAME versus HEADING.

The conceptual difficulties experienced by these subjects suggest that, with regard to novice users, the challenge for designers of packages is to either: 1) ensure that command names are conceptually meaningful to users; 2) point out where conceptual discrepancies may arise, and explain the differences in meanings among the various terms; or 3) accept synonyms for the command names used in a package.

In and out of package

One approach noted in Merlin$_t$, but not in Minitab$_t$, was a tendency by some individuals to go in and out of the package, restarting at any indication of a problem. There were twenty individuals in $Merlin_t$ and five in Extended $Merlin_t$ who moved through the task in this way.

Autoformatting

Another conceptual hurdle, particular to $Merlin_t$, was the effect of autoformatting. $Merlin_t$ automatically formatted the numeric data input to two decimal places. A small number of subjects, four each on $Merlin_t$ and Extended $Merlin_t$, got bogged down trying to remove the decimals. Users need to be informed about how default values can be changed, and also how to separate the essential from the unessential in a task.

Accessing Help

A relatively large number, (considering that these subjects were predominantly novices, and especially, first time users), refused initial help and subsequent menu options, both on $Merlin_t$ and on Extended $Merlin_t$. These were individuals who either did very well or very poorly. On $Merlin_t$, eighteen refused initial help and twenty six refused subsequent menu prompts, such as to name columns and to print on the screen. On the Extended $Merlin_t$, six refused initial help and one refused subsequent menu options. Because of the mixed menu/command format of $Merlin_t$, users can conceivably attempt to perform the tasks without help. This is not possible in $Minitab_t$ because it is entirely command driven, so the user must know the commands before anything can be accomplished. This would explain why no one on $Minitab_t$ refused help.

The challenge for designers of a menu driven system is, therefore: 1) how to get novices to access help, either by prompting them to do so, or by automatically providing it; or 2) how to design the package so that no help is required, a formidable challenge.

Data Entry and Editing

While the task of basic data entry may seem trivial, there were a number of conceptual problems faced by the novice subjects here. Simple things, such as how many spaces to leave between data items, was an issue for some though the packages accommodated this. On $Merlin_t$, twenty-one felt it necessary to leave a large number of spaces between the entries, as did five on $Minitab_t$ and ten on Extended $Merlin_t$. A significant problem revolved around the concept of the separation of data and labels. This issue is particular to statistical packages which are designed around a line-editor concept, such as $Merlin_t$ and the version of $Minitab_t$ used in this experiment. The advent of full-screen editors overcame this problem, and while $Minitab_t$ does have this facility, subjects in the experiment were not permitted to use it. As a consequence, many subjects, while in data entry command mode, tried to enter the column headings before entering the data which is not permitted. There were twenty-nine subjects on $Merlin_t$ who tried this, twelve on $Minitab_t$, and twenty-one on Extended $Merlin_t$. The L2N and L3N subjects, novices with various packages which may or may not include one statistics packages, are perhaps influenced by prior exposure to spreadsheets which do not make the distinction between data and labels.

Once the data was entered, two on each package, attempted to change or remove the row enumeration which is automatically generated by the packages. One individual per package wanted to be able to enter each column of data across, row-wise and not column-wise. In $Merlin_t$, eleven tried unsuccessfully to enter the data one column at a time accompanied by eight in $Minitab_t$ and six in Extended $Merlin_t$. This action is permitted by each of the packages, but it would seem that how it can be accomplished has not been fully explained in any of the packages.

Other data entry difficulties were experienced by a few subjects attempting to access a file they believed already existed. Again, this may be the influence of spreadsheets which provide an empty worksheet to be filled. There were twelve on Merlin$_t$, two on Minitab$_t$, and three on Extended Merlin$_t$ who tried this. The concept of changing one element of the data table also proved a hurdle for some, eight on Merlin$_t$, and twelve on Extended Merlin$_t$. Users were unsure if one element could be changed or whether the whole row had to be changed. There was also difficulty in knowing whether it had to be deleted first and then the change made. A number of subjects had difficulty displaying their data on the screen, twelve on Merlin$_t$, one on Minitab$_t$, and six on Extended Merlin$_t$.

Mental Models

Some condidates displayed a lack of seeming logic or pattern in their path through the tasks. There were eight on Merlin$_t$, three on Extended Merlin$_t$ but none on Minitab$_t$. It is more difficult to detect this in Minitab$_t$, however, because those with difficulties may have simply read the voluminous help texts, without attempting anything.

Anxiety

Interestingly, two subjects demonstrated acute anxiety during the experiment, even though everyone was assured that their standing in the course was not dependent on performance on the experiment. In addition, participation was voluntary, counting for the remission of one assignment. One of these subjects left the room angrily, after shouting her frustrations at this author. No attempts to appease her succeeded, and she refused to leave her diskette and task sheet behind.

The other subject stayed the duration, but also vehemently expressed her frustrations. She was incredulous that she should have had such difficulty, given that she uses a computer regularly in her work. On further investigation, this experience proved to be with canned packages, requiring menu selection, rather than with packages that allow the design and analysis of problems. Evidently, the type of prior experience is an important factor in the performance with and perceptions of a new package which our results also confirmed.

5.B.9.g.ii. **Retest**

The number of subjects experiencing conceptual hurdles was dramatically reduced on the retest a week later, to twenty on Merlin$_t$, eight on Minitab$_t$, and seven on Extended Merlin$_t$. There were six subjects on Merlin$_t$ and one on Minitab$_t$ who still inserted several spaces between data elements. There were seven on Merlin$_t$ and two on Extended Merlin$_t$ who still did not seek initial help; and seven on Merlin$_t$ and five on Extended Merlin$_t$ refused the print and naming menu options. On Merlin$_t$, three persisted in trying to access a file they assumed already existed.

There were four on Minitab$_t$ who still did not appreciate the separation of data and labels; and two sought help for the command TABLE. There were two who tried to change the decimal formatting in Merlin$_t$; and one on Minitab$_t$ tried to change the labels of the automatic row designations supplied by the package. One person each on Merlin$_t$ and Extended Merlin$_t$ still went in and out of the package. There were five on Merlin$_t$ and three on Extended Merlin$_t$ who questioned the meaning of 'filename'. There were three on Minitab$_t$ who saved without a filename; and one individual equated STORE with saving. There were four on Merlin$_t$ who entered the columns one at a time.

It should be noted that Merlin$_t$ does say, in its introductory screen, that data items are separated from data labels. The fact that the users of Merlin$_t$ did not make note of this, points to the necessity of highlighting crucial pieces of information within the text. Computerized Instruction Design guidelines suggest that screen displays present only one idea per screen, and be free from other material which is not required for imparting the particular idea being conveyed. Wording is also very important. Meanings and symbols may not always be self-evident, as indicated by several experienced users asking what '<cr>' meant, that is, carriage return. Another useful recommendation from this area of study includes monitoring performance online and presenting remedial material to rectify difficulties. Those using the Extended Merlin$_t$ system should have had no difficulties if they accessed the 'examples' option on the menu bar, as it provides a complete example of the data entry task. If they followed the menus they should also have had no difficulty; but once off the track, it may not be evident how to return.

First time users of Minitab$_t$ could only have succeeded if they happened by chance onto the correct help menu for the data entry task, which was 'Help Overview 1'. The 'Help Commands 2' option, under which is found the Input and Output commands, gives a list of command names, but not their syntax or context, which unfortunately was where most users expected to find it.

This chapter presented the results of our two studies. We investigated: i) expert opinion on the design and assistance features affecting **ease of use**; ii) the effect on perceived comfort and performance measures relating to **ease of use** of various factors, such as package design, experience levels, gender, quantitative competence and computer anxiety, for users in actual use of three different package treatments; iii) In addition, the performance results on each of the subtasks, and the conceptual hurdles faced by novice categories of users, were discussed in this section. The chapter which follows discusses the implications of these findings for academia and business. We also discuss the limitations of the current study and make suggestions for future research directions.

Chapter 6 - DISCUSSION AND IMPLICATIONS OF THE FINDINGS

In this chapter, we discuss the findings from our two studies and the implications they hold for software design and for the business and academic communities. The limitations of these studies are also examined, and areas for improvements and future research are identified. The discussion takes place in the following manner:

6.A. Findings on Factors affecting **Ease of Use** of Business Software
6.B. Challenges and Recommendations for design arising from findings
6.C. Contributions of the Research to Business and Academia
6.D. Limitations of the Research
6.E. Ways in which Research of this type could be Improved
6.F. Directions for Future Research

6.A. Findings on Factors affecting Ease of Use of Business Software

The two studies we conducted succeeded in giving us some interesting insights into the factors facilitating and detracting from the **ease of use** of business software. These are of potential assistance to researchers, to software designers and to trainers of users, and to those who must evaluate packages to be purchased, whether for personal use or for use by others.

First of all, our findings did not altogether support our assumption of the learning dimensions, **Speed, Memory, Effort and Comfort**, being components of **Ease of Use**. In Study 1, the dimensions seemed to be equally important to our panel of experts. In Study 2, we found that although when more features were perceived to be supporting these dimensions, performance and perceptions were better, as we had expected, the package with the better performance scores (**speed**) did not have the better help and error scores (**memory, effort**), or perceived comfort scores (**comfort**). With use, this package was better on all scores, though the perceived comfort scores were still not statistically significant. The findings do suggest, however, that further research is warranted. There is some evidence that the support given by package features to the learning dimensions, **speed, memory, effort** and **comfort** may be important for **ease of use**.

We found in Study 1 that expert designers agreed to some degree on individual design and assistance features which are important for **ease of use** of a package. The most important identified in terms of Design Features were Command Structure, Consistency and Logic of the Structure; in terms of Assistance Features, they were Manuals, System Messages and Prompts. The agreement was low, however, on the rankings assigned to the features as a whole.

The panel did not totally agree on which dimensions were supported by each of the features, however, though there were some underlying themes noted. Effort was more likely to be associated with Design Features, and Memory, least likely, while Assistance Features were more likely to be associated with Effort and Comfort, and least likely with Speed.

We found that the predictive power of the variables examined in Study 2 was low, nonetheless, the statistical significance of the variables suggests that some plausible relationships may exist and warrant further research. The low predictive value of the models is not surprising given the exploratory nature of the study which has the potential for introducing a high degree of error and the fact that the construct and its components may not be complete nor fully developed. Other variables, as yet unidentified, may add to the predictive power of the models.

In terms of predicting the performance time measure relating to **ease of use, (speed),** we found that all the variables, except perceived quantitative competence and extent of microcomputer experience which could not be modeled owing to the small sample size, seemed to have a significant impact. Prior experience with statistics packages seemed to be the most significant, followed by package design differences. In particular, the mixed menu/command with concise help design seemed to contribute to improved performance. Computer anxiety and gender also seemed to be reasonably significant predictors. Perceived quantitative competence and package differences seemed to be predictive of the performance error (**memory/effort**) and perceived comfort (**comfort**) measures relating to **ease of use.** Poor quantitative competence and the mixed menu/command system seemed to contribute to more errors/score and the expanded help on the command system seemed to contribute to better perceived comfort. Perceived quantitative competence also seemed to account for differences in the performance help measure (**memory/effort**) relating to **ease of use**. These variables are not sufficient, however, to account for all of the variability in the results found. Factors, such as other User Characteristics - Psychostructure makeup and Microcomputer Package Experience Level, and others, may have greater influence on **ease of use**. The individual variables we identified, nonetheless, do provide some insights into **ease of use.**

We found that **ease of use** may be dependent on the command structure and online assistance provided, given the similarities noted in the requirements of the packages to complete the experimental task. In particular, a mixed menu/command structure, designed for speed and with menus organized around the usual task steps, and with help succinctly displayed on two screens, produced better performance time indices than the strictly command-driven structure with full online manual. On the retest, this system also showed greater improvement than the latter, thereby suggesting that more than just the time required to read the verbose help screen in the command system hampered performance with it.

The supplementary assistance provided by a Hypertext-based help index to the stand-alone mixed menu/command system did not improve performance. This may have been due to the additional reading required, or the need to move between keyboards, and the fact that the PS2's used by some subjects on the standalone system are faster than the Olivetti's used by the subjects on the stand alone plus the expanded help system.

The above results, however, were reversed when perceived comfort was the dependent variable. Notwithstanding the better performance time indices with the mixed menu/command structure with concise online help, somewhat lower perceived comfort ratings were given to it than to the command-driven structure with full online manual. Although the concise help may have assisted performance speed, it apparently did not contribute to a feeling of security or comfort. There were also some difficulties experienced with the syntax requirements of the former system, which may have frustrated users. The mixed menu/command system also required a greater conceptual leap to execute editing. It was necessary to change from command mode to edit mode and there were more edit options to differentiate. It would seem that users feel more comfortable when they have access to expanded help. This was further supported by the somewhat better perceived liking ratings when the expanded Hypertext-based help index is added to the mixed-menu command system which was the only thing differentiating these two systems.

The efficacy of a mixed menu/command system, with or without expanded help, is also seen in the error and help performance. Although more errors/score and help calls/score were made initially, probably due to attempting more, these were significantly lower on the retest compared to the strictly command-driven system with full online manual.

One of the interesting findings of the study was that experience level may have a mediating effect on performance and perceived comfort scores relating to **ease of use**. As such, although the task was simple data entry and, therefore, not particular to the package class we investigated, namely, Statistics Packages, those who had previous experience with statistics packages did perform better than those who did not. Because of the distinct separation of Input/Process/Output in statistics packages, the 'look and feel' of these packages is different from that of applications such as wordprocessors, spreadsheets. According to the results we obtained, it seems that prior exposure to this 'look and feel' facilitates use and learning. In addition, those using Merlin$_t$, who had previous experience with Minitab$_t$, rated Merlin$_t$ lower than did those who had not had this exposure. These individuals with prior experience with Minitab$_t$ using Minitab$_t$, in turn, gave better ratings to it than those without this prior experience.

In effect, we found that prior experience with packages of any type, whether of similar or dissimilar function, produced better performance scores than when experience was absent. This was also confirmed by the significant improvement in performance and perceived comfort results from initial use to retest, on the same package. Use and experience, therefore, seem to improve both performance and perceived comfort and this seems to be in proportion to the amount and level of experience possessed.

Our results highlighted the extent to which experience seems to impact **ease of use** in terms of performance. Compared to the Benchmark Index which was estimated to be 3333 on both packages (see Chapter 4.B.8.), that is, completion of 100% of the task error-free in 3 minutes, novices took as much as one hour, sometimes without accomplishing any of the task. The more experienced users, who did manage to complete all of the task, were only able to do so in 20 minutes, the shortest time recorded. These were the Professors of Statistics, who obtained a Time Index of 645 on Merlin$_t$. Excluding this category, the next highest Index obtained on Merlin$_t$ was 444, which rose to 1050 on the retest. On Minitab$_t$, this figure was 363, which rose to 991 on the retest. None of these Indices is anywhere close to the estimated Benchmark Index of 3333. Nonetheless, using this benchmark as our anchor for relative **ease of use** as suggested in Chapter 3, we may say that Merlin$_t$, the mixed menu/command structure with concise online help, is easier to use and learn than Minitab$_t$, the command structure with full online manual, having indices closer to the Benchmark, both initially and on the retest.

Ease of use for novices also seems to be a function of certain conceptual hurdles, namely, command names, the notion of separation of data and labels, the notion of a file, and the saving and retrieving of it. This category of user also experienced more difficulty in coping with the addition of the Hypertext Index than did the other levels. They also made more errors and required more assistance from help. This was also true of those without prior statistics package experience.

We found that the performance and perceived comfort measures relating to **ease of use** seem to be positively related to each other. Although, in general, perceived comfort improved as performance improved, when we compared our package treatments, we found that **ease of use** in terms of performance was not congruent with the perceptions of our user groups. The superior package in terms of performance was given lower perceived comfort ratings, on average, than the other package treatments.

Learning, as measured by retention, which we view as a subset of **ease of use**, would seem to be a function of repeated use. On second exposure to the same package and task after seven days, users' performance and perceptions improved dramatically. This would also seem to be a function of experience level.

It would also seem that anxiety, gender and perceived quantitative competence affected **ease of use**. Higher anxiety levels had adverse effects on performance and lower perceived quantitative competence adversely affected both performance and perceived comfort. Males also seemed to perform better than females. A number of reasons may account for this finding other than gender per se. We already noted that factors, such as parent's income and educational level, exposure to home computer and video arcade games, may confound this type of analysis. It, therefore, must be interpreted judiciously.

In Table 57, following, we summarize the findings from the two studies of the factors which seem to have the potential for facilitating and detracting from **ease of use** as we have been discussing. The reader is reminded that these variables were found to have low predictive value. The results support the guidelines proposed for user-friendly software by The Applied Statistics Research Unit at the University of Kent (Porter, 1993, p. 221) which are:

"-include help facilities;
-use defaults to lead the inexperienced user to a valid analysis;
-provide the options required by the more sophisticated user;
-use clear unambiguous menus and single-key responses;
-support with good (non-jargon) documentation."

Both our studies lend some level of support for the importance of Command Structure and Manuals for **ease of use**, as has also been shown in prior experimental/micro research efforts. (Holcomb and Tharp, 1991; Burns, et.al., 1986; Lee, et.al., 1986; Carroll, 1985; Dunsmore, 1980; Relles, 1979). Our finding of the seeming superiority of a mixed structure over a command structure agreed with Lee, et.al.'s (1986) findings with respect to performance but not perceptions and with those of Teng and Jamison (1990) but not with Khalifa (1990). Our findings with respect to the concise online help seeming to contribute to better performance agrees with Carroll (1985), Dunsmore (1980) and Relles (1979) but not with those found by Ogden, et.al., reported in Paap and Roske-Hofstrand (1988), Whiteside, et.al. (1985), Hauptmann and Green (1983). The reduction in errors attributed to the additional assistance provided by the hypertext-based index agreed with the findings made by Burns, et.al. (1986) regarding improvements made to a system which improved performance.

We also identified a number of conceptual hurdles which corresponded with topics investigated in prior research. These included command naming, syntax of commands, resistance to help, mental models, anxiety and experience. (It should be noted that while some prior research has focused on mental models, and our Study 2 supports this, our experts in Study 1 did not consider this a priority. Perhaps this is because they already have a mental model of some packages.). We also identified other conceptual hurdles not hitherto alluded to in research, such as the propensity for some users to start and restart the package at the first indication of difficulties, the inability to differentiate the need, or not, for changing automatically formatted data, the inability to distinguish between data and labels and how many spaces are required between data entries of a row of data, and the lack of understanding of what a file is and how and where it is saved and retrieved, revealing a need for understanding of random access memory and secondary storage.

We found that experience with packages of similar and dissimilar function seemed to have a statistically significant impact on measures relating to **ease of use** which is consistent with some prior research (Nielsen, 1992; Foss and DeRidder, 1989; Roberts and Moran, 1983; Roth, Bennett and Woods, 1987; Karat, et.al., 1986; Hauptmann and Green, 1983). The extent of this experience seemed to result in performance differences but the sample size was small. This was consistent with the results found by Gugerty and Olson (1986) with respect to programming.

When comparing the packages, our findings seemed to be contradictory with respect to performance and perceived comfort. This was consistent with Roberts and Moran (1983) who found that none of the editors in their study had superior scores on all measures. The same was true in the study by Teng and Jamison (1990).

Our results on perceived quantitative competence affecting measures relating to **ease of use** contradicted those of Evan and Simkin (1989) and, instead, confirmed conventional wisdom, as did our results with respect to computer anxiety. Our finding on gender would seem to support common belief in the male's aptitude for more quantitatively- and mechanically-based tasks, in contrast to Murphy's (1992) finding. However, the reader has already been cautioned about making too facile an interpretation of this result. For one thing, the difference was marginal. It does, however, suggest a potential research area whereby to determine the underlying causes which may account for this result. Possible causes could be less exposure to video and home-computer games, and family traditions and occupations.

Notwithstanding the low predictive power of our variables, our findings on the variables which may possibly influence our measures relating to **ease of use** can be summarized as follows:

Objective - measures	TIDX function (Package, Gender, Anxiety, Package Type Experience) TIDX function (Speed, Memory, Effort, Comfort support) EIDX function (Package, Quantitative Competence) HIDX function (Quantitative Competence)
Subjective - measures	LIKE function (TIDX, Package, Quantitative Competence) LIKE function (Speed, Memory, Effort, Comfort support)

The challenges and recommendations arising from these findings for software design are discussed next.

Table 57
Summary of Findings

Facilitators of Ease of Use	Detractors from Ease of Use
Performance	
Package Design - mixed structure (Mer, Mer+ > Min) (novices on Mer > on Min) - concise help (Mer > Min, Mer+)	Package Design - conceptual hurdles
Learning Dimensions - support of SMEC perceived	Learning Dimensions - support for SMEC not perceived
Experience/Use - Retest > Initial - prior experience with packages (L4 > L3 > L2 > L1 - experience in general (experts > intermediates > novices (Benchmark > highest indices in sample)	Experience/Use - no experience (L1 < all levels) (Novices < Intermediates, Experts)
Gender - males > females	Anxiety - negatively related to performance
Perceived Comfort	
Package Design - command structure (Min > Mer, Mer+) - expanded help (Min, Mer+ > Mer)	Package Design - conceptual hurdles
Learning Dimensions - support for SMEC perceived	Learning Dimensions - support for SMEC not perceived
Experience/Use - Retest > Initial - experience with packages of similar function (L3 on Min > L3 on Mer)	Experience/Use - no experience (L1 < others) - experience with packages of similar function (L3 on Mer < L2, L1 on Mer) (L3 on Mer < L3 on Min)
Quantitative Competence - Excellent > Average, Poor	

Note: L3's have experience with one statistics package which is Minitab
> refers to better performance and perceived comfort ratings
SMEC refers to Speed, Memory, Effort, Comfort

6.B. Challenges and Recommendations for design arising from findings

We found that significant prior exposure to packages of similar function seems to have a positive impact on performance with an unfamiliar package, but a negative impact on perceived comfort. However, in general, there seems to be a positive association between performance and perceived comfort. The challenge, therefore, for the software designer is to take into account, and implement some or all of the following strategies:

1) Create a favourable first impression, by way of the assistance provided, et cetera, so as not to hinder future use;

2) Ensure that the functionality is such that the user will be willing to invest the time to overcome the initial learning hurdles, if the structure is appreciably different from the one with which the user is already accustomed;

3) Standardize on a structure similar to the ones known by most users;

4) Design the structure so that it can adapt to the users' preferences;

5) Design the structure so that it is entirely transparent to the user and does not require learning.

It was already noted that, in designing packages for initial use by novice users, the challenge for designers is, first of all, to benefit from the seeming efficacy of a mixed menu/command structure over a command structure, and a concise online help system over a full online manual or even, it seems, a Hypertext-based online index with explanations and examples. The findings suggest that for subsequent use, the hypertext index becomes more beneficial than the concise online help system alone. Secondly, designers are faced with questions of:

1) how to provide help that novices will access;

2) how to analyze performance and to provide context-based help;

3) how to design the system such that none is needed.

Novices also need initial assistance with such concepts as the separation of data items and data labels, the notion of files, fields, and primary and secondary storage. They also need assistance with the format of data entries and the differences imposed by data entry versus data editing, command syntax conventions, and procedures for changing defaults.

For all categories of users, the issue of synonymous command names is relevant. Designers must decide:

1) whether to point out potential conceptual discrepancies in command names and to explain them;

2) how to make command names conceptually meaningful for most categories of users;

3) whether and how to accommodate synonyms to command names.

The alleviation of anxiety is also important. The challenge for designers, which may achieve this, is:

1) how to make interfaces that are 'easy to use';

2) how to design adaptable interfaces which can conform to the users needs and preferences;

3) whether designs should, in fact, be adaptable or should they be standardized within and across applications.

The findings with respect to anxiety, gender and quantitative competence also suggest that consideration of these should be made when designing systems. In what ways would have to be further researched. Our findings with respect to the learning dimensions which we identified as being components of **ease of use**, namely, **Speed**, **Memory**, **Effort** and **Comfort**, also suggest that support of these be considered in design.

In general, based on the findings in Studies 1 and 2 on performance and perceived comfort and conceptual hurdles faced by our users, design and assistance features should perhaps be such that they support Speed, Memory, Effort and Comfort. It would seem that a mixed menu/command structure is preferable to a strictly command-driven system. The system could further be enhanced by permitting the use of synonyms for commands or by clarifying potential confusion. It should be consistent, flexible and logical from the point of view of the user, according to our expert panel in Study 1. These suggest contradictory solutions to design, on the one hand, standardization, and on the other, adaptable interfaces. This is not easy to resolve in design.

A combination of both expanded and concise online help seems to be indicated, though initially for complete novices, the concise help alone seems to be more productive. This documentation should clearly explain syntax conventions. It also needs to provide novices with orientation modules to explain such concepts as files, primary and secondary storage, the separation of data and labels and how these relate to fields, and the distinctions of input/process/output. The documentation should also make clear how defaults may be altered, files saved and retrieved, and any differences required for data input versus editing. Context help, examples and system messages may further facilitate **ease of use**. The need to prompt users to access help is also indicated, perhaps after an extended period of inaction or error state. Instructional material to compensate for disadvantages arising from poor quantitative competence, high anxiety with computers and gender are also indicated though how is not yet clear.

