National Library of Canada

Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch

Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Testing the Effectiveness of an Automated Tool to Evaluate the Ease of Use and the Ease of Learning of a Computer Interface

Michel Savoie

A Thesis in the Faculty of Commerce and Administration
Master of Science in Administration

Presented in Partial Fulfillment of the Requirements for the Master of Science in Administration at Concordia University
Montreal, Quebec, Canada

December 1994

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN IRREVOCABLE NON-EXCLUSIVE LICENCE ALLOWING THE NATIONAL LIBRARY OF CANADA TO REPRODUCE, LOAN, DISTRIBUTE OR SELL COPIES OF HIS/HER THESIS BY ANY MEANS AND IN ANY FORM OR FORMAT, MAKING THIS THESIS AVAILABLE TO INTERESTED PERSONS.

L'AUTEUR A ACCORDE UNE LICENCE IRREVOCABLE ET NON EXCLUSIVE PERMETTANT A LA BIBLIOTHEQUE NATIONALE DU CANADA DE REPRODUIRE, PRETER, DISTRIBUER OU VENDRE DES COPIES DE SA THESE DE QUELQUE MANIERE ET SOUS QUELQUE FORME QUE CE SOIT POUR METTRE DES EXEMPLAIRES DE CETTE THESE A LA DISPOSITION DES PERSONNE INTERESSEES.

THE AUTHOR RETAINS OWNERSHIP OF THE COPYRIGHT IN HIS/HER THESIS. NEITHER THE THESIS NOR SUBSTANTIAL EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT HIS/HER PERMISSION.

L'AUTEUR CONSERVE LA PROPRIETE DU DROIT D'AUTEUR QUI PROTEGE SA THESE. NI LA THESE NI DES EXTRAITS SUBSTANTIELS DE CELLE-CI NE DOIVENT ETRE IMPRIMES OU AUTREMENT REPRODUITS SANS SON AUTORISATION.

ISBN 0-612-01322-7

Canada

ABSTRACT

Testing the Effectiveness of an Automated Tool to Evaluate the Ease of Use and the Ease of Learning of a Computer Interface

Michel Savoie

This research investigates the effect of using an automated tool to evaluate human-computer interfaces. Such tools have the potential to greatly facilitate the evaluation of computer interfaces. Applying task-theoretic analytic models is one method to measure the effectiveness of an interface These models provide the means to measure interface parameters such as ease of use and ease of learning. One of the difficulties of applying task-theoretic analytic models to measure human factors such as the ease of use and the ease of learning of an interface is the complexity of the models' task representational language. An automated tool simplifies the evaluation process by providing a method to represent interfaces graphically This representation ensures that users can apply analytic models without learning their complicated syntax. An automated tool has the potential to improve the accuracy and speed of an interface evaluation. Until now there has not been any empirical studies conducted to verify these effects. This research paper confirms that the use of an automated tool helps designers to evaluate interfaces more accurately and in less time. Furthermore, the users of the automated tool have a more positive attitude towards the application of the evaluation models to measure the ease of use and the ease of learning of an interface. This study demonstrate that an automated graphical representation tool, such as the one suggested by Khalifa and Kira (1992), improve the interface evaluation process.

TABLE OF CONTENT

I. INTRODUCTION

Historically, the use of information technology was confined to the scientists and computer engineers who were predominately concerned with the efficient use of the limited processing power of the machines available. This was the age of mainframes where economics dictated that training users was better then investing in the development of effective interfaces. Devoting human effort and computing power to the interface was considered a luxury and in many instances was not even necessary since only expert users were involved with the technology.

The fall in cost of computational power has changed all of this. Today, information technology is present in every business activity and the use of computers is no longer confined to the "experts". Information technology is now pervasive and many users are now people who have little technical skill. The user interface-the aspects of the system that the user comes in contact with physically, perceptually, and conceptually- has now become critical. In today's environment where many software applications are aimed at the mass market, an interface that is easy to use and easy to learn may be the determining factor of the product's success. Computer manufacturers such as Apple Computers have had a tremendous success in the marketplace due partly to the efforts that they have devoted to the development of the human-computer interface of their systems.

It has been demonstrated that a good interface can increase the ease of use, reduce the learning time, reduce the number of errors committed by the user, increase satisfaction of the user with the system, and finally improve the user's retention of the system's functionality and syntactic aspects- all impacting the productivity of the user (Nickerson, 1981, Card et al., 1983, Norman, 1986; Gould et. al., 1987; Shneiderman, 1987; Bailey et. al.,1988). The realization that the computer interface can have a significant impact on the productivity of the user and on the success of a product has motivated many researchers to study human-computer interaction.

The basic premise of a good human-computer interface is that it should be compatible with the user's mental model and expectations. This seemingly simple idea is difficult to apply since the designer often has a different conceptual model of the interface then that of the user Bridging the gap between the design model and the user's model is the primary goal of a designer developing an interface (Norman, 1984). In the last decade much of the research effort in the field of human-computer interaction has focused on determining the best way to develop interfaces that are compatible with the user's mental model.