In addition to these support mechanisms, the user should be provided the opportunity for use. Also, exposure to other packages, whether of similar or dissimilar function seems to contribute to alleviating some of the disadvantages wrought by deficiencies in the above factors. Experience can help to compensate for much in terms of both performance and perceptions.

Based on the conduct and results of our studies, a method for evaluating the **ease of use** of software may be suggested. To assist in managing the complexity involved in comparing the **ease of use** of package designs, and in order to capture the richness of the concept and to develop a deeper understanding of it, we propose that a number of steps be adopted, which are pertinent in both academic and business contexts. These steps are as follows:

1. Determine the important design and assistance features required of a package, based on particular user needs and/or preferences, or on established standards, depending on the context and desirability.
2. Identify the learning dimensions considered to be important to be supported by the features in the package, in order to facilitate **ease of use** and accelerate learning. From our studies, support of Speed, Memory, Effort and Comfort seem to have some importance for **ease of use**.
3. Determine the benchmark task(s) to be used for the basis of evaluation.

4. Evaluate the package(s) based on the criteria above and the extent to which support is given to the learning dimensions. In some contexts, the evaluations may require separate assessments of the Input/Process/Output functions of the package(s).

5. If appropriate and possible, validate the evaluations with user performance and perceptions using the package(s) and, in the design context, making use of online protocols to support the results. Preferably, these protocols should include time stamps between keystrokes to permit fine-grained analyses, which are particularly important in research situations. They are a rich source of information from which to unearth users' mental models and sources of obstacles to learning, with the potential to provide untold research discoveries.

6. In the design context, refine those features for which improvement is indicated from the findings above.

7. In the business context, choose the package which best meets the criteria established in Steps 1 and 2.

An Evaluation Form could be derived based on this, which, it is suggested, may prove to be useful both in industry and for academic research purposes. The Form can perhaps provide a more structured medium through which to make evaluations of package designs, without diminishing the qualitative nature of evaluation. It provides for a multi-dimensional assessment of the relative quality of the numerous features available in a package which is based in learning theory, and which expands on the learning dimensions identified by other authors to include a Comfort dimension. The delineation of Design and Assistance Features, as we propose them, is also novel. The Form may have the potential for being a useful basis for the design and evaluation of the various features in a package, but evidently more research is required. For instance, a low rating of a particular feature on these dimensions may help in pin-pointing the exact area which is in need of being rectified. For example, a low rating on user performance speed or effort may indicate that too many operations or too many levels of navigation are required to perform a task. A low performance speed rating may also be indicative of a design too burdensome for the level of technology used. A low rating on memory may indicate that help on that feature is not adequate. A low rating on comfort may be indicative of poor screen design, cumbersome implementation of the feature, difficulty of use or differences in conceptual perceptions of the feature and its use.

Another advantage of a tool such as this is that it is 'easy to use', is not time or effort intensive, and would not require any technical expertise. An example of what this form might look like and how it might be used is shown in Appendix E. Appendix I also gives an example of how it might be adapted for a particular package function, in this case, a statistical package.

The Form can provide not only a tool for evaluation, but also a medium for research. It was noted that experience with packages of similar and dissimilar function and/or operation was found, in our second study, to possibly mediate performance and perceptions. Hence, it is suggested that the validity of the Form could be investigated with respect to these experience level differences. Likewise, further research is warranted on the learning dimensions which we identified. The constructs of **Speed, Memory, Effort** and **Comfort** need to be more clearly defined, and methods for their measurement derived.

None of these recommendations is easy to achieve. We must, first of all, understand what facilitates and what deters from the ease of use and learning of software initially and over time. Even with experience with varied packages, each time a person must resume work on a different package, there will, initially, be a deterioration in performance. (Argote, 1990; Bailey, 1989). The challenge for the designer is finding how to minimize this deterioration and accelerate re-adaptation. A number of researchers are investigating some of these strategies, such as the paper-like interface proposed by Wolf, et. al., 1989; and adaptable interfaces proposed by Vaubel and Gettys, 1990; Mahling and Lefkowitz, 1989; Kantorowitz and Sudarsky, 1989; Maskery, 1985; and Good, et. al., 1984.

The issues of design raised in this thesis, while they may have emerged from a traditional IBM PC type command environment, are perhaps even more relevant to the Windows or icon-based environments. In these latter cases, one picture or one word must convey the full meaning of a command. We noted in the literature survey that users do not always agree on the meanings conveyed by commands. (Landauer, et. al., 1983). The number of windows that the human brain can comfortably and efficiently cope with on the screen, whether to tile or to overlay screens, will have an impact on, and offer different support for, the speed, memory, effort and comfort of the user.

The same issues of consistency across packages are pertinent in recent designs. Menu structures, command names, et cetera, must still be adapted to when switching between applications. In the MacIntosh environment, and now in the IBM Windows environment, there has been standardization across applications, which brings forward another issue of vital consideration. Should designers impose a standard, or seek ways in which users can interact in their preferred ways? Are the standards developed necessarily the best? The QWERTY keyboard is a case in point of how optimal evolution can be stymied as a consequence of users being entrenched in a standard.

The contributions made by our research to our understanding of **ease of use** are discussed next.

6.C. Contributions of the Research to Business and Academia

In sharpening our understanding of the factors which influence the **ease of use** of business software, this thesis contributes, actually and potentially, to the study and evaluation of business software in academia and business. It does so in several ways and on many levels.

Conceptual Contributions

1) The thesis integrates, for the first time, the two major bodies of literature in Learning and Human Factors.

Implications: This permitted us to elaborate on the many and complex factors impacting on **ease of use**. The combined literature also suggests new areas for research, and new questions and possible solutions to the issue of **ease of use**. It also provides a basis on which to develop an annotative bibliography.

2) For the first time, the many and complex factors impacting on **ease of use** are presented in the context of a comprehensive framework.

Implications: The framework is useful to researchers as a context in which to identify and explain the components of their research designs. This, in turn, permits a common basis for comparison across research efforts. The framework is also useful for identifying weak areas of research and for generating research questions accordingly.

3) For the first time, **ease of use** is suggested to be a multidimensional construct which is measurable by the support given to its components speed, memory, effort and comfort. Prior research has tended to view **ease of use** and various aspects of the learning dimensions we identify, as features of the package and not outcomes of the features (Karat, et.al., 1992).

Implications: This approach suggests a more complete view of **ease of use** than that usually adopted in the literature (Karat, et.al., 1992; Khalifa, 1990; Roberts & Moran, 1983; Card, Moran & Newell, 1980). It suggests research into new ways of measuring and assessing ease of use.

Methodological Contributions

4) For the first time, an exploratory/holistic approach was applied to research hitherto studied under experimental/micro conditions.

Implications: A number of factors impacting **ease of use** could be identified and assessed in one research effort. It provided a basis for validating the generalizability of previous research findings from micro studies to a more natural and realistic setting. It provides a basis for generating new research questions.

5) For the first time, a field survey of **ease of use** was conducted which went beyond a uni-dimensional assessment usually employed in research to incorporate consideration of the learning dimensions, speed, memory, effort and comfort.

6) For the first time, data on perceived support offered by package features to the learning dimensions was examined.

7) For the first time, a delineation of Design and Assistance features is made based on those features forming an integral part of the software design (Design) and those which can be added on (Assistance). These latter can often be either internal or external to the package itself. In current research, this distinction is not formally articulated and online assistance is often viewed as one all-inclusive feature.

Implications: These provided a basis for validating our belief in the usefulness of support for the learning dimensions offered by package features for **ease of use**. The results of the two studies, though somewhat tenuous, could lay the foundation for the development of a potential evaluation form designed for a multi-dimensional assessment of **ease of use** in contrast to the uni-dimensional approach used in previous research (Holcomb & Tharp, 1991; Scrivens, 1990; Teng & Jamison, 1990; Davis, 1989; Davis, Bagozzi & Warshaw, 1989; Doll & Torkzadeh, 1988; Roberts & Moran, 1983). Our studies also provide a basis for generating new research questions on ease of use.

8) For the first time, data on experience with packages of similar and dissimilar function and the extent of this experience were examined in one study.

Implications: This generates questions and research ideas on appropriate methods of assessing experience levels.

9) We proposed a method of measuring performance time, errors and help which accounts for the holistic nature of the research which makes exact measurement more difficult than in experimental research (Murphy, 1990; Khalifa, 1990; Roberts & Moran, 1983; Card, Moran & Newell, 1980).

Implications: This raises questions on methods of measuring the various components of **ease of use**.

10) We compared a command structure to a directed and not an undirected menu (Whiteside, et.al., 1985; Hauptmann & Green, 1983).

Implications: This comparison is more representative of real differences between command driven structures and menu structures since undirected menus can be viewed as simply another organization of commands.

Empirical Contributions

11) The thesis has added to the store of knowledge on the factors affecting **ease of use**.

a) Some degree of support was found, in both studies, for the importance of command structures, manuals and of the learning dimensions for **ease of use**. The findings on the importance of command structures and manuals were consistent with previous research. (Holcomb & Tharp, 1991; Burns, et.al., 1986; Lee, et.al., 1986; Carroll, 1985; Dunsmore, 1980; Relles, 1979)

b) We found that experience with packages of similar and dissimilar function seemed to have a significant impact on measures relating to **ease of use** which is consistent with some prior research (Nielson, 1992; Foss & DeRidder, 1989). The extent of this experience was not a significant factor overall compared to the other factors, though it did result in differences in performance, which is consistent with some prior research (Foss and DeRidder, 1989) and not others (Khalifa, 1990; Karat, et.al., 1986; Whiteside, et.al., 1985; Ogden, et.al., 1988; Hauptmann & Green, 1983).

c) We found that computer anxiety, gender and perceived quantitative competence seemed to have an effect on measures relating to **ease of use**. These have not been previously investigated in the same study. Previous research has also been more concerned with the factors affecting computer anxiety than the effect computer anxiety has on performance and perceptions.

d) We identified important conceptual hurdles faced by novice users which were consistent with the streams of previous research and our findings suggest new areas.

e) We seemed to have found contradictions between objective performance and perceived comfort, in comparing packages, which is consistent with the results of other research (Roberts & Moran, 1983).

Implications: The results support the validity of a holistic approach for studying **ease of use**. The consistency of the findings with research conducted at a micro level suggests the measures used are reasonably valid for assessing **ease of use**, but need further research. The findings support the relevance of the current streams of research in command structures, manuals, experience, command naming, syntax of commands, mental models and anxiety. They also suggest new areas in need of investigation, (importance of prompts, impact of conceptual hurdles relating to concepts of file, distinction between data items and labels, space requirements for data entries, random access memory, starting and restarting the package, dealing with autoformatting, resistance to assistance, anxiety, gender, quantitative competence and other factors) and current areas not seemingly as important (conceptual models).

Notwithstanding these contributions, there were some limitations to the study, to be discussed next.

6.D. Limitations of the Research

No research endeavour is without its limitations, and there are a number of criticisms which could be levelled at this one. With the advent of virtual reality and other technological innovations, some may question the validity or usefulness of having used packages in Study 2 which can no longer be considered state-of-the-art, and may suggest that the features identified in Study 1 are not representative of emerging technologies. Our response to these criticisms is that both studies were put forward as an approach to assessing **ease of use**, and not as ends in themselves. Nonetheless, the features and learning dimensions identified in Study 1, we feel, are relevant regardless of the technology on which a package design is based. The implementation of the features may be different depending on the state-of-the-art of the technology being used, but the features and learning dimensions themselves are still pertinent bases for assessing **ease of use**. Again, in Study 2, the two packages chosen were selected for research convenience, primarily because they afforded examination of differing package and assistance designs and because they were available on the students' computer network at Concordia University, affording the collection of a large data set. Study 2 was not meant to be so much a comparison of two packages as to demonstrate how one would, in an exploratory context, approach assessing two packages of similar function for their **ease of use**, regardless of the particular package or its state-of-the-art. Even though both packages can be considered comparable in their state-of-the-art, nonetheless, the approach adopted in our study was able to differentiate performance and perceptions of the packages, and how these are mediated by experience levels and other factors. One should bear in mind too that the implications, for instance, of command naming as discussed previously, are even more relevant in the Windows environment, where one word often has to convey the full meaning of the command or operation.

Our method of assessing **ease of use** is meant to be generic and, therefore, applicable to all technology levels. The features and learning dimensions which we suggest in Study 1 are not necessarily meant to be all-inclusive or unchangeable. These may be based on personal preferences or needs, or established standards. Evaluation of performance and perceptions is a necessary component of any assessment, regardless of the package's state-of-the-art. Our suggestion of confirming the performance and perception results with online protocols is particularly pertinent to technologically advanced designs. We would recommend that all packages be designed with this facility, including time-stamps between keystrokes, which is not currently the case with most packages. (Merlin, provides this feature). These could be used, not only for assessing areas of design deficiency, but also to provide users with a looking-back trace in order to orient them to their place in the package, as well as as a learning tool for self-assessing correct and incorrect actions.

Some questions may arise concerning the choice of task for Study 2, which was simple data entry and, therefore, did not explore the functions for which a statistical package is designed. First of all, data entry is common to all packages and is the first task required to make use of any package. This is where initial perceptions of the package are formed. The data entry task in line-editor-based statistical packages is very different from that in packages such as spreadsheets and wordprocessors. In order to be able to compare our sample subjects on a common basis, the task had to be such that novices could conceivably master it. To further impose the requirement of subject matter knowledge, that is, statistical knowledge, would have confounded the results. The effect of subject matter experience on **ease of use** is, however, an interesting topic in itself.

Time frame was also a consideration. In pretesting, it was found that MBA students in an Introductory Statistics course were unable to accomplish the task of entering data and running a small regression analysis within the one-hour time frame. It was considered important that the task time not exceed one hour, in order to permit the participants time to fill in the Proficiency and Perception questionnaires without feeling pressured. It is worth noting that even with a 'simple' data entry task, the majority of 243 participants did not complete all six sub-tasks, let alone do so successfully. In fact, compared to the estimated Expert Benchmark of performance on this task, these subjects have a long way to go to reach mastery. The estimated Expert Benchmark Time Indices for both Minitab$_t$ and Merlin$_t$ were established at 3333, as noted earlier. This could be achieved by completing the task, on each of the packages, in three minutes without any errors or help calls. The highest score actually obtained by one of our participants was 585, and the highest on the retest was 1100. This is a finding which should be reflected on by those teaching Statistics, and who include computer analyses as part of course requirements. If simply mastering data entry of 20 observations can take anywhere from twenty minutes to an hour, how much longer must it take to master statistical techniques and understand the output? Perhaps more sensitivity to students' complaints is in order, and perhaps more care should be taken in the selection of the software to accompany these computer-calculation-intensive courses. Such software is evidently not as 'simple' as we assume it to be, even for experienced users. These results suggests that **ease of use** is not absolute and so establishing an 'ideal' measure of **ease of use** may, perhaps, not be possible.

The fact that Merlin$_t$'s menu structure leads users through the steps involved in the task of data entry could be considered a source of bias. However, someone looking to choose among packages is going to select one based on what exists in the package, and is not aiming to find packages which are similar in design for comparison. Merlin$_t$'s type of design may, in fact, be considered a design feature. A package designed around functional steps is likely to be beneficial, taking account as it does of the way in which the user works. Interestingly, our findings did not support this, in that users did not give higher subjective ratings to Merlin$_t$, and, in fact, many novices did not make use of the Merlin$_t$ prompts for printing and adding labels which were presented to them. We also found it necessary to delineate the steps required in the task, so that first time statistics package users would not be disadvantaged as a result of not knowing what was required in a data entry task. This necessity became evident in the pretests.

We assumed that the difference seen between the package treatments was attributable to differences in the design of the command structures and online help. Evidently, in an exploratory and holistic study of this nature, isolating package features to study their effects is not feasible. Given that our results are consistent with that of previous experimental/micro research and that an analysis of the package requirements to fulfil the experimental task were deemed very similar for both packages, this assumption seems reasonable.

The procedures for calculating the performance indices could perhaps be improved. Because it was not possible to verify that subjects had completed the entire task before leaving the lab, we were not able to use the straight scores obtained for the task. The scores had to be weighted by the time it took to achieve these scores. The times collected also included distraction time, which was not necessarily the same for all subjects or in all experimental settings. The scores were also developed conceptually, though verified on the basis of the actual errors made on sub-tasks correctly executed which was used as a measure of difficulty. Since the same scoring was applied to both packages, it is possible that some sub-tasks were either over- or under-rated for one or the other of the packages. The Error and Help Indices were based on their raw number divided by the score on the task. This caused the data to be skewed since although many errors and help calls may have been made during the session, if they were not able to complete any of the tasks, the resulting Indices would be zero. In comparison to an individual making the same number of errors or help calls but accomplishing one task, these Indices would be understated.

Some concern may be expressed regarding the sample sizes in Study 1 and in the individual experience categories in Study 2. Given the number of variables in Study 1, a larger sample size would have been preferable, especially for analyzing the Assistance Features. This is 'easily' correctable in future studies, depending on the cooperation of those solicited. More problematic is the issue of adequate sample size in each of the experience categories we identified in Study 2. It was not possible to assess these levels, a priori, because of the short turn-around time from solicitation of our subjects to their participation in the experiment. Future studies should, however, attend to this deficiency. There were a number of levels for which no conclusions could be made because of insufficient sample sizes. The findings of our study on those levels for which there were sufficient sample sizes, suggest that experience level, based on packages of similar and dissimilar function and/or operation, is an important consideration and one worthy of further investigation.

On the question of randomization, it should be remembered that it was not possible to randomly assign subjects in a true sense to the various package treatments in Study 2. Because of limited resources, human as well as physical, subjects were assigned to the treatments according to their choice of day and time for the experiment. Conditions cannot be easily controlled in the context of the real world. We also chose not to randomize the features included on the Evaluation Form in Study 1. Since the variables are dependent on each other, we believed, whether they were randomized or not, it would not diminish from this fact, and patterns would still be evident.

A major assumption of our research is the viability of applying an exploratory/holistic approach to research previously conducted under experimental/micro conditions. Another major assumption is our view of **ease of use** as being made up of the components, **speed, memory, effort** and **comfort**. The results of our two studies and the consistency of these findings with each other and with prior experimental research suggests that these assumptions have some merit and that our measures are valid.

In light of these limitations, certain improvements to the research can be envisioned. These are discussed in the next section.

6.E. Ways in which research of this type could be improved

A number of improvements, for the purpose of expanding this work in the future, can be suggested, depending on the resources available. The major way in which this study could have been executed more effectively, would have been to have had access to a computer lab reserved strictly for research purposes. In this way, control could have been exerted on the timing, duration and environment of the experiments. This demand may sound excessive for an 'exploratory' field study, where for validity it is advocated that 'natural' surroundings should be maintained, but research effectiveness dictates that certain levels of structure and organization be maintained. It was very restrictive having to plan experimental times around regularly scheduled class lab times, especially given the added requirement of getting candidates before they had gained computer experience, as in the case of novices, and experience in Minitab$_t$ in the case of the others. A research lab would also permit such 'luxuries' as built-in video cameras. Although we tried to video-tape some of the sessions, the results were not useful, except in-so-far as assuring us that the subjects did use the Hypertext-based online Index added to Merlin$_t$, which resided on another computer terminal alongside it. To be able to see actual keystrokes, either on the screen or keyboard, one would require individual cameras trained on each computer. We had between fifteen and thirty subjects in any one session, which topic brings us to the next recommendation.

Research of this magnitude should really be a team effort, not an individual one. With 243 subjects being tested in very short time intervals, it was very difficult to monitor the experimental context. The ideal situation would allow one to check each subject's diskette before their departure, so as to ascertain completion of the task, accurate completion of the questionnaires, and also to identify subjects at extremes of the performance ranges for further probing. Much more could be extracted from the protocol traces. They are a rich bed of information. Particularly attractive are the time-stamps between keystrokes which the Merlin, package provides. The potential for discovering fascinating mental processes and paths is great with such a facility. One other recommendation could be that all packages should be designed with this facility. This could assist in identifying problem areas being experienced by users, both as a mechanism for providing feedback and assistance to users in actual use of the packages and as a focus for improving design.

The instrument used to determine the experience level of classes could be refined further. It may be possible to develop questions pertaining to the actual type of work done with the packages, but this would become more time-consuming for the individual. Also, less experienced users sometimes do not know exactly what they have been doing with the packages. In Study 2, where we asked users Yes/No questions on whether they perceived certain features to be in the package they used, the results indicated that they cannot always differentiate the features accurately. In some instances, they perceived features to exist which are not in the packages, and not recognize when some were present. Schroeder and Kletke

(1991) have taken steps forward in developing an instrument to tap into the nature of the work done with packages, asking questions related to Functional Domain expertise, that is, computer usage; Topical Domain expertise, that is, package subject matter expertise; and Entry expertise, that is, typing skills. The instrument is designed to classify users according to Low, Middle, and High expertise. Testing of the instrument suggests that it is a reliable measure of user expertise. It is certainly an important and interesting area in which to delve further.

The research and its contributions, its limitations, and our suggestions of ways to improve such studies point out various directions that future research could take. We explore these in the following section.

6.F. Directions for Future Research

There are many areas in the Human Factors discipline in which research is required. The Framework which we advocate to position research attests to this. The variables are so complex and interrelated, that the depths may never be plumbed, even remotely. Nevertheless, we can, in our limited circumstances, envisage a number of avenues which could feasibly be taken. We enumerate these below:

1. Any study needs to be replicated. Also, more in-depth analyses of the process traces collected could go a far way to unearthing the explanations for the results found. The strategies and ways in which users learn a package, their reactions to the feedback provided from errors made and help sought, as well as the time spent in these states,

have much to tell us about the **ease of use** of a package and in what aspects. It is a useful way of identifying areas in the package design in need of correction and modification, those areas which improve performance or degrade it.

2. It would be beneficial to redo Study 1 with respect to the effect of different experience levels, which, in Study 2, were found to have an effect on **ease of use** as we measured it.

3. Our view of **ease of use** as being composed of **Speed**, **Memory**, **Effort** and **Comfort** and the mixed support for these suggested by our preliminary studies, suggests a more rigorous study of the issue. The dimensions need further refinement and operationalizing, in themselves, and in their relationship to each other. The relationship between package features and support of these dimensions also requires further study. An evaluation which can be 'qualified' on the basis of these dimensions has the potential for improving assessment of **ease of use** over that of a uni-dimensional approach.

4. The Evaluation Form we are suggesting requires validation. We need to discover whether **ease of use** is a generalizable concept or whether it is an individual concept, and whether this is stable over time. Further, is it contingent on experience level, or on other individual differences? To account for the individual experience level differences that we noted, we need to look at the same evaluator on each of the packages being evaluated. A Latin Squares design would be necessary to account for the order effect. It would also be interesting to investigate other techniques for analyzing the results. One notable approach would be to use Repertory Grid Analysis (Gaines and Shaw, 1984), which is used to interactively elicit mappings of multiple factors and effects from respondents. It permits evaluators to change their prioritizing of these elements as they consider each consecutive feature, until they have achieved what they perceive to be the optimal relationships.

5. The dilemma of the possible inconsistency between actual objective and subjective measures relating to **ease of use**, and inconsistency between these and expectations based on evaluations is an important area for further investigation. On what basis can **ease of use** be judged, given these inconsistencies?

6. The whole realm of experience level classification is in need of refinement. Lack of adequate sample sizes made it impossible for us to accurately portray some of our projected classifications, and in some cases, there were no observations on which to report. The findings in Study 2 suggest that there is some validity to our perspective on this issue. More work in this area is indicated.

7. In the section in Chapter 2, on the deterrents and remedies to Human Information Processing we discussed the application of learning theory to solutions for software design. A number of studies could be designed to test the efficacy of the various solutions suggested for **ease of use** of a package. Some of these solutions also suggest challenges from a technological point of view. One example might be to test the usefulness of looking-back traces for assisting learning. This recommendation raises

technical considerations, such as the allocation of memory resources and the duration and extent to which these traces should be stored.

8. We noted in Chapter 2, in our assessment of current research within the context of our proposed framework, that a number of areas are still very under-researched. One example is an assessment of **ease of use** within the context of 'canned' packages versus packages for decision analysis purposes. Numerous others are suggested by the framework.

9. The task we chose for our study was not package-specific. A study requiring use of more of the package-specific features of the software, such as probabilities or regression, is an important next step in the research. The impact of experience is likely to be very important here.

10. An holistic approach to evaluation of **ease of use** suggests a number of studies. One possible enhancement to our current research would be a longitudinal study to examine the effects of the various factors on **ease of learning**, that is, movement along the learning curve and the incremental effect as new features of the package are learnt. Another interesting study might be to investigate the feasibility of developing an 'optimal' expert performance index against which to assess all other levels, and determining the length of time it takes to achieve this level.

11. Our less than overwhelming results may indicate that objective measurement of **ease of use** should be considered separately from subjective measurement of it. They may also suggest that **ease of use** is a separate construct which should be considered as an entity onto itself and not as composed of the dimensions we identified. Some investigation of these possibilities is suggested.

We trust that the discussion above will spark other researchers to share our interest in these areas, and pursue further studies in them. Concluding comments are given in Chapter 7, following.

Chapter 7 - CONCLUSIONS

The growth in personal computer use in the business world is increasingly shifting from what Glennan (1967) refers to as a 'technology-push' to a 'requirements-pull'. Technology was the driving force in computer design and use in its early stages of development and introduction into the business world. However, as these powerful machines are increasingly being used by non-technical, non-computer experts, the need to accommodate users' desires is heightened, in order to exploit the market forces which are developing and maintain the competitive edge that such accommodation promises to software manufacturers sensitive to these needs. Understanding the factors which contribute to meaningful and successful learning and use of software, and knowing how to provide guidance to attain this goal are, therefore, becoming important priorities. In this thesis, we suggested that accomplishment of this objective can be aided by reference to learning theory concepts and the design principles of Computer Assisted Learning, and their incorporation into business software design. In particular, we proposed, and found some modicum of support for, evaluating the **ease of use** of a package on the extent to which its package design and assistance features contribute to support of important learning factors, identified as being **Speed, Memory, Effort,** and **Comfort**.

The studies conducted enabled us to elaborate on the concept of **ease of use**, within this context, and to gain some useful insights into it. The question still remains, however, on what basis should the evaluation of **ease of use** be made? Our findings indicate that subjective measures may not lean in the same direction as the results found on one objective measure. This poses quite a dilemma for those designing packages and those choosing among packages.

We know that user satisfaction with a package is an important component of usage. If the package does not meet some level of perceived **ease of use**, it will either not be used or used sub-optimally. In a review of the WordPerfect, package, newspaper columnist MacGregor, 1993, estimates that, "About 95% of its functions will never be used by 98% of its users". A package's weaknesses will only be overlooked if a package has a function which is considered vital to the user. In that case, some degree of 'discomfort' with the package will be tolerated, since functionality will take precedence over this component of **ease of use**. A trade-off between functionality and **ease of use** often has to be made when designing software. More specialized and complex functions, which often slow down processing, often cannot be accommodated within the confines of user-friendly designs. In some situations, 'good' design principles have to be sacrificed in order to accommodate needs. For example, in one study on which we reported, the screen design violated the rules on the number of items that should be displayed on one screen; however, all the elements were essential to the users of this software. The solution was to group concepts into meaningful patterns, use comprehensible abbreviations and provide training to users. There is no easy solution to the design of **easy to use** software. As we noted previously, on the one hand, users want and need consistency, and on the other, need and want adaptability. Software design and evaluation seem to be always plagued with contradictions. The issue is by no means resolved and becomes even more complex when decisions about software are being made by a third party, such as is done by the Information Centres of some organizations.

Ease of use evidently depends on a number of factors as our research shows. Adams, et.al. (1992, p.245), state in their article,

> "There is no absolute measure of ease of use or usefulness, and user perceptions of these constructs may vary with time and experience for any given application...It may be that a variety of factors, such as user experience, type or sophistication of system use, or other task and user characteristics may mediate the relationship between ease of use and usage."

Also, to reiterate Carroll and Mack, cited in Davis (1989, p. 323) and quoted in our preface,

> "Although objective ease of use is clearly relevant to user performance given the system is used, subjective ease of use is more relevant to the users' decision whether or not to use the system and may not agree with the objective measures."

While we did not assess subjective **ease of use**, we did assess subjective perceptions of the packages, in terms of perceived comfort, and this was found to be in harmony with objective performance overall, but not with respect to the individual packages tested. Overall, users with better performance gave better perceived comfort ratings, but the package which produced superior performance received somewhat lower perceived comfort ratings than did the other package on first use. On reuse, however, the perceived comfort ratings were switched and, though not significant, were now in line with the objective measures.

In the preface, we challenged the assumption of the feasibility of objective design standards being used as a basis for evaluating packages. There, we suggested that:

1) the merits of the inclusion or exclusion of a particular design or assistance feature may be agreed on by experts, but, in fact, the degree to which features are deemed 'good' in a particular package may differ for different classes of users, thereby producing different performance and perception results;

2) assessment of design and assistance features included in a package should be based on learning theory, that is, on the extent of their contribution to the major learning factors of speed, memory, effort, comfort. On this basis, packages which include more features supporting these learning factors may be said to be 'better' than those offering less support;

3) expert designers/users and novice users of a package will favour packages which support different dimensions of learning, that is, their criteria for ease of use and learning will fall under different dimensions. In particular, novices will favour memory and comfort support because of unfamiliarity with, and anxiety resulting from, use of the system, while expert designers and users will favour speed and effort support. Expert designers and users already familiar with a package, or a like package, are interested in getting the job done as quickly and effortlessly as possible.

Our study seemed to show that even as experts do agree to some degree on the importance of certain design and assistance features for package design, user performance and perceptions also seem to differ depending on these features, and on prior experience with packages, among other factors. Those users perceiving more attributes in the package supporting speed, memory, effort and comfort, to some extent had better performance and perceptions of the packages than those perceiving fewer attributes supporting these learning dimensions. Unfortunately, the sample sizes in the individual experience levels were not large enough to test our third assumption. That must be reserved for future research.

The variables we considered did not have strong predictive power. indicating that individual differences account for a large portion of the variance in our measures. Allowances for differences among users may need to be accommodated and incorporated in design more than has been realised to date. This finding, if valid, may call into question a number of design decisions which have been made based on the results obtained from previous experimental/micro research. If, in fact, a particular command structure, or other feature, has been adopted based on results, which according to our findings, have low predictive value with respect to **ease of use**, this would, indeed, imply difficulties for researchers and designers, and would possibly explain why users often have difficulties with the resulting designs.

Our results tend to indicate that an Evaluation Form for **ease of use**, based strictly on design standards, is inadequate. We have suggested that one step towards a greater understanding is the investigation of the viability of an Evaluation Form which considers the extent of the support for the learning dimensions, **speed, memory, effort** and **comfort** offered by the features in the packages. The resulting evaluation may then need to be tempered with type and extent of prior package experience. The relative merits of this approach would have to be tested.

While there was not strong support for our view of **ease of use** as being comprised of **speed, memory, effort** and **comfort**, it is still an heuristically appealing concept and our study has not provided strong evidence to suggest that it should be abandoned without further investigation. It may be that objective measurement of **ease of use** should be considered separately from its subjective measurement. In this way, it may be possible to more accurately articulate **ease of use** and the relationship **speed, memory, effort** and **comfort** may have with each other and with this construct. It may also be that **ease of use** is a construct unto itself, which must be considered as such, as has been done in previous research, and not in terms of the dimensions we identified. Our understanding of the construct of **ease of use** is evidently still incomplete.

We end with a touch of humour, this depiction being all too often a reality, and indicative of the kind of challenge that software designers must face in attempting to provide **easy to use** package designs to meet users' requirements and expectations.

'he Gazette, January 14, 1991

BIBLIOGRAPHY

Adams, D.A., Nelson, R.R. and Todd, P.A. "Perceived Usefulness, Ease of Use, and Usage of Information Technology: A Replication". MIS Quarterly, June 1992, p. 227-247.

Adler, P.S. and Clark, K.B. "Behind the Learning Curve: A Sketch of the Learning Process". Management Science, 1991, 37(3), p. 267-281.

Aldrich, J.H. and Nelson, F.D. "Linear Probability, Logit, and Probit Models". In Series: Quantitative Applications in the Social Sciences, London: Sage University Publications, 1980. p. 9-83.

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

Ambardar, A.K. "Human-Computer Interaction and Individual Differences". In G.Salvendy (Ed.), Human Computer Interaction. New York, N.Y.: Elsevier Publishers, 1984, p. 207-211.

Anderson, J.R. Cognitive Psychology and Its Implications, 2nd. editions. N.Y.: W.H. Freeman and Company, 1985.

Andreali, P. and Sleadman, J. Management Decision Support Systems: Impact on the Decision Process. Masters Thesis, MIT, 1975.

Andriole, S.J Interactive Computer-Based Systems. New York, N.Y.: Petrocelli Books, 1983.

Argote, L. Beckman, S.L., and Epple, D. "The Persistence and Transfer of Learning in Industrial Settings". Management Science, February 1990, 36(2), p. 140-154.

Arnett, K. and Trumbly, J. "The Performance Relationship Model: A Framework for Human Factors Research". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 19-1-19-9.

Badre, A.N. "Designing Transitionality into the User-Computer Interface". Human-Computer Interaction, 1984, p. 27-34.

__________. "Designing Chunks for Sequentially Displayed Information". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. New Jersey, N.Y.: Ablex Publishing, 1982, p. 179-193.

Baecker, R.M. and Buxton, W.A.S. "An Historical and Intellectual Perspective". In J. Preece and L. Keller (Eds.), Human-Computer Interaction, 1990, p. 3-26.

__________. Readings in Human-Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987.

Bailey, C.D. "Forgetting and the Learning Curve: A Laboratory Study". Management Science, March 1989, 35(3), p. 340-352.

Bailey, J.E. and Pearson, S.W. "Development of a Tool for Measuring and Analyzing Computer User Satisfaction". Management Science, 29(5), 1983, p. 530-545.

Bannon, L. and O'Malley, C. "Problems in Evaluation of Human-Computer Interfaces: A Case Study". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985, p. 709-713.

Barnard, P.J. and Grudin, J. "Command Names". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 237-255.

Barnard, P.J., Hammond, N., Maclean, A. and Morton, J. "Learning and remembering interactive commands in a text-editing task". Behaviour and Information Technology, 1, 1982a, p. 347-358.

Barnard, P.J., Hammond, N., Maclean, A. and Morton, J. "Learning and remembering interactive commands". Proceedings of Conference on Human Factors in Computing Systems, 1982b, p. 2-7.

Barnard, P.J., Hammond, N., Morton, J. and Long, J.B. "Consistency and Compatibility in Human-Computer Dialogue". Int. J. of Man-Machine Studies, 1981, 15, p. 87-134.

Bayman, P. and Mayer, R.E. "Instructional manipulation of users' mental models for electronic calculators". Int. J. of Man-Machine Studies, 1984, 20, p. 189-199.

Bennett, J.L. "Analysis and Design of the User Interface for Decision Support Systems". In J.L. Bennett (Ed.), Building Decision Support Systems, Reading, Mass.: Addison Wesley, 1983, p. 41-64.

Bethke, F.J., et.al. "Improving the usability of programming publications". IBM Syst.J.,1981, 20(3), p. 306-320.

Bevan, N. "Is there an optimum speed for presenting text on a VDU?". Int. J. of Man-Machine Studies, 14, 1981, p. 59-76.

Bieber, M.P. and Kimbrough, S.O. "On Generalizing the Concept of Hypertext". MIS Quarterly, March 1992, p. 77-92.

Bigge, M.L. Learning Theories for Teachers. 4th edition. N.Y.: Harper and Row Publishing, 1982.

Bitter, G.G. Using a Microcomputer in the classroom. Reston, Virginia: Prentice Hall, 1984.

Black, J.B., Bechtold, J.S., Mitrani, M. and Carroll, J.M. "On-line Tutorials: What Kind of Inference Leads to the Most Effective Learning?". CHI '89 Proceedings, May 1989, p. 81-83.

Black, J.B., Kay, D.S. and Soloway, E.M. "Goal and Plan Knowledge Representations from Stories to Text-Editors and Programs". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 36-60.

Black, J. and Moran, T. "Learning and remembering command names". Proceedings of Conference on Human Factors in Computing Systems, 1982, p. 8-11.

Borenstein, N.S. The design and evaluation of online help systems. Unpublished Doctoral Dissertation, Dept. of Computer Science, Carnegie Mellon, Pittsburg, PA., 1985.

Borgman, C.L. "The user's mental model of an information retrieval system: an experiment on a prototype online catalog". Int. J. Man-Machine Studies, 1986, 24, p. 47-64.

Borgman, C.L. "Why are online catalogs had to use?" J. of American Society for Information Science, 37(6), 1984, p. 387-400.

Bostrom, R.P., Olfman, L. and Sein, K.M. "The Importance of Learning Styles in End-User Training". MIS/Quarterly, March 1990, p. 101-119.

______________________. "The Effects of Training on Motivation to Use and Self-reported Use of an Electronic Mail System". Proceedings of the second Symposium on Human Factors in MIS, Sacramento,California, 1989, p. 15-1-15-9.

Bovair, S., Kieras, D.E. and Polson, P.G. "The Acquisition and Performance of Text-Editing Skill: A Cognitive Complexity Analysis". Human-Computer Interaction, 1990, 5, p. 1-48.

Boyd, G.M. and Jaworski, W.M. "Pals,Paths,Places,and Producers: Four,more appropriate forms of Computer-Assisted Education". Proceedings of COMPINT85 - IEEE, New York, 1985, p. 614-616.

Brahan, J.W. "Definition, Implementation and Assessment of NATAL-74 - An Author Language for Computer-Aided Learning". INFOR, May 1980, 18, p. 199-207.

__________. "Portability, productivity - the NATAL approach to CAL". CIPS Review, July-August 1982, p. 10-12.

Brancheau, J.C. and Wetherbe, J.C. "Key Issues in Information Systems Management", MIS/Quarterly, March 1987, 11(4), p. 23-45.

Briggs Myers, I. and McCaulley, M.H. Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator. Palo Alto,Calif: Consulting Psychologists Press, 1985.

Briggs, P. "Do they know what they're doing? An Evaluation of Wordprocessor users' implicit and explicit task-relevant knowledge, and its role in self-directed learning". Int. J. of Man-Machine Studies, 1990, 32, p. 385-398.

Brown, J.S. and VanLehn, K. "Repair Theory:A Generative Theory of Bugs in Procedural Skills". Cognitive Science, 1980, 4, p. 379-426.

Brown, J.S. and Burton, R.R. "A paradigmatic example of an artificially intelligent instructional system". Int. J. of Man-Machine Studies, 1978, 10, p. 323-339.

Brown, R. Interfacing with People. Systems Education Couise, Bell Corporation System Organization, 1987.

Burns, M.J., Warren, D.L. and Rudisill, M. "Formatting Space-Related Displays to Optimize Expert and Nonexpert User Performance". CHI Proceedings, April 1986, p. 274-280.

Card, S.K., Moran, T.P. and Newell, A. The Psychology of Human-Computer Interaction. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983.

Card, S.K., Moran, T.P. and Newell, A. "The Keystroke-Level Model for User Performance Time with Interactive Systems". Communciations of the ACM, 1980a, 23(7), p. 396-410.

____________________. "Computer Text-Editing: An Information-Processing Analysis of a Routine Cognitive Skill". Cognitive Psychology, 1980b, 12, p. 32-74.

Carey, J.M. "The Issue of Cognitive Style in MIS Research". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 18-1-18-9.

Carey, T.T. "A Usability Requirements Model for Procurement Life Cycles". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 7-1-7-13.

__________. "User Differences in Interface Design". Computer, November 1982, 18, p. 14-20.

Carroll, J.M. "Minimalist Design for Active Users". Human-Computer Interaction, 1985, p. 39-44.

__________. "The Adventure of Getting to Know a Computer". IEEE, 1982, p. 49-58.

Carroll, J.M. and Aaronson, A.P. "Learning By Doing With Simulated Intelligent Help". Communications of the ACM, Sept.1988, 31(9), p. 1064-1079.

Carroll, J.M. and Carrithers, C. "Training Wheels in a User Interface". Communications of the ACM, 1984, 27(8), p. 800-806.

Carroll, J.M. and Kay, D.S. "Prompting, feedback and error correction in the design of a scenario machine". Int. J. of Man-Machine Studies, 1988, 28, p. 11-27.

Carroll, J.M. and Kay, D.S. "Prompting, feedback and error correction in the design of a scenario machine". Proceedings of CHI'85: Human Factors in Computer Systems, 1985, p. 149-153.

Carroll, J.M. and Mack, R.L. "Learning to Use a Word Processor: By Doing,by Thinking, and by Knowing". In J.C. Thomas and M. Schneider (Eds.), Human Factors in Computing Systems, Norwood, NJ: Ablex Publishing Corp., 1984, p. 13-51.

Carroll, J.M., Mack, R.L. and Kellogg, W.A. "Interface Metaphors and User Interface Design". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 67-85.

Carroll, J.M., and McKendree, J. "Interface Design Issues for Advice-Giving Expert Systems". Communications of the ACM, 1987, 30(1), p. 14-31.

Carroll, J.M. and Olson, J.R. "Mental Models in Human-Computer Interaction". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 45-65.

Carroll, J.M. and Rosson, M.B. "Paradox of the Active User". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 80-111.

Carroll, J.M. and Thomas, J.C. "Metaphor and the Cognitive Representation of Computing Systems". IEEE Transactions on Systems, Man and Cybernetics, 1982, 12(2), p. 107-115.

Catrambone, R. "Specific versus General Procedures in Instruction". Human-Computer Interaction, 1990, 5, p. 49-93.

Catrambone, R. and Carroll, J.M. "Learning a Wordprocessing System with Training Wheels". Proceedings of CHI and GI, 1987, p. 169-174.

Charkins, R.J., et.al. "Linking Teacher and Student Learning Styles with Student Achievement and Attitudes". J.of Economic Education, Spring 1985, p. 111-120.

Charney, D.H. and Reder, L.M. "Designing interactive tutorials for computer users". Human-Computer Interaction, 2, 1986, p. 297-317.

Chignell, M.H. and Hancock, P.A. "Intelligent Interface Design". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 969-995.

Clement, D. "The Role of Documentation in Human-Computer Interaction". Human-Computer Interaction, 1984, p. 203-205.

Collins, A.M. and Stevens, A. "Goals and Strategies of Inquiry Teachers". In R.Glaser, (Ed.), Advances in Instructional Psychology,Vol.2. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1982, p. 65-119.

Cohill, A.M. and Williges, R.C. "Retrieval of HELP Information for Novice Users of Interactive Computer Systems". Human Factors, 1985, 27(3), p. 335-343.

Cohill, A.M. and Williges, R.C. "Computer-augmented retrieval of HELP information for novice users". Proceedings of Human Factors Society - 26th Annual Meeting, 1982, p. 79-82.

Cohill, L.F. "A Taxonomy of User-Computer Interface Functions". In G.Salvendy (Ed.), Human-Computer Interaction. New York, N.Y.: Elsevier Publishers, 1984, p. 125-128.

Cordes, R.E. "Application of Magnitude Estimation for Evaluating Software Ease-of-Use". Human-Computer Interaction, 1984, p. 199-202.

Cronbach, L.J. Essentials of Psychological Testing. N.Y., N.Y.: Harper & Row Publishers, 1990.

CSR Trainer 3000: IBM Personal Computer Instructional System. Avon, Conn.: Computer Systems Research, Inc., 1983.

Curtis, B., et.al., "Software Psychology: The Need for an Interdisciplinary Program". In R.M. Baecker and W.A.S. Buxton (Eds.), Readings in Human Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987, p. 150-164.

Czaja, S.J. "Microcomputers and the Elderly". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 581-598.

Czaja, S.J., Hammond, K, Blascovich, J.J. and Swede, H. "Learning to use a wordprocessing system as a function of training strategy". Behaviour and Information Technology, 5, 1986, p. 203-216.

Dannenbring, G.L. "Effect of computer response time on user performance and satisfaction". Behavioural Research Methods and Instrumentation, 15, 1983, p. 213-216.

Davis, D., Meinert, D.B. and Davis, D.L. "The Link between Satisfaction and Performance in a Decision Support Environment: An Empirical Investigation". Proceedings of the 20th Annual Meeting of the Western Decision Sciences Institute, Lihue, Kauai, Hawaii, March 19-22, 1991, p. 370-372.

Davis, F.D. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". MIS Quarterly, 1989, p. 319-340.

Davis, F.D., Bagozzi, R.P. and Warshaw, P.R. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models". Management Science, 1989, 35(8), p. 982-1003.

Davis, R. "User error or computer error? Observations on a statistical package". Int. J. of Man-Machine Studies, 1983, 19, p. 359-376.

________. "Task analysis and user errors: a methodology for assessing interactions". Int. J. of Man-Machine Studies, 1983, 19, p. 561-574.

DBASE, III+, Trademark, Ashton-Tate.

De Waele, M. "Managerial Style and the Design of Decision Aids". The Int.J.of Management Science, 1978, 6, p. 5-13.

Doane, S.M. A longitudinal study of UNIX users' expertise, UNIX mental models, and task performance. Unpublished Doctoral Dissertation, University of California, Santa Barbara, 1986.

Doll, W.J. and Torkzadeh, G. "The Measurement of End-User Computing Satisfaction: Theoretical and Methodological Issues". MIS Quarterly, March 1991, p. 5-10.

Doll, W.J. and Torkzadeh, G. "The Measurement of End-User Computing Satisfaction". MIS/Quarterly, June 1988, p. 259-274.

Dos Santos, B.L. and Bariff, M.L. "A Study of User Interface Aids for Model-oriented Decision Support Systems". Management Science, 1988, 34(4), p. 461-468.

Douglas, L. An examination of the effect of memory support and advisory support in a learner control computer assisted instruction program. Masters Thesis, Concordia University, Montreal,Quebec, 1982.

Douglas, S.A. Learning to Text Edit: Semantics in Procedural Skill Acquisition. Unpublished Doctoral Dissertation, Stanford University, 1983.

Duffy, T.M. Mehlenbacher, B., and Palmer, J. "The Evaluation of Online Help Systems: A Conceptual Model". In E. Barrett (Ed.), The Society of Text: Hypertext, Hypermedia and the Social Construction of Information, Cambridge, Mass.: MIT Press, 1989.

Dumais, S.T. and Landauer, T.K. "Using examples to describe categories". Proceedings of CHI'83, New York: ACM, 1983, p. 112-115.

Dumais, S.T. and Landauer, T.K. "Psychological Investigations of Natural Terminology for Command and Query Languages". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. N.J.: Ablex Publishing, 1982, p. 95-109.

Dunmore, P.V. "McMaster's System for Computer-Aided Learning in Accounting". CAAA Conference, 1982, p. 49-57.

Dunn, R., et.al. "Identifying Individual Learning Styles". In Student Learning Styles. Reston, Virginia: National Association of Secondary School Principals, 1979, p. 39-54.

Dunsmore, H.E. "Line and Screen Text Editors: Some University Experiment Results". Human-Computer Interaction, 1984, p. 157-164.

Dunsmore, H.E. A report on research. Unpublished manuscript, Purdue University, 1981.

Dunsmore, H.E. "Designing an interactive facilty for non-programmers". Proceedings of ACM National Conference, 1980, p. 475-483.

Earl, T. The Art and Craft of Course Design. London: Kogan Page, 1987.

Ebbinghaus, H. Memory: A Contribution to Experimental Psychology. (Translation by Henry A. Ruger and Clara E. Bussenius and with an introduction by Ernest Hilgard, 1964), Dover, N.Y., 1885.

Eberts, R.E. and Brock, J.F. "Computer-Based Instruction". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 599-627.

Efe, K. "A Proposed Solution to the Problem of Levels in Error-Message Generation". Communications of the ACM, November 1987, 30(11), p. 948-955.

Egan, D.E. "Individual Differences in Human-Computer Interaction". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 543-568.

Egan, D.E. and Gomez, L.M. "Assaying, isolating and accommodating individual differences in learning complex skills". In R. F. Dillon (Ed.), Individual Differences in Cognition - Vol. 2, Orlando, Florida: Academic Press, 1985, p. 173-217.

Egan, D.E., Nakatani, L.H., Shaw, A.C. and Hawley, P. "Empirical studies using TNT: error diagnosis and coaching strategies". Proceedings of EURIT '86: European Conference on Information Technology in Education, Enschede, the Netherlands, 1987.

Ehrenreich, S.L. and Porcu, T. "Abbreviations for Automated Systems: Teaching Operators the Rules". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. N.J.: Ablex Publishing, p. 1982, 111-135.

Einhorn H.J. and Koelb, C.T. "A Psychometric Study of Literary-Critical Judgment". Modern Language Studies, 12(3), 1982, p. 59-82.

Elkerton, J. "Online Aiding for Human-Computer Interface". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 345-364.

Elkerton, J. and Williges, R.C. "A summary of experimental research on command-selection aids". Proceedings of INTERACT '87, Second IFIP Conference on Human-Computer Interaction, New York: North-Holland, 1987, p. 937-942.

Elkerton, J. and Williges, R.C. "Information retrieval strategies in a file-search environment". Human Factors, 26, 1984a, p. 171-184.

Elkerton, J. and Williges, R.C. " The effectiveness of a performance-based assistant in an information retrieval environment". Proceedings of the Human Factors Society 28th Annual Meeting, Santa Monica, CA: Human Factors Society, 1984b, p. 634-638.

Emmons, W.H. "A comparison of cursor-key arrangements (box versus cross) for VDUs". In Grandjean Etienne (Ed.), Ergonomics and Health in Modern Offices, Philadelphia: Taylor and Francis, 1984, p. 214-219.

Entwistle, N.J. "Symposium:Learning Processes and Strategies IV-Knowledge Structures and Styles of Learning: A Summary of Pask's Recent Research". British J. of Educational Psychology, 1978, 48, p. 255-265.

Etezadi-Amoli, J. and Farhoomand, A.F. "On End-User Computing Satisfaction". MIS Quarterly, March 1991, p. 1-4.

Evans, G.E. and Simkin, M.G. "What Best Predicts Computer Proficiency?". Communications of the ACM, November 1989, 32(11), p. 1323-1327.

Favaro, P.J. Educator's Guide to Micro-Computers and Learning. New Jersey: Prentice Hall, 1986.

Fertuck, L. "The Use of Computers in Teaching Accounting". Canadian Academic Accounting Assoc. Annual Conference, 1982, p. 41-48.

Ford, N. "Learning Styles and Strategies of Postgraduate Students". Br.J.of Educational Psychology, January 1985, 16, p. 65-79.

Foss, D.J. and DeRidder, M. "Technology Transfer: On Learning a New Computer-Based System". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 159-183.

Fox, R.D. "Learning Styles and Instructional Preferences in Continuing Education for Health Professionals: A Validity Study of the LSI". Adult Education Quarterly, Winter 1984, 35, p. 72-85.

Framework(r), Trademark, Ashton-Tate.

Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T. "Statistical Semantics: Analysis of the Potential Performance of Keyword Information Systems". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. N.J.: Ablex Publishing, 1982, p. 187-241.

Gable, A. and Page, C.V. "The use of Artificial Intelligence techniques in Computer-Assisted Instruction: an overview". Int. J. of Man-Machine Studies, 1980, 12, p. 259-282.

Gagne, R.M. The Conditions of Learning. N.Y.: Holt Reinhart and Winston, Inc., 1977.

_________. Essentials of Learning for Instruction. Hindsdale,Ill.: Dryden Press, 1974.

Gagne, R.M. and Briggs, L.J. Principles of Instructional Design. N.Y.: Holt Reinhart and Winston, Inc., 1979.

Gaines, B.R. "From Ergonomics to the Fifth Generation:30 years of Human-Computer Interaction Studies". In B.Shackel (Ed.), Human-Computer Interaction. New York, N.Y.: Elsevier Publishers, 1985, p. 3-7.

Gaines, B.R. and Shaw, M.L.G. The Art of Computer Conversation. Toronto, Canada: Prentice Hall, 1984.

Galletta, D.F. and Lederer, A.L. "Some Cautions on the Measurement of User Information Satisfaction". Decision Sciences, 1989, 20, p. 419-438.

Ganning, R.G. (Ed.). "Training for End Users". EDP Analyzer, August 1982, 20, p. 1-14.

Gazzaniga, M.S. and Ledoux, J. The integrated mind. N.Y.: Plenum Press, 1978.

Gerlach, J.H. and Kuo, F. "Understanding Human-Computer Interaction for Information Systems Design". MIS Quarterly, December 1991, p. 526-549.

Gewirtz, M. "Training after Training". Information Centre, October 1988, IV(10), p. 41.

Gilfoil, D.M. "Warming up to computers: A study of cognitive and affective interaction over time". Proceedings of the Conference on Human Factors in Computer Systems, (Gaithersburg, Md., March 15-17), Washington, D.C.: ACM, 1982, p. 245-250.

Gilroy, F.D. and Desai, H.B. "Computer anxiety: sex, race and age". International J. of Man-Machine Studies, 1986, 25, p. 711-719.

Glaser,R. Adaptive Education:Individual Diversity and Learning. Montreal, Quebec: Holt, Rinehart and Winston, 1977.

Glennan, T.K. Jr. "Issues in the Choice of Development Policies". In T. Manschak, T.K. Glennan, Jr., and R. Summers (Eds.), Strategies for research and development. N.Y.: Springer-Velaz, 1967.

Godfrey, D. and Sterling, S. The Elements of CAL. Toronto, Ontario: Press Porcepic Ltd., 1982.

Goleman, D. "Holographic Memory", Psychology Today, February 1979, p. 71-84.

Gomes, L. "Less may be more in software, consumer survey suggests". The Gazette, December 30, 1992.

Gomez, L.M., Egan, D.E. and Bowers, C. "Learning to use Text-Editors", Human-Computer Interaction, 2, 1986, p. 1-23.

Good, M.D., Whiteside, J.A., Wixon, D.R. and Jones, S.J. "Building a User-Derived Interface". Communications of the ACM, 1984, 27(10), p. 1032-1043.

Goodwin, L. and Sanati, M. "Learning computer programming through dynamic representation of computer functioning: Evaluation of a new learning package for Pascal". Int. J. of Man-Machine Studies. 25, p. 327-341.

Gould, J. "How to Design Usable Systems". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 757-789.

Green, A.J.K. and Gilhooly, K.J. "Individual differences and effective learning procedures: the case of statistical computing". Int. J. of Man-Machine Studies, 1990, 33, p. 97-119.

Greenacre, M.J. Theory and Applications of Correspondence Analysis. Montreal, Canada: Academic Press, 1984.

Greenacre, M.J. and Hastie, T. "The Geometric Interpretation of Correspondence Analysis". J. of the American Statistical Association, 1987, 82(398), p. 437-447.

Grudin, J. "The Case Against User Interface Consistency". Communications of the ACM, October 1989, 32(10), p. 1164-1173.

Grudin, J. and Barnard, P. "The Role of Prior Task Experience in Command Name Abbreviation". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985a, p. 295-299.

Grudin, J. and Barnard, P. "When does an abbreviation become a word and related questions". Proceedings of CHI'85 Conference on Human Factors in Computing Systems, 1985b, p. 121-126.

Grudin, J. and Maclean, A. "Adapting a Psychophysical Method to Measure Performance and Preference Tradeoffs in Human-Computer Interactions". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985, p. 737-741.

Gugerty, L. and Olson, G.M. "Debugging by Skilled and Novice Programmers". CHI Proceedings, April 1986, p. 171-174.

Guillemette, R.A. "The Usability Criterion for Designing Information Systems: a conspectus". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 6-1-6-23.

Gwei, G.M. and Foxley, E. "Towards a Consultative Online Help System". Int. J. of Man-Machine Studies, 32, 1990, p.363-383.

Haertel, G.D., Walberg, H.J. and Weinstein, T. "Psychological Models of Educational Performance: A Theoretical Synthesis of Constructs". Review of Educational Research, Spring 1983, 53(1), p. 75-91.

Hammond, N. "Recall as an Indicant of Performance in Interactive Systems". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985, p. 311-315.

Harmon, P. "Intelligent Job Aids:How Artificial Intelligence will Change Training in the Next Five Years". In G.Kearsley (Ed.), Artificial Intelligence and Instruction. Don Mills, Ontario: Addison-Wesley Publishing, 1987, p. 165-190.

Hauptmann, A.G. and Green, B.F. "A comparison of command, menu-selection, and natural language computer programs". Behaviour and Information Technology, 2(1), 1983, p. 163-178.

Heeter, C., Atwater, T., Stanley, C. and Baldwin, T. "Videotex: Creating and Examining Experienced Users". Telematics and Informatics, 1987, 4(1), p. 63-74.

Herndon, J.N. "Learner Interests, Achievement, and Continuing Motivation in Instruction". J.of Instructional Development, 1987, 10(3), p. 11-14.

Hiltz, S.R. and Kerr, E.B. "Learning modes and subsequent use of computer-mediated communication systems". CHI Proceedings, 1986, p. 149-155.

Holcomb, R. and Tharp, A.L. "What Users Say About Software Usability". Int. J. of Human-Computer Interaction, 1991, 3(1), p. 49-78.

Holt, R.W., Boehm-Davis, D.A. and Schultz, A.C. "The Effects of Structured, Multi-Level Documentation". CHI Proceedings, April 1986, p. 122-128.

Houghton, Jr., R.C., "Online Help Systems: A Conspectus", Communications of the ACM, 27, February 1984, p. 126-133.

Howard, G.S. Computer Anxiety and the Use of Microcomputers in Management. Ann Arbour, Michigan: UMI Research Press, 1986.

Howard, G.S. and Kernan, M.M. "Computer Anxiety: Concepts and Instruments". Proceedings of Annual Meeting of Decision Sciences, 1990, p. 937-939.

Howard, G.S. and Smith, R.D. "Computer Anxiety in Management: Myth or Reality?". Communications of the ACM, 1986, 29(7), p. 611-615.

Hunt,R.G. and Sanders,G.L. "Propaedeutics of Decision-making:Supporting Managerial Learning and Innovation. Decision Support Systems, 1986, 2, p. 125-134.

Igbaria, M. "End-User Computing Effectiveness: A Structural Equation Analysis". Omega, 1990, 18(6), p. 637-652.

Ives, B. "Graphical User Interfaces for Business Information Systems". MIS Quarterly/ Special Issue, 1982, p. 15-47.

Ives, B., Olson, M.H. and Baroudi, J.J. "The measurement of User Information Satisfaction". Communciations of the ACM, 26(10), October 1983, p. 785-793.

Janosky, B., Smith, P.J. and Hildreth, C. "Online Library Catalog Systems: an Analysis of User Errors". Int. J. of Man-Machine Studies, 1986, 25, p. 573-592.

Jarman, R.O. and Arnett, K.P. "Evaluating User Preference and Performance with Different Menus: A Laboratory Experiment with Wordprocessing Tasks". Proceedings 1990 - Annual Meeting, Decision Sciences Institute, San Diego, November 19-21, 1990, p. 978-980.

Jeffries, R., Miller, J.R., Wharton, C. and Uyeda, K.M. "User interface evaluation in the real world: a comparison of four techniques". Proceeding of the ACM - CHI '91, 1991, p. 119-124.

Jonassen, D.H. and Hannum, W.H. "Research-Based Principles for Designing Computer Software". Educational Technology, December 1987, p. 7-14.

Kagan, D.M. and Pietron, L.R. "Aptitude for Computer Literacy". Int. J. of Man-Machine Studies, 1986, 25, p. 685-696.

Kammersgaard, J. "Four Different Perspectives on Human-Computer Interaction". In J. Preece and L. Keller, (Eds.), Human-Computer Interaction, 1990, p. 42-63.

Kantorowitz, E. and Sudarsky, O. "The Adaptable User Interface". Communications of the ACM, 1989, 32(11), p. 1352-1358.

Karat, J. "Software Evaluation Methodologies". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 891-903.

Karat, J., Boyes, L., Weisgerber, S. and Schafer, C. "Transfer between Wordprocessing Systems". CHI Proceedings, April 1986, p. 67-71.

Karat, J., McDonald, J.E. and Anderson, M. "A Comparison of Selection Techniques: Touch panel, mouse, and keyboard". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985, p. 189-193.

Katz, R. "Interfacing Users and Software". BYTE, December 1983, p. 250-251.

Kearsley, G. (Ed.). Artificial Intelligence and Instruction. Don Mills, Ontario: Addison-Wesley, 1987.

__________. "Microcomputer Software: Design and Development Principles". J. of Educational Computing Research, 1985, 1(2), p. 209-220.

__________. "Authoring Systems in Computer Based Education". Communications of the ACM, July 1982, 25, p. 429-437.

Kearsley, G.et.al. "Two Decades of Computer Based Instruction Projects: What Have We Learned?". THE Journal, February 1983, 10, p. 88-96.

Keller, A. When Machines Teach. N.Y.: Harper and Row, Publishers, 1987.

Keller, J.M. "Development and Use of the ARCS Model of Instructional Design". J. of Instructional Development, 1987, 10(3), p. 2-10.

Kerlinger, F.N. Foundations of Behavioural Research. Montreal, Canada: Holt, Rinehart & Winston, Inc., 1973.

Kern, G.M. and Matta, K.F. "The influence of personality on self-paced instruction". J. of Computer-Based Instruction, Summer 1988, 15(3), p. 104-108.

Khalifa, M. A Quantitative Model for Predicting Ease of Learning of Interactive Computer Systems. Unpublished Doctoral Dissertation, Wharton School, University of Pennsylvania, 1990.

Kieras, D.E. "Towards a Practical GOMS Models Methodology for User Interface Design". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 135-157.

Kieras, D.E. and Polson, P.G. "An approach to the formal analysis of user complexity". Int. J. of Man-Machine Studies, 1985, 22, p. 365-394.

Knoke, D. and Burke, P.J. "Log-Linear Models". In Series: Quantitative Applications in the Social Sciences, London: Sage University Publications, 1980.

Koenig, J. and Lee, M. Authoring Computer Aided Learning Materials: Some guidelines, Toronto, Ontario: Interactive Image Technologies Inc., November 1982.

Kogan, N. "A Cognitive-Style Approach to Metaphoric Thinking". In R.E.Snow, et.al., (Eds.), Aptitude,Learning, and Instruction, Vol.1. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1980, p. 247-281.

Kolers, P.A. "Reading a Year Later". J. Experimental Psychology - Human Learning and Memory, 2, 1976, p. 554-565.

Landauer, T.K. "Cognitive Psychology and Computer System Design". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 10-35.

Landauer, T.K. "Research Methods in Human-Computer Interaction". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 905-928.

Landauer, T.K., Galotti, K.M. and Hartwell, S. "Natural Command Names and Initial Learning: A Study of Text-Editing Terms". Communications of the ACM, July 1983, 26(7), p. 495-503.

Lawrence, G. "A Synthesis of Learning Style Research Involving the MBTI". J. Psychological Type, 8, 1984, p. 2-15.

Ledgard, H., Whiteside, J.A., Singer, A. and Seymour, W. "The Natural Language of Interactive Systems". Communications of the ACM, October 1980, 23(10), p. 556-563.

Lee, D.M.S. "Usage Pattern and Sources of Assistance for Personal Computer Users". MIS Quarterly, December 1986, p. 313-325.

Lee, E., MacGregor, J. Lam, N. and Chao, G. "Keyword-menu retrieval: an effective alternative to menu indexes". Ergonomics, 29(1), 1986, p. 115-130.

Lewis, C. "A Model of Mental Model Construction". CHI Proceedings, April 1986, p. 306-313.

Lewis, C. and Norman, D.A. "Designing for Error". In R.M. Baecker and W.A.S. Buxton (Eds.), Readings in Human Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987, p. 627-638.

Long, J. "Effects of delayed irregular feedback on unskilled and skilled keying performance". Ergonomics, 19(2), 1976, p. 183-202.

Lotus, 1-2-3, Trademark, Lotus Development Corporation.

Lucas, Jr., H.C. and Nielsen, N.R. "The Impact of the Mode of Information Presentation on Learning and Performance". Management Science, October 1980, 26(10), p. 982-993.

Lucas, Jr., H.C., Walton, E.J. and Ginzberg, M.J. "Implementing Packaged Software". MIS/Quarterly, December 1988, p. 537-549.

Macaulay, L.A. and Norman, M.A. "Designing Interfaces for Different Types of Users - Experimental Critieria". Human-Computer Interaction, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1985, p. 643-647.

MacGregor, C. "Settling the WordPerfect/WordStar argument - almost". The Gazette, January 2, 1993.

Mack, R.L., Lewis, C.H. and Carroll, J.M. "Learning to use word processors: problems and prospects". In R.M. Baecker and W.A.S. Buxton (Eds.), Readings in Human-Computer Interaction: A Multidisciplinary Approach, Los Altos, Calif.: Morgan Kaufmann Publishers, Inc., 1987, p. 269-277.

Mack, R.L., Lewis, C.H. and Carroll, J.M. "Learning to use word processors: problems and prospects". ACM Transactions on Office Information Systems, 1(3), 1983, p. 254-271.

MacKnight, C.B. and Balagopalan, S. "An Evaluation Tool for Measuring Authoring System Performance". Communications of the ACM, October 1989, 32(10), p. 1231-1236.

Magers, C.S. "An experimental evaluation of on-line HELP for non-programmers". Proceedings of CHI'83 Conference: Human Factors in Computing Systems, 1983, p. 277-281.

Mahling, D.E. and Lefkowitz, L.S. "Using Task Context to Support an Adaptable Interface" In G. Salvendy and M.J. Smith (Eds.), Designing and Using Human-Computer Interfaces and Knowledge Based Systems, Amsterdan, Netherlands: Elsevier Science Publishers B.V., 1989, p. 565-573.

Malone, T.B., Kirkpatrick, M and Heasley, C "Human-Computer Interface Effectiveness Evaluation". Human-Computer Interaction, 1984, p. 117-121.

Mancini, G. "Commentary: Models of the Decision Maker in Unforeseen Accidents". Int. J. of Man-Machine Studies, 1987, 27, p. 631-639.

Markham, S.E. Veliyath, R. and Napoliello, M. "Defining and Measuring User Sophistication in MIS Research". Proceedings 1990 - Annual Meeting, Decision Sciences Institute, San Diego, November 19-21, 1990, p. 943-945.

Martin, M.P. "The Dominant Problem of Training Novice Users". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 14-1-14-11.

__________. "Adaptive Models". J. Systems Management, 19, October 1986, p. 23-29.

Martin, M.P. and Fuerst, W.L. "Effect of computer knowledge on user performance over time". Information and Software Technology, 1988, 30(9), p. 561-566.

__________________________. "Using computer knowledge in the design of interactive systems". Int.J. of Man-Machine Studies, 1987, 26, p. 333-342.

Maskery, H.S. "Adaptive Interfaces for Naive Users - An experimental study". Human-Computer Interaction, 1985, p. 343-349.

Matos, V.M. and Jalics, P.J. "An Experimental Analysis of the Performance of 4th Generation Tools and PCs". Communications of the ACM, November 1989, 32(11), p. 1340-1351.

Mayer, R.E. "Cognitive Aspects of Learning and Using a Programming Language". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 61-79.

Mayer, R.E. "From Novice to Expert". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 569-580.

Mayer, R.E. "The Psychology of How Novices Learn Computer Programming". Computing Surveys, March 1981, 13(1), 121-141.

Mawhinnev, C.H. "A Study of Computer Use by Knowledge Workers: User Satisfaction versus System Use". Proceedings 1990 - Annual Meeting, Decision Sciences Institute, San Diego, November 19-21, 1990, p. 954-956.

McDonald, J.E. Dearholt, D.W., Paap, K.R. and Schvaneldt, R. "A Formal Interface Design Methodology Based on User Knowledge". CHI Proceedings, 1986, p. 285-290.

McGrath, J.E. "Towards a Theory or Method for Research on Organizations". In W.W. Cooper, H.L. Leavitt and M.W. Shelly (Eds.), New Perspectives in Organizational Research, New York, N.Y.: John Wiley and Sons, 1964, p. 533-537.

McKendree, J. and Anderson, J.R. "Effect of Practice on Knowledge and Use of Basic LISP". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 236-259.

McKendree, J. and Carroll, J.M. "Advicing Roles of a Computer Consultant". CHI Proceedings, 1986, p. 35-40.

Meddis, R. Statistics using Ranks: a unified approach. N.Y., N.Y.: Basil Blackwell Publishing, 1984.

Merlin$_{(}$, Author P.F.Wade, Concordia University, Montreal, copyright, 1985.

Merrill, M.D., et.al. TICCIT. Englewood Cliff, N.J.: Educational Technology Publications, 1980.

Meyer, K., and Harper, M. "User Friendliness". MIS Quarterly, March 1984, p. 1-3.

Miller, G.A. "The Magical Number Seven plus or minus two:some limits on our capacity for processing information". Psychology Review, March 1956, 63(2), p. 52-60.

Mills, C.B., et. al., "Usability Testing in the Real World". CHI Proceedings, 1986, p. 212-215.

Millward, R.B. "Models of Concept Formation". In R.E.Snow, et.al. (Eds.), Aptitude,Learning and Instruction, Vol.2. Hillsdale,N.J.: Lawrence Erlbaum Associates, 1980, p. 245-275.

Minitab$_{(}$, Data Analysis Software, Release 7.1 - Standard Version, Copyright, Minitab, Inc., 1989.

Moray, N. "Intelligent aids, mental models, and the theory of machines". Int. J. of Man-Machine Studies, 1987, 27, p. 619-629.

Moses, F.L., Mendez, C.H. and Ehrenreich, S.L. "The effect of learning and context on abbreviation intelligibility". Paper presented at the 88th Annual Convention of the American Psychological Association, Montreal, September 1980.

Mosier, J.N. and Smith, S.L. "Application of Guidelines for Designing User Interface Software". Behaviour and Information Technology, 1986, 5(1), p. 39-46.

Muir, B.M. "Trust between humans and machines, and the design of decision aids". Int. J. of Man-Machine Studies, 1987, 27, p. 527-539.

Murphy, M. A Model and Tool for Testing the Learning of Cognition-Based Human-Computer Interfaces. Unpublished Doctoral Dissertation, McMaster University, Hamilton, Ontario, Canada, 1990.

Murray, R.P. and Abrahamson, D. "The effect of system response time delay variablilty on inexperienced videotex users". Behaviour and Information Technology, 2(3), 1983, p. 237-251.

Nakatani, L.H., et.al., "TNT: A Talking Tutor 'N' Trainer for Teaching the Use of Interactive Computer Systems". CHI Proceedings, April 1986, p. 29-34.

Napier, H.A., Lane, D.M., Batsell, R.R. and Guadango, N.S. "Impact on a Restricted Natural Language Interface on Ease of Learning and Productivity". Communications of the ACM, October 1989, 32(10), p. 1190-1198.

Nielsen, J. Usability Engineering. San Diego, Calif.: Academic Press, 1993.

_________. "Finding usability problems through heuristic evaluation". Proceedings of the ACM - CHI '92, May 1992, p. 373-379.

Nelson, R.R. and Cheney, P.H. "Training Today's User". Datamation, May 1987a, p. 121-122.

__________________________. "Training End Users: An Exploratory Study". MIS Quarterly, December 1987b, 11(4), p. 547-559.

Neter, J., Wasserman, W. and Kutner, M.H. Applied Linear Statistical Models. Homewood, Ill.: Richard D. Irwin, 1985.

Nickerson, R.S. "Why Interactive Computer Systems are Sometimes Not Used by People Who Might Benefit from Them". Int. J. of Man-Machine Studies, 23, 1981, p. 469-483.

Nievergelt, J. and Weydert, J. "Sites, modes and trails: Telling the user of an interactive system where he is, what he can do, and how to get to places". Readings in Human-Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987, p.

Nisbett, R.E. and Ross, L. Human inference: strategies and shortcomings of social judgment. Englewood Cliffs, N.J.: Prentice Hall, 1980.

Norman, D.A. "Cognitive Engineering - Cognitive Science". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 323-336.

__________. "Stages and Levels in Human-Machine Interaction". Int. J. of Man-Machine Studies, 1984, 21, p. 365-375.

__________. "Design Rules Based on Analyses of Human Error". Communications of the ACM, April 1983, 26(4), p. 254-258.

__________. "Discussion: Teaching,Learning and the Representation of Knowledge". In R.E. Snow, et.al. (Eds.), Aptitude,Learning and Instruction,Vol.2. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1980, p. 237-244.

Ogden, W.C., Isa, B.S., Wolfe, S. and Korenshtein, R. "Human factors simulation of QMF navigation". In K. Paap and R. Roske-Hofstrand, "Designing of Menus", in M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers, B.V. 1988, p. 205-235.

Ogozalek, V.Z. and Van Praag, J. "Comparison of Elderly and Younger Users on Keyboard and Voice Input Computer-based Composition Tasks". CHI Proceedings, April 1986, p. 205-211.

Oliver, B. Production, Implementation and Evaluation of Interactive Videodisks for Education. Montreal, Quebec: Educational Technology Conference, Concordia University, November 23, 1984.

Olson, J.R. "Cognitive Analysis of People's Use of Software". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 260-293.

Osin, L. "SMITH: How to produce CAI courses without prompting". Int. J. of Man-Machine Studies, 1976, 8, p. 207-241.

Paap, K. and Roske-Hofstrand, R. "Designing of Menus". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 205-235.

Parasuraman, S. and Igbaria, M. "An Examination of Gender Differences in the Determinants of Computer Anxiety and Attitudes toward Microcomputers among Managers". Int. J. of Man-Machine Studies, 1990, 32, p. 327-340.

Pask, G. "Styles and Strategies of Learning". Br. J. Educational Psychology, 1976, 46, 128-148.

Pask, G. and Scott, B.C.E. "Learning Strategies and Individual Competence". In J.M. Whitehead (Ed.), Personality and Learning. London: Hodder and Stoughton, 1975, p. 257-287.

Pask, G. and Scott, B.C.E. "Learning Strategies and Individual Competence". Int. J. of Man-Machine Studies, 4, 1972, p. 217-253.

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

Perlman, G. "Making the right choices with menus". Human-Computer Interaction, 1984, p. 317-321.

Polson, P.G. "A Quantitative Theory of Human-Computer Interaction". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 184-235.

Polson, P.G. and Kieras, D.E. "A Quantitative model of the learning and performance of text-editing knowledge". Proceedings of CHI'85: Human Factors in Computing Systems, New York: ACM, 1985, p. 207-212.

Polson, P.G., Muncher, E. and Engelbeck, G. "A Test of a Common Elements Theory of Transfer". CHI Proceedings, 1986, p. 78-83.

Porter, M.A. "The role of the statistician in industry". The Statistician, 42(3), 1993.

Potosnak, K.M., et. al., "Classifying Users: A Hard Look at Some Controversial Issues". CHI Proceedings, 1986, p. 84-88.

Preece, J. and Keller, L. (Eds.). Human-Computer Interaction. Hemel Hempstead:Prentice Hall, 1990.

Pressey, S. "A simple apparatus which gives tests, and scores and teaches". School and Society, March 20, 1926, 23, p. 373-376.

Price, L.A. "Using Office Documentation Online". Communications of the ACM - SIGSOC Bulletin, 1981, 12, p. 15-20.

Quattro Pro®, Trademark, Borland International.

Quinn, L and Russell, D.M. "Intelligent Interfaces: User Models and Planners". CHI Proceedings, April 1986, p. 314-320.

Ramaprasad, A. "Cognitive Process as a Basis for MIS and DSS Design". Management Science, 1987, 33(2), p. 139-148.

Rasmussen, J. and Andersen, H.B. (Eds.) Human-Computer Interaction: Research Directions in Cognitive Science. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1991.

Rathswohl, E.J. "Applying Hermeneutics to Human Factors in MIS". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 20-1-20-6.

Raub, A.C. Correlates of computer anxiety in college students. Unpublished Doctoral Dissertation, University of Pennsylvania, Philadephia, 1981.

Reid, S. Memo: Radio Shack - Courseware Standards. Montreal, Quebec: Holt,Reinhart and Winston of Canada Ltd., November 24, 1982.

Reisner, P. "Discussion: HCI, What Is It and What Research is Needed?". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 337-352.

Relles, N. The design and implementation of user-oriented systems. Unpublished Doctoral Dissertation, Technical Report 357, University of Wisconsin, Madison, WI, July 1979.

Relles, N. and Price, L. "A User Interface for Online Assistance". Proceedings of the fifth Internation Conference on Software Engineering, March 1981, p. 400-408.

Relles, N., Sondheimer, N.K. and Ingargiola, G.P. "Recent Advances in User Assistance". ACM SIGSOC Bulletin, 1981, 12, p. 1-5.

Remington, W. "Interface Preference vs. User Experience Level: an Empirical Investigation". Proceedings 1990 - Annual Meeting, Decision Sciences Institute, San Diego, November 19-21, 1990, p. 1253-1255.

Rigney, J.W. "Cognitive Learning Strategies and Dualities in Information Processing". In R.E., Snow, et.al. (Eds.), Aptitude, Learning, and Instruction, Vol.1. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1980, p. 315-343.

Rissland, E.L. "Ingredients of Intelligent User Interfaces". Int. J. of Man-Machine Studies, 1984, 21, p. 377-388.

Rivard, S. and Huff, S.L. "An Empirical Study of Users as Application Developers". Information Management, 1985, 8, p. 89-102.

Rizzo, A. Bagnara, S. and Visciola, M. "Human Error Detection Processes". Int. J. of Man-Machine Studies, 1987, 27, p. 555-570.

Roberts, T.L. "Text-editors". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 655-672.

Roberts, T.L. and Moran, T.P. "The Evaluation of Text-Editors: Methodology and Empirical Results". In R.M. Baecker, and W.A.S. Buxton (Eds.), Readings in Human-Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987, p. 250-268.

Roberts, T.L. and Moran, T.P. "The Evaluation of Text-Exitors". Communications of the ACM, 26, p. 265-283.

Rockart, J.F. and Flannery, L.S. "Tne Management of End User Computing". Communications of the ACM, October 1983, 26(10), p. 776-785.

Roemer, J.M. and Chapanis, A. "Learning performance and attitudes as a function of the reading grade level of a computer-presented tutorial". Proceedings of Human Factors in Computing Systems, 1982, p. 239-244.

Romiszowski, A.J. Designing Instructional Systems. London: Kogan Page, 1981.

Rosenberg, J. "Teacher's Aide". PC Mag, April 16, 1985, p. 269-270.

Roth, E.M., Bennett, K.B. and Woods, D.D. "Human Interaction with an 'Intelligent' Machine". Int. J. of Man-Machine Studies, 1987, 27, p. 479-525.

Rouse, W.B. "Design and Evaluation of Computer-Based Decision Support Systems". In G. Salvendy, (Ed.), Human-Computer Interactions, North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1984, p. 229-245.

Rubinstein, R. and Hersh, H. The Human Factor. Mass.: Digital Press, 1984.

Rushinek, A., Rushinek, S.F. and Stutz, J. "A methodology for interactive evaluation of user reactions to software packages: an empirical analysis of system performance, interaction, and run time". Int. J. of Man-Machine Studies, 1984, 20, p. 169-188.

Salomon, G. "Transcending the Qualitative-Quantitative Debate: The Analytic and Systemic Approaches to Educational Research". Educational Researcher, 1991, 20(6), p. 10-18.

Savage, R.E. and Habinek, J.K. "A Multilevel Menu-Driven User Interface: Design and Evaluation through Simulation". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. N.J.: Ablex Publishing, 1982, p. 165-185.

Schlager, M.S. and Ogden, W.C. "A Cognitive Model of Database Querying: A Tool for Novice Instruction". CHI Proceedings, April 1986, p. 107-113.

Schneider, M.L. "Models for the Design of Static Software User Assistance". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction, N.J.: Ablex Publishing, 1982, p. 137-148.

Schneier, C.A. and Mehal, M.E. "Evaluating Usability of Application Interfaces". In G. Salvendy (Ed.), Human-Computer Interaction. New York, N.Y.: Elsevier Publishers, 1984, p. 129-132.

Schott, F. and Olson, M. "Designing Usability in Systems: Driving for Normalcy". Datamation, May 15, 1988, p. 68-76.

Schroeder, D.L. and Kletke, M. "The Development and Validation of an Instrument for Assessing Relative Expertise Levels among Microcomputer End Users". Proceedings of 1990 Annual Meeting - Decision Sciencs Institute, November 19-21, 1990, p. 937-939.

Scott, W.R. "Field Methods in the Study of Organizations". In J.G. March (Ed.), Handbook of Organizations. Chicago: Rand McNally, 1960, p. 261-303.

Scriven, M. "The Evaluation of Hardware and Software". Studies in Educational Evaluation, 1990, 16, p. 3-40.

Sein, M.K., et.al. "Training End Users to Compute: Cognitive, Motivational and Social Issues". INFOR, 1987, 25(3), p. 236-255.

Sekaran, U. Research Methods for Managers: A Skill-Building Approach. Toronto, Canada: John Wiley & Sons, Inc., 1984.

Seymour, J. "Do Graphical Results Require a Graphical Interface?". PC Magazine, January 15 1991, p. 87.

Shackel, B. "Human Factors and Usability". In J. Preece and L. Keller (Eds.), Human-Computer Interaction, 1990, p. 27-41.

Sheppard, S.B., Bailey, J.W. and Bailey, E.K. "An Empirical Evaluation of Software Documentation Formats". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. N.J.: Ablex Publishing, 1982, p. 135-163.

Shneiderman, B. Designing the User Interface. Don Mills, Ontario: Addison-Wesley, 1987.

____________. "Seven Plus or Minus Two: Central Issues in Human-Computer Interaction". CHI Proceedings, 1986, p. 343-349.

____________. "Correct, Complete Operations and Other Principles of Interaction". Human-Computer Interaction, 1984, p. 135-147.

____________. "System Message Design: Guidelines and Experimental Results". In A. Badre and B. Shneiderman (Eds.), Directions in Human-Computer Interaction. New Jersey, N.Y.: Ablex Publishing, 1982, p. 55-78.

Shneiderman, B. and Kearsley, G. Hypertext Hands-On!. Don Mills, Ontario: Addison-Wesley Publishing, 1989.

Shrager, J. and Klahr, D. "Instructionless learning about a complex device: the paradigm and observations". Int. J. of Man-Machine Studies, 1986, 25, p. 153-189.

Shuell, T.J. "Learning Theory,Instructional Theory,and Adaptation". In R.E. Snow, et.al., (Eds.), Aptitude, Learning,and Instruction,Vol.2. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1980, p. 277-302.

Shutoh, T., Tsurata, S., Kawai, R. and Shutoh, M. "Voice operation in CAD system". In H.W. Hendrick and O. Brown, Jr. (Eds.), Human Factors in Organizational Design and Management, North Holland, Amsterdam: Elsevier Science Publishers, B.V., 1984, p. 205-209.

Silver, M.S. "Decisional Guidance for Computer-Based Decision Support". MIS Quarterly, March 1991, p. 104-122.

Simonsen, R.H. and Renshaw, K.S. "CAI - Boon or Boondoggle". Datamation, March 1974, p. 90-102.

Singley, M.K. and Anderson, J.R. "The Transfer of Text-editing Skill". Int. J. of Man-Machine Studies, 1985, 22, p. 403-423.

Smartware II,, Trademark, Informix Software Inc., 16011 College Blvd., Lenexa Kansas, 66219, U.S.A.

Smith, D. "Faster is better: A business case for subsecond response time". Computerworld, April 18, 1983, p. 1-11.

Smith, S.L. "Standards versus Guidelines for Designing User Interface Software". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 877-889.

Smith, S.L. and Mosier, J.N. "Guidelines for Designing User Interface Software". Bedford, MA.: Mitre Corporation, 1986.

____________________. "The User Interface to Computer-Based Information Systems: A Survey of Current Software Design Practice". Behaviour and Information Technology, 1984, 3(3), p. 195-203.

Spohrer, J.C. and Soloway, E. "Novice Mistakes: Are the Folk Wisdoms Correct?". Communications of the ACM, July 1986, 29(7), p. 624-632.

Stabell, C.B. Individual difference in managerial decision making processes:A study of conversational computer usage. Unpublished Doctoral Dissertation, MIT, 1974.

Stephenson, W.R. "A Study of Student Reaction to the Use of Minitab in an Introductory Statistics Course". The American Statistician, 1990, 44(3), p. 231-235.

Sterman, J.D. "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment". Management Science, March 1989, 35(3), p. 321-339.

Supercalc3$_{(r)}$, Trademark, Sorcim Corporation.

Tallon, J.L. "Computer-Assisted Instruction and the University Library". Canadian Journal of Information Science, 1982, 7, p. 49-55.

Teng, J.T.C. and Jamison, W. "User Evaluation of Database Query Languages: A Comparison of SQL and DBASE III". INFOR, 1990, 28(3), p. 282-296.

Thomas, J.C. and Carroll, J.M. "Human factors in communication". IBM Systems Journal, 20, p. 237-263.

Thomas, J.D.E. Instructive Guidance for Users of Business Software. Unpublished Theory Paper, Concordia University, Montreal, 1989.

Todd, P. and Benbasat, I. "An Experimental Investigation of the Impact of Computer Based Decision Aids on Decision Making Strategies". Information Systems Research, 1991, 2(2), p. 87-115.

Tombaugh, J.W. and McEwen, S.A. "Comparison of two information retrieval methods on videotex: Tree-structure versus alphabetic directory". Proceedings of Human Factors in Computing Systems, 1982, p. 106-110.

Travers, R.M.W., (Ed.). Essentials of Learning - 4th edition. N.Y.: MacMillan Publishing, 1977.

_______________. Essentials of Learning - 2nd edition. N.Y.: MacMillan Publishing, 1975.

_______________. Essentials of Learning - 1st edition. N.Y.: MacMillan Publishing, 1974.

Trumbly, J. "Triggering Methods for Adaptive User Interface". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 1-1-1-8.

Tullis, T.S. "Screen Design". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 377-411.

Van Dyke, C. "Taking 'Computer Literacy' Literally". Communications of the ACM, 1987, 30(5), p. 366-374.

Van Lehn, K. "Toward a Theory of Impasse-Driven Learning". In H. Mandl and A. Lesgold, (Eds.), Learning issues for intelligent tutoring systems. N.Y.: Springer-Verlag, 1988, p. 19-40.

Vassilou, Y., et.al. "Natural Language for Database Queries: A Laboratory Study". MIS Quarterly, December 1983, p. 47-61.

Vaubel, K.P. and Gettys, C.F. "Inferring User Expertise for Adaptive Interfaces". Human-Computer Interaction, 1990, 5, p. 95-117.

Vincente, K.J. and Williges, R.C. "Accommodating individual differences in searching a hierarchical file system". Int. J. of Man-Machine Studies, 1988, 29, p. 647-668.

Waern, Y. "Learning Computerized Tasks as Related to Prior Task Knowledge". Int. J. of Man-Machine Studies, 1985, 22, p. 441-455.

Wager, W. "Design Considerations for Instructional Computing Programs". J. of Educational Technology Systems, 1981-82, 10, p. 261-269.

Walther, G.H. and O'Neil, H.F. "On-line user-computer interface - the effects of interface flexibility, terminal type, and experience on performance". National Computer Conference, 1974, p. 379-384.

Watley, C. and Mulford, J. A comparison of commands' documentation: online versus hardcopy. Unpublished student project, University of Maryland, December 1983.

Weiner, B. An attributional theory of motivation and emotion. N.Y.: Springer-Verlag, 1986.

Wetherill, G.W. and Curram, J.B. The Design and Evaluation of Statistics Software for Microcomputers. Universtiy of Kent at Canterbury, Applied Statistics Research Unit, 1984.

Whiteside, J., Bennett, J. and Holtzblatt, K. "Usability Engineering: Our Experience and Evolution". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 791-817.

Whiteside, J. and Wixon, D. "Discussion: Human-Computer Interaction - A Quest for Cognitive Science". In J.M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, Mass.: Bradford Book, MIT Press, 1989, p. 353-365.

Whiteside, J., Jones, S., Levy, P.S. and Wixon, D. "User performance with command, menu, and iconic interfaces". Proceedings of CHI'83, New York: ACM, 1985, p. 144-148.

Wickelgren, W.A. "Memory Storage Dynamics". In W.K. Estes (Ed.), Handbook of Learning and Cognitive Processes, 4, Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976, p. 321-361.

Williges, R.C., Elkerton, J., Pittman, J.A. and Cohill, A.M. "Providing Online Assistance to Inexperienced Computer Users". Human-Computer Interaction, 1985, p. 765-769.

Wilson, J. R. and Rutherford, A. "Mental Models: Theory and Applications in Human Factors". Human Factors, 1989, 31(6), p. 617-634.

Winston, P.H. "Learning and Reasoning by Analogy". Communications of the ACM, December 1980, 53(12), p. 689-702.

Witkin, H.A. "Some implications of research on cognitive style for problems of education". In J.M. Whitehead (Ed.), Personality and Learning. London: Hodder and Stoughton, 1975, p. 288-314.

Witkin, H.A. "Some implications of research on cognitive style for problems of education". In M.B. Gottesgen and G.B. Gottesgen (Eds.), Professional School Psychology, 3, New York: Grune and Stratton, 1969, p. 198-227.

Wittrock, M.C. "Students' Thought Processes". In R.M.W. Travers (Ed.), Second Handbook of Research on Teaching. Chicago: Rand McNally College Publishing, 1974, p. 297-314.

Wolf, C.G., Rhyne, J.R. and Ellozy, H.A. "The Paper-Like Interface". In G. Salvendy and M.J. Smith (Eds.), Designing and Using Human-Computer Interfaces and Knowledge Based Systems, Amsterdan, Netherlands: Elsevier Science Publishers B.V., 1989, p. 494-501.

WordPerfect, 5.0, Wordperfect Corporation, Orem, Utah, Copyright 1988.

Wright, P. "Issues of Content and Presentation in Document Design". In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York, N.Y.: Elsevier Science Publishers B.V., 1988, p. 629-652.

Wright, P. "Manual Dexterity: A User-Oriented Approach to Creating Computer Documentation". In R.M. Baecker and W.A.S. Buxton (Eds.), Readings in Human Computer Interaction: A Multidisciplinary Approach. Los Altos, Calif.: Morgan Kaufmann Publishers, 1987, p. 613-620.

Young, L.F. "Computer Support of the Creativity Factor in Systems Design: The case of the Air Force Weapons Systems". Proceedings of the second Symposium on Human Factors in MIS, Sacramento, California, 1989, p. 3-1-3-12.

Ziegler, J.E., Hoppe, H.U. and Fahnrich, K.P. "Learning and Transfer for Text and Graphics Editing with a Direct Manipulation Interface". CHI Proceedings, 1986, p. 72-77.

Zinn, K.L. "Requirements for Effective Authoring Systems and Assistance". Int. J. of Man-Machine Studies, 1974, 6, p. 381-400.

Appendices

Appendix A

Survey Kit sent to Expert Panel in Study 1

Concordia
UNIVERSITY

Covering Letter

August 23, 1991

Dear Participant,

Attached is a Software Evaluation tool which is part of a doctoral dissertation aimed at developing a methodology for the evaluation of commercial software which are used in business for decision making purposes. It would be greatly appreciated if you would spend some time to assist in this project. Instructions for completing the task are given overleaf. If there are features and/or learning factors which you consider are essential, but have been left off of the instrument, please feel free to add them on a separate sheet and include them in your ranking considerations. The task is aimed at eliciting personal perceptions, so should be done without reference to others. Please use the definitions and descriptions of features and learning factors, which have been provided, as a guide when filling in the evaluation instrument. Could you also provide the background information asked for on the last two pages.

If you wish to receive a summary of my findings please write your full name and address on this sheet and return it with your completed form and I would be pleased to do so. Would you also indicate whether you would be willing to do a follow-up at a later date. Thank you for your time and effort.

Sincerely,

Jennifer Thomas
Ph.D. Candidate
Dept. DS/MIS
GM-209-9

1455, de Maisonneuve Blvd. W
Montreal, Quebec H3G 1M8

Definitions and Descriptions of Features and Learning Factors

Learning Factors:

Speed - time required to perform an operation
Memory - support offered to recall, retention over time
Effort - cognitive complexity, mental strain required to perform operation
Comfort - impact on anxiety, peace of mind, aesthetic appeal

Design Features:

Interface Command Structure - menus, commands, direct manipulation, natural language
Depth of Structure - levels required to perform an operation
Logic of Structure - makes sense, is understandable
Consistency - not ambiguous or contradictory in conventions, functions, messaging
Screen Design - cosmetics/aesthetics - highlighting, animation, layout, etc.
- grouping meaningful - placement on screen logical with respect to purpose
- language/wording - clear, understandable, inoffensive

Flexible - more than one way to perform an operation
Error Trapping/Recovery - warning and/or correction of errors outside parameter ranges, or of "o","l" (letters) to "0","1" (numericals) in data input
System Response Time - elapse time between user input and response from system
Data Input - full-screen editor or line editor
Macros - automation of frequently used string of operations
Autosave/Backup - inputs intermittently saved by system
Auto-adjustment - changes in input reflected automatically in analyses and graphs
Auto-formatting - automatic formatting of input into decimal places, scientific notation, currency,etc.

Assistance Features:

Manuals - on and offline manuals explaining package features and how to use them
System Messages - messages generated by system due to unanticipated entries
Tutorials - on and offline training modules on how to use the package
Prompts - warning of errors made, how to correct and why occurred
Keyboard Templates - on and offline memory jogs of function keys associated with operations
Defaults - expected or anticipated responses supplies which are modifiable
Examples - examples of correct or valid operations and actions, could also include non-examples
Index of commands and operations - quick reference to commands and operations
Glossary - definitions of terms and commands
Unsolicited Help - help supplied at system initiation after prolonged delay in input
Cautions - warnings against overwriting files, deleting file, etc.
Checklist & Memory Jogs - traces of most recent or most used operations
Navigational Aids - trees, maps, control keys
Instructive Feedback - input assessed for correctness and communicated to user with possible action
Context Help - help related to place error made
Expertise Accommodation - accommodation of novice to expert transition
Restricted Options - restricted to certain operations at certain junctures
Subject Matter Aid - assistance with subject matter concepts, e.g. statistics, accounting principles
Conceptual Models - mental models of the system and how it works

SOFTWARE EVALUATION

Factors important for learning in general, and for learning of software packages can be identified as those which:

a. support **Speed**, that is reduce performance time;
b. alleviate **Memory** load, that is promote retention or not require it;
c. minimize **Mental effort**, that is reduce mental strain;
d. provide **Psychological comfort**, that is reduce anxiety, promote peace of mind.

The purpose of this task is to formulate a mapping of software design and assistance features with these learning factors. You are asked to do this in three stages:

1. Assign a weighting out of 100% to each of the learning factors, as defined above - speed, memory, effort, comfort - to indicate the expected importance they have for your own learning of a package. Equal weighting is permitted but weights must add to 100%. Place weights in the brackets below the learning factors.

2. Rank the design features listed below, from 1 to 12, then the assistance features following, from 1 to 19, according to importance for your learning and use. Place your assigned rank in the rank column. Equal ranking is permitted.

3. Match these features with the associated learning factor(s) you expect to be supported by the inclusion of the particular feature in software by placing an 'X' in the box.

e.g.	Speed	Memory	Effort	Comfort
Feature 1	X		X	
Feature 2		X	X	X
Etc.				

Design Features:	Rank	Speed ( )	Memory ( )	Effort ( )	Comfort ( )
1. Interface Command Structure - menus,commands, etc.					
2. Depth of Structure					
3. Logic of Structure					
4. Consistency					
5. Screen Design - cosmetic/aesthetics - meaningful grouping - language, wording					
6. Flexible					
7. Error Trapping/Recovery					
8. System Response Time					
9. Data Input - full-screen, line editor					
10. Macros					
11. Autosave/backup					
12. Auto-adjustment					
13. Auto-formatting					

Assistance Features:	Rank	Speed ( )	Memory ( )	Effort ( )	Comfort ( )
1. Manuals					
2. System Messages					
3. Tutorials					
4. Prompts					
5. Keyboard Templates					
6. Defaults					
7. Examples					
8. Index of commands/operations					
9. Glossary					
10. Unsolicited Help					
11. Cautions					
12. Checklists & Memory Jogs					
13. Navigational Aids					
14. Instructive Feedback					
15. Context Help					
16. Expertise Accommodation					
17. Restricted Options					
18. Subject Matter Aid					
19. Conceptual Models					

PROFICIENCY QUESTIONNAIRE

Name: *Age: <25 25-34 35-44 44+* *Sex:*

Major in B.Comm.: *Other Degrees & Majors:*

 # Years Work Experience: *Area of Major Work Experience:*
Current Job & Title:

Have you used a mainframe computer before? *Yes* *No*

Have you used a microcomputer before? *Yes* *No*

If yes, IBM, MacIntosh or other?

Place a tick beside the category which best describes your level of expertise with computers
( ) Expert *( ) Intermediate* *( ) Novice*

How often do you use a computer?
( ) Never *( ) Less than once per month* *( ) Once per month*
( ) Few times per month *( ) Few times per week* *( ) Once per day* *( ) Several times per day*

Place a tick beside the category which best describes your level of expertise with DOS
( ) Expert *( ) Intermediate* *( ) Novice*

How often do you use DOS?
( ) Never *( ) Less than once per month* *( ) Once per month*
( ) Few times per month *( ) Few times per week* *( ) Once per day* *( ) Several times per day*

If you have used DOS before, what are the commands to accomplish the following:

Create a directory?
Copy a file from one disk drive to another?
Access a directory different from the one you are currently in?
Print a file onto the screen?
List all the files on a diskette in pages?
Change the name of a file on a diskette?

Circle the choice that best describes your reaction to each statement following:

1 = Strongly disagree *2 = Disagree to some extent* *3 = Uncertain*
4 = Agree to some extent *5 = Strongly agree*

I am confident that I could learn computer skills	5 4 3 2 1
I am unsure of my ability to learn a computer programming language	5 4 3 2 1
I will be able to keep up with important technological advances in computers	5 4 3 2 1
I feel apprehensive about using a computer terminal	5 4 3 2 1
If given the opportunity to use a computer, I am afraid that I might damage it	5 4 3 2 1
I have avoided computers because they are unfamiliar to me	5 4 3 2 1
I hesitate to use a computer for fear of making mistakes that I cannot correct	5 4 3 2 1
I am unsure of my ability to interpret a computer printout	5 4 3 2 1
I have difficulty understanding most technological matters	5 4 3 2 1
Computer terminology sounds like confusing jargon to me	5 4 3 2 1

Beside each of the software packages listed below indicate:

Your level of **expertise** (E)*xpert* (I)*ntermediate* (N)*ovice*
Your **frequency** *of use* (1)*Never* (2)*Less than once a month* (3)*Once a month* (4)*Few times a month* (5)*Few times a week* (6)*Once a day* (7)*Several times a day*
Operation *of the package* (M)*enu* (C)*ommand* (Mi)*xed* (Mo)*use*
Purpose *for which used* (W)*ork* (Sc)*hool* (P)*ersonal*
Where **learnt** (Co)*urse* *on your* (O)*wn*

Write the letter or number associated with the above five categories beside each of the packages listed below as shown in the example. Add any which are not mentioned.

	EXPERTISE	FREQUENCY	OPERATION	PURPOSE	LEARNT
E.g. *Spreadsheet* *-Lotus* *-SuperCalc*	 *I* *N*	 2 1	 *Mi* *Mi*	 *Sc* *P*	 *Co* *O*
Spreadsheet *-Lotus* *-SuperCalc* *-Multiplan* *-Quatro* *-Other* *(Specify)*					
Wordprocessor *-Wordperfect* *-Wordstar* *-Multimate* *-Other* *(Specify)*					
Database *-DBase* *-Smart* *-Framework* *-Other* *(Specify)*					
Programming Languages *-Basic* *-Fortran* *-Pascl* *-Other* *(Specify)*					
Statistical *-Minitab* *-Merlin* *-SPSS* *-SAS* *-Other* *(Specify)*					

How would you rate your competence in quantitative-type courses? ( ) *Excellent* ( ) *Average* ( ) *Poor*

Appendix B

Comparison of Merlin, and Minitab, Package Features

	Merlin,	Minitab,
FUNCTION	Statistics	Statistics
INTERACTION	menu or command	command
INPUT STRUCTURE:		
Accessing	menu or command	command
Input Mode	line entry	full screen or line entry
End Entry	keystroke	command
Labelling	menu or command	command
Printing	menu or command	command
EDIT STRUCTURE:		
Accessing	command	keystroke or command
Input Mode	line	full screen or line entry
Change Data	command	full screen or command
Add Data	command	full screen or command
PROCESSING:		
Regression	mixed	command
Plotting	menu or command	command
SAVE OUTPUT:		
Data	automatic or command	command
Analysis	command	command
EXIT:	command	command

Comparison of Merlin, and Minitab, Assistance Features

HELP STRUCTURE:	Merlin,	Minitab,
Examples	Yes	Yes
Tutorials	No	No
Manuals	Yes	Yes
System Messages	Yes	Yes
Input Error Trapping	Yes	Yes
Input Error Correction	Yes	No
Error Recovery	Yes	No
Index	Yes	Yes
Default Values	Yes	No
Keyboard Templates	No	No
Checklists	No	No
Memory Jogs	Yes	No
Expertise Accommodation	Yes	No
Context Help	Partial	No
Command Assistance	Yes	Yes
Prompting	No	No
Question or Prompt Assistance	Yes	No
Menu Assistance	No	No
Unsolicited Help	No	No
Navigational Aids	Yes	No
Feedback	No	No

Comparison of Merlin$_t$ and Minitab$_t$ Data Entry Task Requirements

Unit Tasks	Merlin$_t$	Minitab$_t$
Data Entry:		
Accessing	Menu or DATA	READ or SET column nos.
Input Mode	Line entry	Full-screen or line entry
End Entry	carriage return	END
Labelling	Menu or NAME	NAME columns
Print	Menu or PRIN	PRINT column nos.
Editing:		
Accessing	EDIT	LET or Esc to Full-screen
Change Data	R ROW15; enter value R el15(1); enter value R el15(2); enter value	Full-screen or Let C1(15) = value Let C2(15) = value
Add Data	A COL4 = 1.1 * COL3	Full-screen or LET C4 = 1.1 * C3
Output:		
Save Data	automatic or SAVE	SAVE 'filename'
Exit:	S or STOP Bye Done Quit Off	STOP

Appendix C

Sample Printout of Data Entry Task on Minitab

```
                          M I N I T A B

                     Data Analysis Software
                Release 7.1 -- Standard Version
               Copyright(C)   Minitab, Inc. 1989

                 This software is licensed to:
                Commerce and Administration - FIS

                   Serial # 711-0001-000452

You may use Minitab under the terms of the License Agreement enclosed with this
program; please read it.  This License entitles: a) one user to run this copy
of Minitab on any number of computers; b) more than one user to run this copy
of Minitab on a single computer, BUT it is a violation of the License to run
this copy of Minitab on more than one computer simultaneously.  Government
users see HELP FGU.
                                          Press any key to continue. MINITAB Rel
Standard Version *** Storage Available: 16179


Use the ESCape key to toggle between Minitab and the Data Editor

   MTB > read c1 c2
   DATA> 50 69
   DATA> 66 85
   DATA> 73 88
   DATA> 84 70
   DATA> 57 84
   DATA> 83 78
   DATA> 76 90
   DATA> 95 97
   DATA> 73 79
   DATA> 78 95
   DATA> 48 67
   DATA> 53 60
   DATA> 54 79
   DATA> 79 79
   DATA> 88 76
   DATA> 90 98
   DATA> 60 56
   DATA> 89 87
   DATA> 83 91
   DATA> 81 86
   DATA> end
        20 ROWS READ
   MTB > name c1='Test1' c2='Test2'
   MTB > print c1-c2
```

```
 ROW  Test1  Test2

   1     50     69
   2     66     85
   3     73     88
   4     84     70
   5     57     84
   6     83     78
   7     76     90
   8     95     97
   9     73     79
  10     78     95
  11     48     67
  12     53     60
  13     54     79
  14     79     79
  15     88     76
  16     90     98
  17     60     56
  18     89     87
  19     83     91
  20     81     86

MTB > save 'a:demo'

Worksheet saved into file: a:demo.MTW
MTB > let c1(15)=55
MTB > print c1-c2

 ROW  Test1  Test2

   1     50     69
   2     66     85
   3     73     88
   4     84     70
   5     57     84
   6     83     78
   7     76     90
   8     95     97
   9     73     79
  10     78     95
  11     48     67
  12     53     60
  13     54     79
  14     79     79
  15     55     76
  16     90     98
  17     60     56
  18     89     87
  19     83     91
  20     81     86

MTB > stop
*** Minitab Release 7.1 *** Minitab, Inc. ***
Storage available 16179

C:\DOS>
```

```
MTB > help help

The HELP facility in Minitab helps you learn about Minitab.

A.  For general information about Minitab, type:

              HELP OVERVIEW

You will be given a choice of topics, including  1. Introduction
to Minitab,  2. Syntax of Commands,  3. Subcommands.

B.  To see what commands Minitab has, type:

              HELP COMMANDS

You will be given a choice of categories, including  1. General
Information,  2. Input,  3. Output,  4. Editing and Manipulating
Data.


Do you wish to see more information on the HELP facility in
Minitab?  When Minitab types MORE?, type Y if you do, and N if
you do not.
n
MTB > help commands

To get a list of the Minitab commands in one of the categories
below, type HELP COMMANDS followed by the appropriate number,
(HELP COMMANDS 1, for example).

 1 General Information           11 Tables
 2 Input and Output of Data      12 Time Series
 3 Editing and Manipulating      13 Statistical Process Control
   Data                          14 Exploratory Data Analysis
 4 Arithmetic                    15 Distributions & Random Data
 5 Plotting Data                 16 Sorting
 6 Basic Statistics              17 Matrices
 7 Regression                    18 Miscellaneous
 8 Analysis of Variance          19 Stored Commands and Loops
 9 Multivariate Analysis         20 How Commands are Explained
10 Nonparametrics                   in HELP
```

```
MTB > help commands 2

COMMANDS 2. Input and Output of Data

"Data Editor" (full screen data entry and editing)
READ       (enter data from terminal or data file)
SET        (enter data from terminal or data file)
INSERT     (enter data from terminal or data file)
END        (end input entered with READ, SET, INSERT, and STORE)
RETRIEVE   (enter data from a saved worksheet file)
NAME       (name columns)
PRINT      (view data on the screen)
WRITE      (output data to screen or data file)
SAVE       (output data to a saved worksheet file)
MTB > print
* ERROR * 0 is too few arguments
```

Sample Screen of Minitab's Online Help for Data Entry of one column at a time

```
MTB > set c1 c2
* ERROR * 2 is an illegal number of arguments

MTB > set c1
DATA> 3
DATA> 4
DATA> 5
DATA> 6
DATA> end
MTB > set c2
DATA> 6
DATA> 7
DATA> 8
DATA> 9
DATA> end
MTB > regress c1 c2

* ERROR * 2 is too few arguments

MTB > list
* ERROR * Name not found in dictionary


MTB > help

You are in Minitab (Standard Microcomputer Version).


Minitab is a general purpose statistics package.


There is a HELP facility that helps you learn about Minitab.
To see how it works, type:

                  HELP HELP


To leave Minitab, type:

                  STOP
```

Appendix D

Sample Printout of Data Entry Task on Merlin,
Sample Screens of Merlin's Online Help

```
MERLIN* V4.2
(C) Copyright P.F.Wade 1972, 1985
 *  Reg'd Cdn. & U.S. Trademark of CP

MER1

Ready


AUTO BACKUP on file BACKUP.MER every 15 min.

Do you wish operating instructions? y or n
= y

MERLIN is a data analysis package for:
                    data extraction
                    regression
                    plotting
                    sorting
                    forecasting
                    histograms
                    analysis of variance
                    stat. tests
                    control charts
                    data modification, etc.

Entered data (numbers only) are stored as a 2-dimensional array
in a memory workspace with a maximum of  4000 elements.
Column labels are entered separately.

Selected 4-letter commands (e.g. sort) are entered in response
to the prompt "ENTER COMMAND".  MERLIN through a question-and-
answer sequence identifies the user's requirements, executes
them and returns to "ENTER COMMAND".

Commands are divided into two programs, MER1 and MER2, as indicated in the
catalogue obtained by typing "cata" or "?" in response to "ENTER COMMAND".
Pause.
Please press <return> to continue.

Data Input
Enter data when requested, or use the command "data".
Data can be entered from the keyboard or read from a file.
Numbers are entered a row at a time.

Saving data
Use the command "save" to save data. Data are saved
automatically every 15 minutes in a file BACKUP.MER.

Saving output (ie displays such as plots or tables)
Output is normally directed to the screen.
Use the command "file" to save it as a printable file.

Special conventions.
The user can return to "ENTER COMMAND" at any time by typing "r",
can back up one question by typing "b",
or can terminate the run by typing "s".

HELP
```

```
Where indicated, type "?" for explanation of prompts.
                         * * * * *


Do you wish to 1 enter data from keyboard
               2 read a file
               3 generate random nos. in col. 1
               4 create stepped or const. data through edit sub-system
= 1

How many rows do you need in your workspace?
 Depress "RETURN" for default of  200 rows. (? for HELP)
=

There are 20 columns available

Enter data one row at a time. Numbers should be separated by one or more blanks.
Use "b" to back up a row for correction.
To terminate entry, depress "RETURN".
max number of observations (rows) = 200
max number of variables (columns) =  20
Row   1 = 50 69
Row   2 = 66 85
Row   3 = 73 88
Row   4 = 84 70
Row   5 = 57 84
Row   6 = 83 78
Row   7 = 76 90
Row   8 = 95 97
Row   9 = 73 79
Row  10 = 78 95
Row  11 = 48 67
Row  12 = 53 60
Row  13 = 54 79
Row  14 = 79 79
Row  15 = 88 86
Row  16 = 90 98
Row  17 = 60 56
Row  18 = 89 87
Row  19 = 83 91
Row  20 = 81 86
Row  21 =

  20 rows  2 columns have been read

Do you wish to label columns? y or n
= y

Enter  2 names, one per line, up to 8 characters.
= TEST1
= TEST2

 2 names entered

Do you wish to print your data? y or n
= y

Enter 1 to print all
      2 for a row
      3 for a column
```

```
-

= 1


    TEST1     TEST2
    50.00     69.00
    66.00     85.00
    73.00     88.00
    84.00     70.00
    57.00     84.00
    83.00     78.00
    76.00     90.00
    95.00     97.00
    73.00     79.00
    78.00     95.00
    48.00     67.00
    53.00     60.00
    54.00     79.00
    79.00     79.00
    88.00     76.00
    90.00     98.00
    60.00     56.00
    89.00     87.00
    83.00     91.00
    81.00     86.00

Do you want the catalogue of commands? y or n
= y

Type 4-letter command. ( )indicates program  (eg MER1 or MER2)

       inst   for general instructions on running MERLIN (1)(2)
       cata   to list the catalogue of commands (1)(2)
       con1   for regular shewart control chart (n=1)  (2)
       con2   for special shewart control chart (2)
       data   to enter data (1)all formats (2)MERLIN & .prn only
       date   for date and time (1)(2)
       disc   to read command stack from a file (1)(2)
       edit   for modification and printing of data (1)
       eras   to erase files (1)(2)
       fcst   for forecasting routines (2)
       file   to direct output to a file (1)(2)
       filt   for selection of specific rows and/or columns (1)
       freq   for frequency tables or histograms
              & fitting of probability distributions (1)(2)
       hed1   for titles with auto. centering (1)(2)
       hed2   for titles printed as entered (1)(2)
Pause.
Please press <return> to continue.
       inle   to change input line length (1)(2)
       name   to label columns (1)(2)
       norm   for plotting on normal probability paper (2)
       outl   to change output line length & printer codes (1)(2)
       page   to set printer at top of page (1)(2)
       plo1   for simple plot (2)
       plo2   for plot with more options (2)
       prin   to list data (for more options use "edit") (1)(2)
       reg1   for regular regression (2)
       reg2   for regression with more options (2)
       reme   to remember command files (1)(2)
```

```
      save    to save data (1)(2)
      sort    to sort data (1)
      stat    for mean, sd., std.error, max. & min, range;
              t or f tests; analysis of variance;
              & sampling probabilities (1)
      stop    to stop execution (1)(2)
      term    to redirect output to the keyboard (1)(2)
      time    for date and time (1)(2)


 Enter command
 = edit

 Enter edit command (? for instructions). Depress "RETURN" to exit
 = ?

 Commands: print or list(p); add(a); delete(d); insert(i); replace(r).
     Also: print with entered format(pf); print with dates as row labels(pd);
           suppress decimal in print(ps)

 Operands: row; column(col); element(el); column heading(nam); all(*).

 Row and column functions:  square root(sqrt); exponential(exp);
                           log to the base 10(log); log to the base e(ln);
                           totalling(sum); truncati .i(int); absolute(abs);
                           accumulation(cum).

 Column functions: random numbers[rnd];
                   col shift[sfty:x(coln)], where y=u for up, y=d for down
                           and x=no of rows to be shifted eg sftu:6(col3)
                   stepped values[stp(sv step)], where sv is starting value.
                            eg stp(25 5).

 Use single character d to display array & workspace dimensions.
 Pause.
 Please press <return> to continue.

 Rules: To terminate edit command depress "RETURN"

        Multiple row or col references valid for print and delete only.

        References to indiv. rows or cols must be separated by a blank eg col3 9

        Blocks are indicated by a dash eg col1-5

        For print, row and col combinations are separated by a colon(:)
             eg p col1-3:row1-10

        Element: row no., col no., in parentheses eg el(2 3)
                 Only 1 element per instruction.

 Pause.
 Please press <return> to continue.

 Examples:  p *
            p col3 1 2
            p row2-6
            p el(2 5)
            p row1-3:col7 3
            pds *
```

```
            a col2          (entry of a col. of nos will be requested)
            a col4 = sqrt(col1) - col2**(col3+1) - 27.3
            a row26 = sum(row*)
            a col5 = cum(col4)

            r nam* or r nam1       (entry of labels will be requested)
            r el(12 3)   but not   r el(12 3)=6
            r col6 = -col2
            r col6 = rnd
            r col7 = stp(25.5 -.5)

            i col10 = sftu:12(col4)
            i row25 = sum(row1-24)/100

            d row1-5
            d col1-3 5 8   but not   d col*
Enter edit command
= r el(15 1)

Enter   1 numbers
= 55

el.  replaced

Enter edit command
=

Enter command
= save

Enter filename (? for HELP)
= ?
c:*        file       size         date       time

    MER1N.EXE    339673    10-22-90     5:55
   WMERBX.CMD         0    12-29-84    14:17
    TRACE.TCE         0    12-29-84    14:17
   MERLIN.DRV         8    11-05-90    23:13


Use extension .dat for free format
              .fmt for entered (fixed) format
              .bin for binary format
              .mer for MERLIN format (default)
              .prn for 1-2-3 format

Enter filename (? for HELP)
= try

Data on disk as permanent file, c:TRY.MER

Enter command
= s

Backup written.


C:\MERLIN>
```

MERLIN comprehensive HELP system

Developed by P.F.Wade

This HELP system has been designed to augment the context sensitive HELP messages available within MERLIN.

Note to users: Unlike a spreadsheet program, MERLIN distinguishes between:

.data - the data being analysed and

.output - the results generated by the analysis

The documentation which follows reflects these concepts.

* * * *

- HELP conventions -

To exit HELP

(1) alt+l.shift (restart where terminate)

(2) esc restart at beginning

E means press E for more details on ....

pgup for previous screen

home for first screen

Press space bar to move to INDEX

Sample Screens of Merlin's Hypertext Help Index Annex
Main Menu

```
                          - INDEX -
       Press  A  for Commands, list of
              B      Entering data (DATA)
              C      Exporting data (SAVE)
              D      Express mode
              E      Fixed format specifications
              F      Histograms & bar charts (FREQ)
              G      Importing data (DATA)
              H      MERLIN files & file commands
              I      Overview of MERLIN structure & conventions
              J      Plotting data (incl. control charts)
              K      Printing or displaying workspace data (EDIT, PRIN)
              L      Printing output (FILE, TERM, HED1, HED2, OUTL, PAGE)
              M      Sampling distributions
              N      Saving commands & using command files (REME, DISC)
              O      Saving data (SAVE)
              P      Saving output (same as L)
              Q      Selecting data subsets (FILT) + selective processing
              R      Sorting (SORT)
              S      Statistical tests & techniques
              T      Transforming & modifying data (EDIT)
 X ► examples    Y ► trouble    Z ► tricks    U ► installation    esc  to exit
```

Sample Screens of Merlin's Hypertext Help Index Annex

Data Entry Example

```
Example #1: Data entry from keyboard                                  Screen #1

Enter command
= data

Do you wish to 1 enter data from keyboard
               2 read a file
               3 generate random nos. in col. 1
               4 create stepped or const. data through edit sub-system
= 1

How many rows do you need in your workspace?
 Press "RETURN" for default of 200 rows. (? for HELP)
= <CR>

There are 20 columns available

The data, separated by blanks are read in free format (? for HELP).
Press "RETURN" to exit.
= 2.01 73 125
= 1.22 54 195
-------------------------------------------------------------------------------
space bar to continue                                              esc to exit
```

Sample Hypertext-based Help Screens on Extended Merlin.

```
                        -MERLIN overview-                          Screen #1

              MERLIN is a data analysis package for:
                        data extraction
                        regression
                        plotting
                        sorting
                        forecasting
                        histograms
                        analysis of variance
                        statistical tests
                        control charts
                        data modification
                        random number generation
                        .... etc

Entered data (numbers only) are stored as a 2-dimensional array in a memory
workspace with a maximum of 4000 elements. The user is asked to set the di-
mensions of this workspace.  Column labels are entered separately.

 space bar to continue                                          esc  to exit
```

```
                      - MERLIN overview cont'd-                    Screen #2

In response to ENTER COMMAND, a 4-letter command (eg. SORT) is entered.
A series of menus and questions then identifies the required options.
Commands can be entered in 3 modes:
              .conversational mode - user responds interactively
              .express mode - user enters responses in a string
              .command file - responses are read from a file

Commands are separated into two programs, MER1 and MER2, but the transfer
from one to another is made automatically.

Data input
Enter data values when requested, or use the command, DATA. The numbers are
entered a row at a time and rows must be of equal length.  Data can be
entered from the keyboard or a file.

Saving data
Use the command, SAVE, to save data contained in the workspace.
(Data are saved automatically every 15 minutes while you are working and
 put into a file named BACKUP.MER.)

 space bar to continue                                          esc  to exit
```

```
                         -MERLIN overview cont'd-                    Screen #3

Printing data, tables graphs etc.
Use CTR+P to put your printer into lock-step with the screen or save output
in a file and print it later. See next section.

Saving output in a file
Output such as data listings, tables and graphs is normally directed to the
screen.  Use the command, FILE, to save it as a printable file.
Note: (1) The output will not be displayed on the screen in this case
          but verification messages will appear.
      (2) Output can be re-directed back to the screen using command, TERM.

                      ** Special Conventions **
     At any time: "r" will return you to ENTER COMMAND
                  "b" will back up to the previous response
                  "s" will terminate the run

     Entry of commands may be in upper or lower case

     For HELP: Where indicated, type "?" for explanation of prompts.
-----------------------------------------------------------------------------
 pgup  for previous screen                                       esc  to exit
```

```
List of MERLIN commands                                          Screen #1

          inst   for overview of MERLIN command structure

          cata   to give catalogue of commands (this screen)
          con1   for Shewhart & CUSUM charts (n=1)
          con2   for Shewhart & CUSUM charts (n>1)
          data   to enter data from keyboard or a file(ie. import)
          date   for current date & time from computer clock
          disc   to read a command file (like a macro)
          edit   for modification and listing of data
          eras   to erase files from DOS
          fcst   for forecasting routines
          file   to direct output to a file instead of the screen
          filt   to select a subset of rows from the data array
          freq   for frequency tables & histograms plus fitting
                 of probability distributions
          hed1   for titles on your output - automatic centering
          hed2   for titles - displayed as entered

                                        ......continued on next screen

  space bar to continue                                      esc  to exit
```

```
MERLIN commands continued                                        Screen #2

     inle   to change input line length
     name   to add labels for data columns
     norm   for plotting on normal probability paper
     outl   to change output line length & send printer control codes
     page   to set printer at new page
     plo1   for plots with limited options
     plo2   for plots with multiple options
     prin   to list data (for more flexible formatting use EDIT)
     reg1   for regression analysis with limited options (incl. plot)
     reg2   for regression analysis with multiple options (incl. plot)
     reme   go to/from remember mode to create command files (toggle)
     save   to save data (a variety of formats can be used to export)
     sort   to sort data based on one or more data columns
     stat   for mean/std.dev/max/min/range of a data column
            t & F tests/1-way & 2-way analysis of variance
            sampling probabilities (poisson, binomial, hypergeom.)
     stop   to terminate MERLIN session (at any entry point use 's')
     term   to redirect output to the screen (after using FILE)

  pgup  for previous screen                                  esc  to exit
```

```
HELP for EDIT commands                                              Screen #1

 The EDIT function does not use menus but is command driven. Valid
 commands are: [p]print or list; [a]add; [d]delete; [i]insert; [r]replace

 Operands: row[row]; column[col]; element[el]; column label[nam]; all[*]

 Row and column functions: square root[sqrt]; exponential[exp];
                           log to base 10[log]; log to base e[ln];
                           totalling[sum]); truncation[int]; absolut[abs];
                           accumulation[cum];  eg col5=cum(col1)

 Column functions: random numbers[rdn];
                   col shift[sfty:x(coln)] where y=u for up, y=d for down
                       and x=no of rows to be shifted  eg sftu:6(col3)
                   stepped values[stp(sv step)] where sv is starting value
                       and step is step size   eg stp(20.1 .5)

 Use single character "d" to display data and workspace dimensions.

                   -- See Screen #4 for examples --
------------------------------------------------------------------------
 space bar to continue                                        esc  to exit
```

```
HELP for EDIT commands continued                                    Screen #3

 General Rules
   .To terminate EDIT function, press "RETURN".

   .Blocks are indicated by a dash  eg col1-5

   .Multiple row or col operands may be used for print & delete only.
                                                          eg d col1-3
   .References to multiple operands must be separated by a blank or
     use block convention.  eg p col1 7 9-11 to list cols 1,7,9,10,11

   .For print, row amd column combinations are separated by a colon(:)
                                                 eg p col1-3:row1-10
   .A specific cell or element in the data array is addressed as el(r c)
     where r is the row # and c is the col#  eg el(2 3) for row2 col3
     An instruction is limited to 1 element

   .To format numbers when listing use: [pf] to list with entered format;
     [ps] to list with decimals suppressed; [pd] for dates as row labels
                  -- See Screen #4 for examples --
------------------------------------------------------------------------
 space bar for next screen                                    esc  to exit
```

```
HELP for EDIT commands          - EXAMPLES -                        Screen #4
  Enter edit command
  =  p *                     display complete data array
     p col3 1 2              display columns 3,1,2
     p el(2 5)               display cell at row2, col5
     p row1-3 5:col7 3       display subset, 4 rows x 2 cols
     pd * (or pds *)         display array with date row labels
     a row26=sum(row*)/25    add row26 = average of all 25 rows
     a col5=cum(col4)        add col5 = col 4 accumulated

  =  r nam*                  replace all col labels (then enter n labels)
     r el(12 5)              replace cell at row12 col5 (then enter 1 #)
     r col6=-col2            replace col6 by negative of col 2
     r col6=rnd              replace col6 by rnd nos (then give distn)
     r col7 = stp(24 -1)     replace col7 by 24 23 22 21 .........

  =  i col1=sftu:12(col4)    insert col1 = new col 4 shifted up 12 rows
     d row1-3 5 8            delete rows 1,2,3,5,8
-------------------------------------------------------------------------------
pgup  for previous screen                                         esc  to exit
```

Sample Evaluation of Package using Evaluation Form

Factors important for learning in general, and for learning of software can be identified as those which:

a. support **Speed**, that is reduce performance time;
b. alleviate **Memory Load**, that is promote retention or not require it;
c. minimize **Mental Effort**, that is reduce mental strain;
d. provide **Psychological Comfort**, that is reduce anxiety, promote peace of mind.

After you have completed the task which is attached using each of the statistical packages, Merlin, then Minitab, please rate the package and each feature listed below according to the extent to which it supports these learning factors - **Speed, Memory, Effort, Comfort**. Definitions and descriptions of the features and learning factors are attached as a reference.

1. Indicate the package being evaluated. Next, an overall rating of the package should first be made by assigning a number between 0 - very poor and 10 - excellent.

2. Next, assign a weighting out of 100% to each of the learning factors to indicate the extent to which you perceived they were supported in the package overall. Equal weighting is permitted but weights must add to 100%. Place weights in the brackets below the learning factors.

3. Third, a rating is to be done of each of the design and assistance features to indicate the extent to which you perceived Speed, Memory, Effort, Comfort, were supported in the package as a consequence of the presence or absence of the particular feature by assigning a number between 0 - no support, and 10 - great support.

Package: Minitab **Overall Rating:** 7

Design Features:	Speed (50)	Memory (10)	Effort (30)	Comfort (10)
1. Interface Command Structure - menus, commands, etc.	6	2	4	7
2. Depth of Structure	8	2	3	4
3. Logic of Structure	8	7	6	5
4. Consistency	9	9	9	9
5. Screen Design - cosmetic/aesthetics - meaningful grouping - language, wording	NA	NA	NA	NA
6. Flexible	5	2	3	3
7. Error Trapping/Recovery	5	2	2	2
8. System Response Time	10	9	9	9
9. Data Input - full-screen, line editor	8	5	5	5
10. Macros	NA	NA	NA	NA
11. Autosave/Backup	DK	DK	DK	DK
12. Auto-adjustment	DK	DK	DK	DK
13. Auto-formatting	DK	DK	DK	DK

Don't know

Assistance Features:	Speed (50)	Memory (10)	Effort (30)	Comfort (10)
14. Manuals	NA	NA	NA	NA
15. System Messages	8	5	5	2
16. Tutorials	NA	NA	NA	NA
17. Prompts	NA	NA	NA	NA
18. Keyboard Templates	NA	NA	NA	NA
19. Defaults	OK	OK	OK	OK
20. Examples	10	10	10	10
21. Index of commands/operations	10	2	6	3
22. Glossary	NA	2	2	4
23. Unsolicited Help	NA	NA	NA	NA
24. Cautions	OK	OK	OK	OK
25. Checklists & Memory Jogs	OK	OK	OK	OK
26. Navigational Aids	NA	NA	NA	NA
27. Instructive Feedback	9	7	6	6
28. Context Help	10	8	9	10
29. Expertise Accommodation	OK	OK	OK	OK
30. Restricted Options	OK	OK	OK	OK
31. Subject Matter Aid	OK	OK	OK	OK
32. Conceptual Models	5	2	2	2

Appendix F

User Perception Questionnaire		
Overall liking		
How did you like using this package?		
( ) Very much ( ) Above average ( ) Average ( ) Below average ( ) Not at all		
Design		
Command Structure		
Speed. Did you find the menus/commands fast to use?	*Yes*	*No*
Memory. Was it easy to remember which menus/commands to use?	*Yes*	*No*
Effort. Were the menus/commands difficult to use?	*Yes*	*No*
Comfort. Did you feel at ease using the menus/commands?	*Yes*	*No*
Depth of Structure		
Speed. Did you find you were able to move through the menus/commands quickly?	*Yes*	*No*
Memory. Was it easy to remember the sequence of menus/commands to accomplish something?	*Yes*	*No*
Effort. Was it difficult to move through a sequence of menus?	*Yes*	*No*
Comfort. Did you feel at ease going through a sequence of commands?	*Yes*	*No*
Logic of Structure		
Did the overall structure of the package make sense to you?	*Yes*	*No*
Speed. Did you feel you were able to accomplish your task quickly because of this?	*Yes*	*No*
Memory. Did you feel the package required you to remember a lot?	*Yes*	*No*
Effort. Did you find the package overall to be difficult to use?	*Yes*	*No*
Comfort. Did you feel comfortable using the package?	*Yes*	*No*
Consistency		
Did you find menu/commands were used consistently throughout?	*Yes*	*No*
Speed. Did this speed up your use of the package?	*Yes*	*No*
Memory. Did this impose more of a memory strain?	*Yes*	*No*
Effort. Did this require more effort in order to perform your task?	*Yes*	*No*
Comfort. Did this make you feel less at ease performing your task?	*Yes*	*No*

Screen design	
Did you find the screens aesthetically pleasing?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Did you feel the screens were easy to read?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Were the instructions easy to follow?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Was the wording clear?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Was the wording offensive?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*

Flexibility	
Did you find there was more than one way to do things?	*Yes No*
Speed. Did this allow you to accomplish your task faster?	*Yes No*
Memory. Did this tax your memory?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Error trapping/ Recovery	
Were you prompted when you entered something incorrectly?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Were your errors corrected for you?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Was it easy for you to correct your errors?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
System Response	
Did you have to wait long after you entered something to get a response from the computer?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*

Data Input	
Speed. *Did you find it fast to enter data?*	*Yes No*
Memory. *Was it easy to remember how to enter data?*	*Yes No*
Effort. *Was it burdensome to enter data?*	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Autosave/backup	
Did the package automatically save your inputs intermittently?	*Yes No*
Speed. *Did this help you to accomplish your task faster?*	*Yes No*
Memory. *Did this impose more pressure to remember things?*	*Yes No*
Effort. *Did this make accomplishing your task harder?*	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Assistance	
System Messages	
Were the messages given by the system clear?	*Yes No*
Speed. *Did this help you to accomplish your task faster?*	*Yes No*
Memory. *Did this impose more pressure to remember things?*	*Yes No*
Effort. *Did this make accomplishing your task harder?*	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Were the messages offensive?	*Yes No*
Speed. *Did this help you to accomplish your task faster?*	*Yes No*
Memory. *Did this impose more pressure to remember things?*	*Yes No*
Effort. *Did this make accomplishing your task harder?*	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*

Prompts		
Were you warned when you made an error?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
Were you told how to correct it?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
Were you told why it occurred?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
Defaults		
Were expected responses automatically provided which you could then modify?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
Examples		
Were examples given?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*

Index	
Was an index of commands and operations available?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Cautions	
Were you cautioned about making certain actions, e.g. saving a file under an already existing filename?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Navigational aids	
Was it easy to move around the package?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Was it easy to know where you were in the package at most times?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*
Were you able to move backwards and forwards easily in the package?	*Yes No*
Speed. Did this help you to accomplish your task faster?	*Yes No*
Memory. Did this impose more pressure to remember things?	*Yes No*
Effort. Did this make accomplishing your task harder?	*Yes No*
Comfort. Did this make you feel more at ease performing your task?	*Yes No*

<u>*Context help*</u>		
Were you able to get help related to where you made an error?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
<u>*Expertise accommodation*</u>		
Would you say the package is geared for novice users?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
Does it have the same commands and procedures for novice and expert users?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*
<u>*Subject matter expertise*</u>		
Were you given assistance with learning statistical concepts needed for the task?	*Yes*	*No*
Speed. Did this help you to accomplish your task faster?	*Yes*	*No*
Memory. Did this impose more pressure to remember things?	*Yes*	*No*
Effort. Did this make accomplishing your task harder?	*Yes*	*No*
Comfort. Did this make you feel more at ease performing your task?	*Yes*	*No*

Appendix G

Data Entry Task used in Study 2

SIGN ON PROCEDURES

Enter the group name, account number and password you have been assigned.

Group Name ________________

Account Number ________________

Password ________________

Do not change your password.

The following menu will appear:

F1	DOS AND BASIC	F6	MATHEMATICAL PACKAGES
F2	WATCOM TEXT EDITOR	F7	SIMULATION SOFTWARE
F3	FORMAT A DISKETTE	F8	SPREADSHEETS
F4	DATABASE SOFTWARE	F9	STATISTICAL SOFTWARE
F5	PROGRAMMING LANGUAGES	F10	LOGOFF

PLEASE SELECT AN APPLICATION

Press the F9 key to select STATISTICS

The following menu will appear:

F1	MERLIN	F6	RETURN TO PREVIOUS MENU
F2			
F3			
F4	MINITAB		

Press either F1 or F4 depending on the name at the top of your task sheet.

MERLIN, /MINITAB, TASK

Below is information on test scores collected for thirty one students. Your task is to enter this data into a stanstical software package which will be used eventually to predict student performance on the particular course. You will find assistance for doing so by typing ? . HELP HELP You are to enter the data as well as the headings for each column. Verify your entries and correct any that are incorrect. Save your work. This done, change Student 15 's Test1 score to 55. Exit the program.

Student	Test1	Test2
1	50	69
2	66	85
3	73	88
4	84	70
5	57	84
6	83	78
7	76	90
8	95	97
9	73	79
10	78	95
11	48	67
12	53	60
13	54	79
14	79	79
15	88	76
16	90	98
17	60	56
18	89	87
19	83	91
20	81	86

Appendix H

Sample Keystroke Tracing in Merlin,

```
 1/15/1988
16:38:24            MER1
16:38:33       9.   N
16:38:42       9.   1
16:39: 1      19.   31
16:39:41      40.   50 69
16:39:47       6.   66 85
16:40:19      32.   73 88
16:40:26       7.   84 70
16:40:32       6.   57 84
16:40:40       8.   83 78
16:40:45       5.   76 90
16:40:52       7.   95 97
16:40:58       6.   73 79
16:41: 4       6.   78 95
16:41:19      15.   48 67
16:41:29      10.   53 60
16:41:35       6.   54 79
16:41:40       5.   79 79
16:41:46       6.   88 76
16:41:51       5.   90 98
16:41:57       6.   60 56
16:42: 2       5.   89 87
16:42: 8       6.   83 91
16:42:14       6.   81 86
16:42:23       9.   57 69
16:42:34      11.   71 75
16:42:39       5.   86 98
16:42:48       9.   82 70
16:43: 0      12.   95 91
16:43: 9       9.   42 48
16:43:13       4.   75 52
16:43:18       5.   54 44
16:43:27       9.   54 51
16:43:33       6.   65 73
16:43:40       7.   61 52
16:43:48       8.
16:44:23      35.   Y
16:44:34      11.      first
16:44:38       4.     second
16:44:52      14.   Y
16:44:57       5.   1
16:45:14      17.   Y
16:45:39      25.   EDIT
16:45:39       0.
16:45:47       8.   ?
16:48:15     148.   ?
16:48:56      41.   R ROW15
16:49: 2       6.   76 88
                                CORRECTION
16:49:15      13.
16:49:19       4.
16:49:20       1.
16:49:32      12.   ?
16:49:32       0.
16:50:23      51.   REG1
16:50:23       0.

Total module time =     12.0 minutes
 1/15/1988
```

Appendix I

Package Functions Annex to Evaluation Form

An additional set of features, related to functionality, would have to be attached to the instrument. The features included would have to be derived from an assessment of the task requirement of the specific package type. For instance, the functional requirements of a wordprocessor are different from those of a spreadsheet from those of a statistical package. The degree to which these functions exist in a package and the ease with which they are achieved will impact the choices made among the available packages.

The requirements of a statistical package can be considered within the classical system cycle of input, process, output, and feedback. A user will need either to be able to create a file of data or read it in from an existent file. Once the data is in the program, some transformations and editing of data may be required. This implies the facility to create new columns, new rows, delete columns and rows, modify data elements, use subsets of variables and observations. Next, a range of analyses are used to interpret and analyse the data. These include the calculation of means and standard deviations, chi-square tests, F and t-tests, graphs, anovas, regression analyses, forecasts, et cetera. Output of the data and of these various analyses and graphs will be required either on paper, screen and/or to a file. The evaluation instrument, therefore, would have the following features attached:

Suggested List of Statistical Functions to be evaluated based on their support for the identified learning dimensions

	Speed	Memory	Effort	Comfort
c. Functionality				
Data Input				
-Create file				
-Read file				
Data Editing				
-Changing data elements				
-Adding rows/columns				
-Deleting rows/columns				
-Extracting subsets				
Data Analyses				
-Descriptive statistics - mean,mode,median,std.dev.,number				
-Tests - chi-square, F, t				
-Regression - simple, multiple				
-Forecasts - time series, moving averages				
-Plotting - graphs, charts, tables				
-Multivariate				
-Etc.				
Data Input/Analyses Output				
-To a file				
-To the screen				
-To a printer				

Since the task in this experiment does not entail using any of these statistical functions, evaluation of these features will be excluded from the instrument. As explained previously however, they are an essential part of the instrument, and would vary depending on the package type being evaluated.

Appendix J

Table A1

Findings Summary of Research into Factors Affecting Performance with Software

DESIGN CHARACTERISTICS () indicates # of related studies		
ASSISTANCE	**FEATURES**	**INTERACTION**
explicit instructions and training on reduced package features aid initial learning but can hamper transfer learning implying need for both general, conceptual explanation as well as specific, step-by-step instruction (12); content of instruction is more important than delivery method though generally printed manuals out-perform online manuals, except in Hiltz & Kerr,1986 study in which users self-reports of learning methods indicates that offline documentation is the least adopted mode of learning (6); users perform better with manuals but exhibit a tendency to prefer & use self-discovery learning and to learn & use more advanced features as a consequence (3); there is a tradeoff between time and mastery learning, hence users often forgo learning advanced features (2); relevance to task is important for transfer learning, as is feedback (5)	there is an optimum level of response & display rates out-side of which users' perform-ance diminishes (6); meaning-ful grouping of information improves performance, espec-ially if it is in a familiar order (2); abbreviated, even arbitrary command names do not impede learning substan-tially but users need to understand the underlying or associated meanings (6); word meanings differ across users (2); performance & preference are enhanced with natural language(3)	users tend to consistently prefer one mode of interaction over another (1); studies are inconclusive about the advisability of menu or command for users, especially novices (10); voice interaction seems to be preferred, though results on performance are conflicting (3)
USER CHARACTERISTICS		
KNOWLEDGE/EXPERIENCE	**PSYCHOSTRUCTURE**	**DEMOGRAPHICS**
results on novice/expert performance varied; improvements generally beneficial to both groups (1); no consistent results on use & value of help, though experts are better able to know what to seek help on & to understand the help received (6); experts are better able to monitor themselves; experts exhibit higher level understanding & know more complex procedures than novices (3); prior package experience seems to produce more negative transfer than positive (2); task experience has some impact on performance (5)	there is some evidence that matching learning style to type of assistance and interface enhances performance (3), and that intellectual capacity impacts performance (4)	age has more influence on motivation than on performance (1)
PERFORMANCE		
PREDICTION	**IMPEDIMENTS**	
Card, Newell & Moran,1983 - learning time is the sum of the number of keystroke * the time per keystroke + the mental effort, etc. Polson & Kieras,1985 - learning time is a function of the number of new production rules which have to be learnt	results of transfer are inconclusive (8); timing of training important for retent-ion & transfer (1); error patterns exist among users (1); users need hooks such as models (3)	

Table A2

Author Summary of Research into Factors Affecting Performance with Software

DESIGN CHARACTERISTICS		
ASSISTANCE	**FEATURES**	**INTERACTION**
Documentation	**Screen Design**	**Mode**
Black,et.al.,1989; Carroll & Aaronson,1988; Catrambone & Carroll,1987; Elkerton & Williges, 1987; Schlager & Ogden,1986; Holt,et.al.,1986; Charney & Reder,1986; Carroll,1985; Carroll & Kay,1985; Borenstein,1985; Carroll & Mack, 1984; Cordes,1984; Bayman & Mayer,1984; Watley & Mulford,1983; Magers,1983; Roemer & Chapanis,1982; Shneiderman,1982; Cohill & Williges,1982; Dunsmore,1980; Relles,1979	Burns,et.al., 1986; Badre, 1982	Ogden,et.al.,(in press),Mack, Lewis & Carroll, 1987; Ziegler,et.al.,1986; Lee, MacGregory, et.al.,1986, Ogozalek & Praag,1986; Whiteside, et.al.,1985; Maskery,1985; Cordes,1984; Perlman,1984; Emmons,1984; Ambardar,1984; Shutoh,1984; Murray,et.al.,1983, Hauptmann & Green,1983; Smith,1983; Gilfoil,1982, Tombaugh & McEwen,1982; Dunsmore,1981; Ledgard,et.al., 1980; Walther & O'Neil,1974
Training	**Display & Response Times**	**Naming**
Olfman & Bostrom,1988; Nakatani,et.al., 1987; Hiltz & Kerr,1986; Czaja,1986; Goodwin & Sanati,1986;	Dannenbring, 1983; Smith,1983; Bevan,1981; Long,1976	Grudin & Barnard,1985; Landauer,et.al., 1983,Dumais & Landauer,1983; Barnard, et.al., 1982; Black & Moran,1982; Ehrenreich & Poreu, 1982, Moses, Mendez, Ehrenreich,1980
Feedback		**Package**
Egan, et.al.,1987		Napier,et. al.,1989, Foss & DeRidder;
USER CHARACTERISTICS		
KNOWLEDGE/EXPERIENCE	**PSYCHOSTRUCTURE**	**DEMOGRAPHICS**
Computer/Dos	**Learning Styles**	**Age**
Doane,1986; Holt,,et.al.,1986; Burns,1986; Borenstein, 1985; Emmons,1984; Elkerton & Williges,1984; Gilfoil,1982; Relles,1979	Bostrom, Olfman & Sein,1990; Kern & Matta,1988; Ambardar,1984	Ogozalek & Praag,1986
Package	**Meta-cognitive skills**	
Karat,et.al.,1986; Hauptmann & Green, 1983; Ledgard,et.al., 1980; Walther & O'Neil; 1974	Evans & Simkin,1989 Gomez, Egan & Bowers,1986; Egan & Gomez, 1985, Roemer & Chapanis,1982	
Task		
Roth, Bennett & Woods,1987; Gugerty & Olson, 1986; Maskery,1985; Shutoh,1984; Roberts & Moran,1983; Dannenbring,1983; Smith, 1983; Murray,et.al, 1983; Bevan,1981; Long,1976		
PERFORMANCE		
PREDICTION	**IMPEDIMENTS**	
Card, Newell & Moran,1983; Polson & Kieras,	Elkerton & Williges,1987; Egan,et.al.,1987; Schlager & Ogden,1986; Ziegler,et.al., 1986; Czaja,1986; Charney & Reder,1986; Carroll & Kay,1985, Carroll,1985, Foss & DeRidder,	

Table A3a.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS FEATURES	USER CHARACTERISTICS () # of subjects	FINDINGS
Burns, et.al., 1986	reformatted/ current display	novice/expert	speed & accuracy improved for novices; experts had fewer errors, no change in times
Badre, 1982 Exp.I	information presentation in meaningful chunks in familiar order/non-meaningful chunks/ order unfamiliar	experienced tactical decision makers (36)	greater recall with meaningful chunks & order; non-meaningful gave greater accuracy than meaningful, unfamiliar order
Exp.II	display placement	experienced tactical decision makers (35)	no recall difference with placement
Grudin & Barnard, 1985*	full command abbreviations/ abbreviations/ create own abbreviations	novice	full command with abbreviation outperformed other groups
Landauer, et.al., 1983* Exp.I	natural language to explain text-editing task to "other"	computer naive (22/26)	current names on system not chosen; little agreement in naming
Exp.II	old command/ new/random	secretarial student/ high school students with typing skills (65/56)	performance time same; old set somewhat preferred
Dumais & Landauer, 1983	command naming	secretarial & high school students with typing but no computer experience (48)	subjects did not agree on naming
Barnard, et.al. 1982*	specific/general commands		recall & recognition higher for specific commands
Black & Moran, 1982*	8 command sets	(84)	infrequent,discriminating commands result in faster learning and superior recall; general words perform worst; nonsense words perform well

Table A3a.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS FEATURES	USER CHARACTERISTICS () # of subjects	FINDINGS
Ehrenreich & Porsu, 1982	truncate vs. contraction/ fixed vs. variable length/ non-systematic/ endings	military enlisted personnel (144)	for encoding commands, truncation easier than contraction if rules known, length & non-systematic had no impact, performance lower if endings added, rules superior to no rules, for decoding, no difference for truncation or contraction if rules known; if rules unknown truncation superior; variable length decoded more often than fixed length, decoding easier if endings incorporated and rules known
Ledgard, et.al., 1980*	symbol/keyword commands	inexperienced/ familiar/experienced text editors (8)	keyword improved performance but less so for experienced users
Moses, Mendez, Ehrenreich, 1980	truncate- variable/ contraction- variable/ no systematic pattern		for encoding, no significant difference between no pattern & truncation, both superior to contraction; no significant difference for decoding
Dannenbring, 1983*	response times	novice/expert programmers	no effect of response time on performance or satisfaction of either group, though scores between groups differ
Smith, 1983*	response times	skilled vs. novice CAD designers	reduced response time increased interaction rates for both groups
Bevan, 1981*	display rates	low & high reading abilities (24)	lesson time decreased as display rates increased but errors increased; high rate disliked most by both groups, with low ability preferring lowest rate
Long, 1976*	response times	skilled/unskilled typists	longer response times resulted in reduced performance measures over time for both

*In Shneiderman (1987)
**In Houghton, Jr.(1984)

Table A3a.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS INTERAC ON	USER CHARACTERISTICS () # of subjects	FINDINGS
Ogden, et.al.,(in press)	menu/command in same system	(15)	menu was selected if access was immediate but decreased as levels increased; only one third of subjects consistently used menus; one method consistently preferred over the other; subjects with least experience relied on command, most experience on menus
Mack, Lewis & Carroll, 1987	menu/command	office temporaries (10)	learning is difficult; lack basic knowledge; make ad hoc interpretations; generalize from what they know; have trouble following directions; problems interact; interface features may not be obvious; help does not always help (exploratory)
Ziegler, et.al., 1986	direct manipula-tion of text/ graphic editing	novice in computer systems (12)	learning curve same on both; greater transfer from text to graphic than reverse
Lee, MacGreg-ory, et.al., 1986	menu/menu + keyword command	naive	menu + keyword outperformed menu only; keywords preferred as way to bypass hierarchy
Ogozalek & Praag, 1986	voice/keyboard	elderly/younger (12/12)	no performance difference on age or input device, but voice preferred; elders more motivated
Whiteside, et.al., 1985	7 different DBMS systems with menu/ icon/command		command best for all, novices great difficulty with menus
Maskery, 1985	system leads-user forced choice/system leads-user free choice/user leads-user free choice adaptive interface for statistical tool	naive users with basic Statistics knowledge (18)	weekly usage promoted better learning than daily; break of 5-6 weeks led to initial decrement in learning followed by rapid improve-ment; change between dialogue styles must be consistent & predictable with previous system; subjects confused by adaptive design
Perlman, 1984	mnemonic menu/ sequential letter/ numeric menu		think time lowest with mnemonics, highest with sequential, numerical in-between
Cordes, 1984	form-fill in/ improved form/ commands	some experience with similar system (16)	relied on online help; commands outperformed forms, original forms poorer than either commands or improved forms

Table A3a.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS INTERACTION	USER CHARACTERISTICS () # of subjects	FINDINGS
Shutoh, 1984**	voice/menu	novice/advanced CAD system users	voice faster than menu
Murray, et.al., 1983*	voice/cursor	novice/experienced typist	both preferred keyboard, except initially novices preferred voice; no difference in performance
Hauptmann & Green, 1983	menu/command/natur al language	novice to experienced programmers with no package experience (12/12/12)	no difference in time or accuracy
Gilfoil, 1982*	menu/command	novice to experienced over time	use of help facility decreased as experience increased, to point switch to command mode when help again sought till mode mastered
Tombaugh & McEwen, 1982*	alphabetic menu/ tree search	(30)	no difference in database search performance
Dunsmore, 1981*	item-return/ immediate response/ high-light-return	high school students (36)	slightly faster with immediate response but error rate higher; highlight preferred
Walther & O'Neil, 1974	flexible/inflex-ible text editor	novice/expert	novices performance quicker with inflexible system, experts with flexible
Napier, et.al., 1989	HAL/Lotus1-2-3	inexperienced with computers & spread-sheets (22/20) Lotus/HAL	user acceptance & performance higher with HAL interface
Foss & DeRidder,	EMAC/SOS/ BASIC vs. DEC K52	college students with typing but no com-puter text editing experience (36)	positive transfer from other editors to DEC editor but not from BASIC, transfer was not attributable to the number of production rules or similarity of subgoals

*in Shneiderman (1987)
**in Houghton, Jr.(1984)

Table A3b.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS ASSISTANCE	USER CHARACTERISTICS 0 # of subjects	FINDINGS
Black, et.al., 1989	general-specific/ explanation-specific/ specific-specific/ explicit instructions	grade 10,11, minimum computer experience (42)	initial learning faster with explicit instruction; task time faster and more accurate with general to specific instruction, less errors with explanation- specific than specific-specific or explicit instruction
Carroll & Aaronson, 1988	how-to-do-it/how-it-works manual	experience with computers, not business software (8)	both types of help needed (exploratory)
Catrambone & Carroll, 1987	training wheel/full system		training wheels accelerated learning and encouraged learning more features of a wordprocessor
Elkerton & Williges, 1987	step-by-step/incomplete assistance	novices	step-by-step was preferred but did not improve transfer performance on file searching tasks
Holt, et.al., 1986	general global/ detailed step-by-step/ combined global & detailed	high/medium/low experienced with computers (198)	global inferior to detailed or combined; more experience leads to faster task completion; as experience increases so do favourable perceptions with a decrease in the referral to documentation
Schlager & Ogden, 1986	original vs. revised expert cognitive model/novice conceptual vs. procedural vs. neither	college students with no prior computer database management system experience (16)	revised text superior for initial training & retention; both models faster than without, but not from each other; for unknown condition searches conceptual model faster
Charney & Reder, 1986	task oriented/step-by-step		task-oriented assistance improved performance on transfer tasks
Carroll, 1985	minimal/full manual	novice (49)	minimal outperformed full manual on initial and transfer tasks
Carroll & Kay, 1985	training wheel scenario/full system		the scenario training wheel format decreased training time, but transfer was not affected
Borenstein, 1985*	UNIX manual/ natural language/ human	novice/expert computer users	human assistance speeded performance; content of text more important than mode of delivery
Carroll & Mack, 1984	discovery/manual	novice	self-discovery preferred over manual in wordprocessing task (exploratory)

Table A3b.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS ASSISTANCE	USER CHARACTERISTICS () # of subjects	FINDINGS
Cordes, 1984	online/online+ manual	some experience with similar system (16)	relied on online help, commands outperformed forms; original forms poorer than either commands or improved forms
Watley & Mulford, 1983*	online/hardcopy		less time and greater comprehension with hardcopy
Magers, 1983*	online/enhanced online	novice (30)	revised online with context sensitive help, examples, synonyms, task-oriented examples improved performance
Shneiderman, 1982 Exp.I	current error messages/improved tone/increased specificity/tone & specificity	intro computer undergrads (40)	increased performance & subjective evaluations significant for specificity only; all better than current messages
Exp.II	text-editor/editor+ message	undergrads with computer experience, but not with UNIX (21)	message improved performance & satisfaction with UNIX system
Exp.III	current message/ specific-courteous/ hostile-vague	intro Fortran course students with computer experience, no text-editor experience	specific-courteous outperformed current but not hostile-vague message, and was preferred over all
Exp.IV	current message/ improved message	intro Cobol undergrads, familiar with UNIVAC job-control language (66)	improved message outperformed current message on UNIVAC & IBM
Roemer & Chapanis, 1982*	written levels of tutorials	technical/nontechnical subjects with low/middle/high reading ability (54)	higher reading ability led to significantly different comple-tion times, errors, scores; increased complexity in writing style did not affect performance but preference was for 5th grade level
Cohill & Williges, 1982	no online/online/ printed manual	novice (72)	no online performed worse than other groups; performance best for user-initiated and selected help, with printed manual
Dunsmore, 1980**	brief offline summary only/brief summary+ online manual/brief summary+offline manual	novice (12)	summary with offline manual performed best; online worst

Table A3b.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS ASSISTANCE	USER CHARACTERISTICS () # of subjects	FINDINGS
Relles, 1979**	levels of online sophistication with & without printed manual/printed manual only	novice/experience with computer (6/30)	novices performed poorly with online assistance compared to manual only; performance better with online than manual only for experienced users
Olfman & Bostrom, 1988	application-based/ construct-based		application-based training on Lotus 1-2-3 resulted in greater use after training than construct-based training
Nakatani, et.al., 1987*	mastery tutorial/ conventional tutorial		mastery tutorial produced performance far superior to conventional but at the expense of greater time
Hiltz & Kerr, 1986	printed/skimmed/ online/personal/ group/ human/trial		one third never used advanced features, trial & error most likely method of learning; most users skim documentation and these log less time online; offline documentation rarely read; formal training takes longer to reach mastery level; online guided tutorial to novices increased subsequent use (exploratory)
Czaja, 1986	online tutorial/document-based/instructor-based		online tutorial resulted in worse performance on transfer task than others
Goodwin & Sanati, 1986	traditional course/ concrete model	novice Pascal students (600)	in traditional course material group previous experience predicted performance in programming but not in concrete models group
Bayman & Mayer, 1984	line model/stack/ no model	intro Psychology students with no previous computer programming experience, casual calculator users (72)	user's mental models differ & maybe inaccurate: teaching the model improved performance
Egan, et.al., 1987	diagnostic/remedial assessment of errors		small set of error patterns occurred in sizable portion of users & indicates areas of missing knowledge; both assessments reduced tendency to make errors repeatedly

*In Shneiderman (1987)
**In Houghton, Jr (1984)

Table A3c.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS			USER CHARACTERISTICS	FINDINGS
	ASSISTANCE	FEATURES	INTERACTION	0 # of subjects	
Karat, et.al., 1986				naive/experienced with wordprocessing system on new system (20/40)	performance lower compared to group on one system only; syntax differences caused subjects difficulty
Ledgard, et.al., 1980*			symbol/keyword commands	inexperienced/ familiar/experienced text editors (8)	keyword improved performance but less so for experienced users
Burns, et.al., 1986		reformatted/ current display		novice/expert	speed & accuracy improved for novices, experts had fewer errors, no change in times
Whiteside, et.al., 1985			7 different DBMS systems with menu/ icon/command		command best for all; novices great difficulty with menus
Elkerton & Williges, 1984				novice/expert	novices scrolled and paged through files, experts also used string search procedures
Murray, et.al., 1983*			voice/cursor	novice/experienced typist	both preferred keyboard, except initially novices preferred voice, no difference in performance
Gilfoil, 1982*			menu/command	novice to experienced over time	use of help facility decreased as experience increased, to point switch to command mode when help again sought till mode mastered
Walther & O'Neil, 1974			flexible/inflex-ible text editor	novice/expert	novices performance quicker with inflexible system, experts with flexible
Relles, 1979**	levels of online sophistication with & without printed manual/printed manual only			novice/experience with computer (6/30)	novices performed poorly with online assistance compared to manual only; performance better with online than manual only for experienced users

Table A3c. Research into Factors Affecting Performance with Software					
AUTHORS	DESIGN CHARACTERISTICS			USER CHARACTERISTICS 0 # of subjects	FINDINGS
	ASSISTANCE	FEATURES	INTERACTION		
Roth, Bennett, Woods, 1987				technicians with limited experience-limited training on previous & current system/ extensive experience- limited training on previous system/ extensive experience & training on both systems (1/2/1)	most & least experience took active role interacting & making decision based on machine expert which led to better & faster solutions; misinterpretations resulted from lack of knowing intended purpose of machine's request
Roberts & Moran, 1983				technical/nontechnical	non-technical skilled typists slower than technical non-skilled typists & spent three times longer in error states
Doane, 1986				novice/expert UNIX users	experts & novices had same lower level models of UNIX structure & information flow but differ on higher levels
Smith, 1983*		response times	lightpen	skilled vs. novice CAD designers	reduced response time increased interaction rates for both groups
Long, 1976*		response times		skilled/unskilled typists	longer response times resulted in reduced performance measures over time for both
Gugerty & Olson, 1986				novice/experienced with programming (18/6) Logo (10/10) Pascal	experts debug faster and more accurately and run program checks more often; novices add bugs; both equally studied program & description of problem
Dannenbring, 1983*		response times		novice/expert programmers	no effect of response time on peformance or satisfaction of either group, though scores between groups differ
Hauptmann & Green, 1983			menu/command /natural language	novice to experienced programmers with no package experience (12/12/12)	no difference in time or accuracy
Evans & Simkin, 1989				students of required intro computer course	no single variable of demographic, behaviour, cognitive or problem solving factors was best predictor of computer proficiency

Table A3c.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS: ASSISTANCE	FEATURES	INTERACTION	USER CHARACTERISTICS 0 # of subjects	FINDINGS
Ogozalek & Praag, 1986			voice/keyboard	elderly/younger (12/12)	no performance difference on age or input device, but voice preferred, elderly more motivated
Gomez, Egan & Bowers, 1986					high deductive reasoning skills associated with high performance on advance editing exercises, performance increased with lower age
Egan & Gomez, 1985				novice	low spatial memory test score equated with low performance; typing skill was a predictor only after experience with editing system gained
Roemer & Chapanis, 1982*	written levels of tutorials			technical/nontechnical subjects with low/middle/high reading ability (64)	higher reading ability led to significantly different completions times, errors, scores.increased complexity in writing style did not affect performance but preference was for 5th grade level
Bevan, 1981*		display rates		low & high reading abilities (24)	lesson time decreased as display rates increased but errors increased, high rate disliked most by both groups, with low ability preferring lowest rate
Bostrom, Olfman, & Sein, 1990	abstract(same system) vs. analogical(diff. system) models/ application- based (own problem) vs. construct-based (syntax,functions of system) training			undergrad/MBA/ undergrad/full-time employees (19/29/102/61) Kolb learning styles active(AE)/ concrete(CE)/ reflective(RO)/ abstract(AC)	abstracts performed better than concretes but not significantly; actives somewhat better than reflectives; abstracts and actives better with abstract model, reflectives with analogical, concretes and abstracts motivation higher with application based training, concretes with construct based
Kern & Matta, 1988				Myers Brigg Type Indicator (MBTI) sensing/judging/intuitive/perceptive (90)	sensing individuals performed better than intuitives on self-paced Lotus 1-2-3 task
Ambardar, 1984			sequential item # /keyword command for database access	field dependent (FD)/ field independent (FI)	FD preferred sequential item # search, FI keyword; both took longer if used non-preferred, FI separated functional use of commands not FD

Table A3c.

Research into Factors Affecting Performance with Software

AUTHORS	DESIGN CHARACTERISTICS			USER CHARACTERISTICS	FINDINGS
	ASSISTANCE	FEATURES	INTERACTION	() # of subjects	
Borgman, 1984,86					science & engineering majors more successful at online library search than humanities & social sciences; most frequent library users had most difficulty with online search

*In Shneiderman (1987)
**In Houghton, Jr.(1984)

Appendix K

Paraphrased Comments of Participants concerning Packages

Merlin$_t$

1. - should have example off-screen of package structure, such as,

	Index	Pg	Pg 2	Pg 3
A	—		A	B
B	—			
C	—			

2. - hard to get out of function once in it
   - saving and retrieving unclear - spent half the time trying to find file

3. - help system good, but how to save need clarification
   - would prefer full screen editor to enter data
   - edit is clear because of examples

4. - trouble retrieving and no explanation given for wrong commands used

5. - enjoyed very much
   - simple to use and understand, but would be difficult for computer novices
   - terminology technical and command procedures difficult to understand, e.g. formatting saved files, edit commands

6. - difficulty finding listing procedure

7. - package good for novices like myself
   - guides you along
   - the index was unclear and confusing
   - CATA (built-in help) more helpful
   - *have experience with Minitab$_t$ and Merlin$_t$ much easier*

8. - for novice, package very confusing

9. - statistics package of little use in work place
   - needs better instructions and organization
   - not very helpful with problems

10. - frustrating without access to human assistance

11. - extra screen helpful, especially for examples
    - easy to more around package
    - edit has too much information to take in

12. - editing long and complicated to do when make error half way through, ended up reentering everything

13. - useful to know can use numeric keypad to enter data
    - windows may help to be able to view data on the screen which scrolls off

14. - package easy to use, especially with extra help screen and ?
    - saving and retrieving not clear though should be part of major help
    - maybe my own fault. I am not really an expert user

15. - used other software, albeit, seldom, but this one annoying
    - could not figure out what to do with the instructions

Paraphrased Comments of Participants concerning Packages

Minitab,

1. - prior experience with Lotus, and programming languages needed
   - enjoyed participating
   - feels novices would need explanation of files and worksheets
   - scrolling through help tedious
   - would prefer 'hot-key' to access help than having to keep typing string of commands

Appendix L

Table A4
Performance Results by Package

		Total	Merlin,	Minitab,	Extended Merlin,
		243	99	93	51
Score(1)	N Mean s.d.	229 61.2 32.7	97 64 6 34 0	82 54.3 31.0	50 65.8 31 4
Score(2)	N Mean [illegible]	130 77.7 23 6	35 86 3 16 1	48 70.3 23.7	47 78.7 26.2
Time(1)	[illegible] Mean s.d.	231 49.7 14 4	95 43.1 14.3	86 54.1 13.1	50 54.5 12.2
Time(2)	N Mean s.d.	130 37 8 18.8	34 30.6 16.3	49 44.9 16 6	47 35 6 20.5
Tasks(1)	N Mean s.d.	229 3.7 1.9	97 4.1 1 8	82 3.2 1.8	50 3 9 1 9
Tasks(2)	N Mean s.d.	130 4 6 1.4	35 5.2 1.0	48 4.1 1 4	47 4.6 1.5

Table A5

Distribution of Performance Scores by Package

	Percent Distribution	Merlin	Minitab	Extended Merlin
TIDX(1)	N	95	82	50
	Mean	178.7	109.3	133.8
	s.d.	140.0	75.4	81.5
	Lower25%	0 - 72	0 - 65	0 - 82
	50%	72 - 241	65 - 166	82 - 178
	15%	241 - 355	166 - 205	178 - 235
	Upper10%	355 - 645	205 - 363	235 - 375
TIDX(2)	N	50	47	31
	Mean	334.8	287.7	265.9
	s.d.	226.5	235.6	242.5
	Lower25%	0 - 141	6 - 110	0 - 132
	50%	141 - 481	110 - 414	132 - 284
	15%	481 - 665	414 - 661	284 - 597
	Upper10%	665 - 1050	661 - 991	597 - 1100
EIDX(1)	N	91	71	46
	Mean	59.1	27.8	33.8
	s.d.	101.8	21.9	64.8
	Lower25%	0 - 9	0 - 11	2 - 8
	50%	9 - 47	11 - 40	8 - 32
	15%	47 - 208	40 - 60	32 - 65
	Upper10%	208 - 520	60 - 100	65 - 400
EIDX(2)	N	27	33	26
	Mean	42.3	28.9	12.0
	s.d.	83.4	26.7	17.8
	Lower25%	0 - 2	0 - 8	0 - 2
	50%	2 - 24	8 - 42	2 - 12
	15%	24 - 155	42 - 75	12 - 38
	Upper10%	155 - 360	75 - 103	38 - 76
HIDX(1)	N	91	71	N.A.
	Mean	77.1	57.7	
	s.d.	140.0	50.6	
	Lower25%	0 - 15	7 - 26	
	50%	15 - 49	26 - 74	
	15%	49 - 264	74 - 125	
	Upper10%	264 - 760	125 - 320	
HIDX(2)	N	27	33	N.A.
	Mean	58.2	38.6	
	s.d.	108.1	30.9	
	Lower25%	4 - 9	0 - 16	
	50%	9 - 60	16 - 46	
	15%	60 - 181	46 - 100	
	Upper10%	181 - 520	100 - 116	

Legend: N.A. - not applicable
(1) - Indicates Initial Use
(2) - Indicates Retest

Appendix M

Table 15
General Spreadsheet Package Experience

		Total		Merlin$_t$		Minitab$_t$		Extended Merlin$_t$	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
Spreadsheet	Total	172	100.	72	100.	67	100	33	100.
	1- Lotus	161	93 6	67	93 0	64	95 5	30	91 0
	2- Supercalc	1	0.6	1	1.4	0	0 0	0	0 0
	3- Multiplan	3	1.7	2	2.8	0	0 0	1	3 0
	4- Quatro	2	1.2	0	0 0	1	1.5	1	3 0
	5- Other	5	2 9	2	2 8	2	3 0	1	3 0
Spreadsheet Experience	Total	239	100.	95	100.	93	100.	51	100.
	1- Novice	116	48.5	41	43 2	43	46 2	32	62.7
	2- Intermediate	117	49.0	52	54.7	46	49 5	19	37 3
	3- Expert	6	2 5	2	2.1	4	4 3	0	0 0
Frequency of Use of Spreadsheet	Total	170	100.	71	100.	66	100	33	100.
	1- Nev.r	15	8 8	5	7 0	7	10 6	3	9 1
	2- < Once/mth	92	54.1	37	52 1	34	51 5	21	63 6
	3- Once/mth	15	8 8	6	8 5	8	12 1	1	3 0
	4- Few times/mth	29	17.1	14	19.7	10	15 2	5	15 2
	5- Few times/wk	16	9.4	8	11.3	5	7 6	3	9 1
	6- Once a day	1	0 6	0	0 0	1	1 5	0	0 0
	7- Several times/dy	2	1 2	1	1 4	1	1 5	0	0 0
Perceived Command Structure of Spreadsheet	Total	162	100.	68	100.	65	100.	29	100.
	1- Menu	58	35.8	29	42.6	17	26.2	12	41 4
	2- Command	5	3 1	3	4.4	2	3.1	0	0 0
	3- Mixed	97	59 9	35	51.5	45	69 2	17	58 6
	4- Mouse	2	1.2	1	1.5	1	1 5	0	0 0
Purpose for which Spreadsheet Used	Total	167	100	70	100.	65	100	32	100
	1- Work	10	6 0	5	7 2	3	4 6	2	6 3
	2- School	123	73 7	49	70 0	52	80 0	22	68 7
	3- Personal	10	6.0	1	1.4	7	10 8	2	6 3
	4- Work/School	4	2.4	3	4.3	1	1 5	0	0 0
	5- School/Personal	9	5 4	4	5 7	2	3 1	3	9 3
	6- Work/Personal	5	3.0	4	5 7	0	0 0	1	3 1
	7- All of the above	6	3.6	4	5 7	0	0 0	2	6 3
Where Learnt Spreadsheet	Total	165	100	68	100	65	100	32	100
	1- Course	118	71.5	48	70 6	48	73 8	22	68 7
	2- Self-taught	33	20 0	15	22.1	12	18 5	6	18 8
	3- Both	14	8 5	5	7.3	5	7.7	4	12 5

Table 16
General Wordprocessing Package Experience

		Total		$Merlin_t$		$Minitab_t$		Extended $Merlin_t$	
		#	%	#	%	#	%	#	%
		243	100	99	40.7	93	38 3	51	21.0
Word-processor	Total	155	100.	66	100.	58	100	31	100.
	1- Wordperfect	140	90.3	63	95 5	46	79 3	31	100.
	2- Wordstar	7	4 5	1	1.5	6	10.4	0	0.0
	3- Multimate	1	0 6	0	0.0	1	1.7	0	0 0
	4- Other	7	4.5	2	3 0	5	8 6	0	0 0
Word-processor Experience	Total	236	100.	95	100.	90	100.	51	100
	1- Novice	116	49 2	39	41.1	47	52.2	30	58 8
	2- Intermediate	101	42 8	46	48 4	36	40 0	19	37.3
	3- Expert	19	8 0	10	10 5	7	7 8	2	3.9
Frequency of Use of Word-processor	Total	156	100.	66	100.	58	100.	32	100.
	1- Never	14	9 0	1	1.5	8	13 8	5	15 6
	2- < Once/mth	45	28 8	19	28 8	13	22.4	13	40 6
	3- Once/mth	25	16 0	10	15.1	14	24.1	1	3.1
	4- Few times/mth	46	29 5	25	37.9	15	25 9	6	18 8
	5- Few times/wk	17	10 9	5	7 6	6	10 4	6	18 8
	6- Once a day	2	1 3	1	1.5	1	1.7	0	0.0
	7- Several times/dy	7	4 5	5	7.6	1	1.7	1	3.1
Perceived Command Structure of Word-processor	Total	147	100.	64	100.	54	100.	29	100.
	1- Menu	44	29 9	17	26 6	17	31.5	10	34 5
	2- Command	23	15 7	15	23 4	5	9 2	3	10 3
	3- Mixed	74	50.3	30	46 9	28	51 9	16	55 2
	4- Mouse	6	4.1	2	3.1	4	7 4	0	0 0
Purpose for which Word-processor Used	Total	155	100.	66	100.	57	100.	32	100.
	1- Work	8	5 2	6	9 0	2	3 5	0	0 0
	2- School	80	51 6	31	47 0	31	54 4	18	56.2
	3- Personal	13	8.4	2	3 0	10	17 5	1	3.1
	4- Work/School	7	4.5	4	6.1	1	1.8	2	6.3
	5- School/Personal	26	16 8	11	16 7	10	17.5	5	15.6
	6- Work/Personal	8	5 2	4	6.1	2	3 5	2	6.3
	7- All of the above	13	8 4	8	12.1	1	1.8	4	12.5
Where Learnt Word-processor	Total	156	100.	65	100.	58	100.	33	100.
	1- Course	71	45 5	[illegible]	41.5	24	41 4	20	60 6
	2- Self taught	69	44 2	31	47 7	29	50 0	9	27.3
	3- Both	16	10 3	7	10 8	5	8 6	4	12.1

Table 17
General Database Package Experience

		Total		$Merlin_t$		$Minitab_t$		Extended $Merlin_t$	
		#	%	#	%	#	%	#	%
		243	100	99	40 7	93	38 3	51	21 0
Database	Total	61	100	25	100.	26	100.	10	100
	1- Dbase	58	95 2	24	96 0	25	96.2	9	90 0
	2- Smart	1	1 6	1	4 0	0	0 0	0	0 0
	3- Framework	1	1 6	0	0 0	1	3 8	0	0 0
	4- Other	1	1 6	0	0 0	0	0 0	1	10 0
Database Experience	Total	240	100	96	100	93	100	51	100
	1- Novice	203	84 6	77	80 2	78	83 9	48	94 1
	2- Intermediate	34	14 2	17	17 7	14	15 0	3	5 9
	3- Expert	3	1 3	2	2 1	1	1.1	0	0 0
Frequency of Use of Database	Total	62	100.	26	100	26	100	10	100
	1- Never	14	22 6	3	11 5	7	26 9	4	40 0
	2- < Once/mth	25	40 3	12	46 2	9	34 6	4	40 0
	3- Once/mth	7	11 3	2	7 [illegible]	5	19 2	0	0 0
	4- Few times/mth	9	14 5	6	23.1	2	7.7	1	10 0
	5- Few times/wk	4	6 5	0	0 0	3	11 6	1	10 0
	6- Once a day	2	3 2	2	7 7	0	0 0	0	0 0
	7- Several times/dy	1	1.6	1	3 8	0	0 0	0	0 0
Perceived Command Structure c Database	Total	57	100	25	100.	25	100	7	100
	1- Menu	17	29 8	7	28 0	9	36 0	1	14 3
	2- Command	12	21.1	6	24 0	5	20 0	1	14 3
	3- Mixed	28	49 1	12	48 0	11	44 0	5	71 4
	4- Mouse	0	0 0	0	0 0	0	0 0	0	0 0
Purpose for which Database Used	Total	61	100	26	100	26	100	9	100
	1- Work	5	8 2	3	11.6	1	3 8	1	11 1
	2- School	47	77.1	19	73 1	21	80 8	7	77 8
	3- Personal	7	11 5	2	7 7	4	15 4	1	11 1
	4- Work/School	1	1.6	1	3 8	0	0 0	0	0 0
	5- School/Personal	0	0 0	0	0 0	0	0 0	0	0 0
	6- Work/Personal	1	1 6	1	3 8	0	0 0	0	0 0
	7- All of the above	0	0 0	0	0 0	0	0 0	0	0 0
Where Learnt Database	Total	62	100	26	100.	26	100	10	100
	1- Course	46	74 2	18	69 2	20	76 9	8	80 0
	2- Self-taught	13	21 0	7	26 9	4	15.4	2	20 0
	3- Both	3	4 8	1	3 9	2	7 7	0	0 0

Table 18
General Programming Language Experience

		Total		$Merlin_t$		$Minitab_t$		Extended $Merlin_t$	
		#	%	#	%	#	%	#	%
		243	100	99	40.7	93	38.3	51	21.0
Programming Language	Total	122	100	45	100.	51	100.	26	100.
	1- Basic	106	86 9	37	82.2	46	90 2	23	88 5
	2- Fortran	5	4.1	5	11.1	0	0 0	0	0.0
	3- Pascal	7	5.7	2	4.5	3	5 9	2	7.7
	4- Other	4	3 3	1	2 2	2	3 9	1	3.8
Programming Language Experience	Total	239	100	96	100.	92	100.	51	100.
	1- Novice	161	67.3	66	68 8	58	63 0	37	72.5
	2- Intermediate	68	28 5	22	22.9	32	34.8	14	27.5
	3- Expert	10	4 2	8	8.3	2	2 2	0	0 0
Frequency of Use of Programming Language	Total	116	100.	43	100.	48	100.	25	100.
	1- Never	29	25 0	11	25.7	8	16.7	10	40 0
	2- < Once/mth	62	53 4	23	53.5	27	56 2	12	48 0
	3- Once/mth	14	12.1	5	11 6	8	16.7	1	4.0
	4- Few times/mth	5	4 3	1	2 3	3	6 2	1	4.0
	5- Few times/wk	4	3.4	2	4 7	1	2.1	1	4 0
	6- Once a day	1	0 9	1	2.3	0	0.0	0	0 0
	7- Several times/dy	1	0.9	0	0 0	1	2 1	0	0 0
Perceived Command Structure of Programming Language	Total	105	100.	38	100.	45	100.	22	100.
	1- Menu	15	14.3	5	13.2	7	15 6	3	13 6
	2- Command	47	44 8	21	55 3	15	33 3	11	50 0
	3- Mixed	42	40 0	11	28.9	23	51.1	8	36.4
	4- Mouse	1	0.9	1	2 6	0	0 0	0	0.0
Purpose for which Programming Language Used	Total	111	100.	42	100.	46	100.	23	100.
	1- Work	7	6.3	4	9 5	3	6 5	0	0 0
	2- School	91	82.0	35	83 3	35	76 1	21	91.4
	3- Personal	6	5 4	2	4 8	3	6 5	1	4.3
	4- Work/School	0	0 0	0	0 0	0	0 0	0	0.0
	5- School/Personal	3	2.7	0	0.0	3	6.5	0	0 0
	6- Work/Personal	4	3.6	1	2.4	2	4 4	1	4.3
	7- All of the above	0	0 0	0	0 0	0	0 0	0	0 0
Where Learnt Programming Language	Total	110	100.	40	100.	47	100.	23	100
	1- Course	92	83 6	33	82 5	39	83 0	20	87 0
	2- Self-taught	10	9.1	4	10 0	4	8 5	2	8.7
	3- Both	8	7 3	3	7.5	4	8.5	1	4 3