Designing a good interface is extremely difficult, time consuming, and can easily double the cost of developing a system (Benbasat, et, al., 1984) The designer must be aware of the design alternatives and understand their impact on the user This requires an in-depth knowledge of the psychology of the user An adequate model of the brain, a complete human psychology, or even a comprehensive study of human factors in human-computer interaction (HCI) could greatly facilitate the task of the interface designer Unfortunately, none of these three items are available and consequently, designing a good interface is quite challenging. For this reason several approaches have been suggested to aid the designer in the task of creating effective interfaces These techniques are an attempt to improve the productivity and the effectiveness of the interface designer

Initially, the development of design guidelines was suggested The purpose of these guidelines were to aid the developer in making interface design decisions Guidelines would, for example, state that semantic organization of menu items are preferable. It was soon realized that this approach was not feasible since the design of the interface is dependent on the task context and the user behavior, consequently, the usefulness of design guidelines diminished (Gould and Lewis, 1985, Moran, 1981) The solution was to develop methods that model the behavior of the users performing specific tasks. This approach is heavily dependent on the understanding of cognitive processes Examples of human-computer interaction (HCI) models to predict user behavior are the Model Human Processor (Card et al., 1983) and Soar (Laird, et al , 1987) Methods to predict the ease of use and ease of learning based on theoretical models were suggested (Card, et al., 1983; Kieras and Polson, 1985, and Khalifa, 1991) Although a promising approach, these models have proven difficult to apply in real life development projects because of the complicated nature of the theories used In a survey conducted by Gould and Lewis (1985) among interface developers, the majority of designers agreed that human factors are important but that the research findings in HCI is too complex and cumbersome and consequently not feasible for use in real-life projects

Several authors have suggested the use of iterative design methodologies (Anderson and Olson, 1985; Shneiderman, 1985; Gould, 1985) to ensure the development of good interfaces. The premise of this approach is that the only way to develop a good interface is to first build a prototype of the interface and test it with the actual users The interface is tested and revised iteratively until it complies with human factor goals The effectiveness of this method has been validated in real life projects (Gould and Lewis, 1987)

Without an easily applied theoretical construct of the psychology of the human-computer interaction, the only effective method to develop an interface is iteratively. In their survey of interface developers Gould and Lewis (1987) found that iterative design methodologies are seldom applied because they are too expensive, time consuming, risky, and that meeting deadlines is the critical factor and is incompatible with the methodologies Iterative design methodologies seem to be one of the roadblocks to the application of human factors to interface design.

Several researchers (Card, et al , 1983; Reisner, 1984; Anderson and Olson, 1985; Kieras and Polson, 1985; and Khalifa, 1991) have suggested that the use of analytic models can ameliorate the interface design on paper before building a prototype . This approach can significantly reduce the number of iteration that must be performed to improve the prototype and obtain an adequate interface Several analytic models have been developed to accurately predict human factors such as ease of use and ease of learning. The interface designer can develop an interface design on paper and predict the learning time and the execution time of a particular task. These are two important parameters of an effective interface. This process of designing an interface and measuring human factors goal can be repeated until the interface is deemed adequate. A real prototype can then be developed and tested with the actual users.

Khalifa and Kira (1992) suggest that the analytic models are often not used due to the complexity ot iearning their formal representational languages and their task analysis methodologies. The actual mechanics of applying a model such as TAMPEL (Khalifa, 1991) to measure the ease of learning of an interface is too complex for the average programmer who has not been trained in cognitive psychology. Finding a way to apply these types of models to the development of interfaces has the potential to: 1) Significantly reduce the number of design iterations 2) Improve the actual interface and 3) Improve designer productivity.

To alleviate the complexity associated with the application of analytic models Khalifa and Kira (1992) propose the use of an Interface Description and Evaluation System (IDES). IDES is an automated tool that assists the designer in describing and evaluating an interface design according to several possible analytic models without having to understand the specific details of these models The designer uses IDES at the evaluation stage of the paper-and-pencil interface The authors argue that such a decision support system will make it more attractive for designers to use proven analytic models in the design process of an interface

The purpose of this research is to assess the effectiveness and efficiency of using IDES as a tool to evaluate a human computer interface Effectiveness is defined as the accuracy of measuring ease of use and ease of learning with IDES Efficiency is defined as the time it takes to measure ease of use and ease of learning A controlled experiment is designed to compare two groups, one using IDES and another (control group) using a manual method to evaluate a simple interface and a complex interface The two groups are compared on the speed and accuracy of their evaluation Also, their attitude towards the techniques used to evaluate the interfaces is assessed The main hypothesis of this research paper is that evaluating a compluter interface in terms of ease of use and ease of learning using a tool such as IDES is easier, faster, and more accurate then using the standard manual method. Providing means for the interface developer to effectively apply formal analytic models may be the best approach to improve interface development This paper intends to advance the research knowledge of HCI in attempting to demonstrate that formal models can be easily applied to evaluate human computer interfaces by using the IDES.

The paper is divided into three main sections. 1) a literature review, 2) a research methodology description, and 3) a discussion section explaining the results Much research effort has been devoted to human computer interaction The first part of the paper is a survey of these research findings More specifically the importance of human computer interface, the tools and methodologies used to develop interfaces, the different methods to evaluate interfaces, and a description of cognitive models used in HCI is discussed. Furthermore, a detailed description of the IDES proposed by Khalifa and Kira (1992) is provided. The next section reviews the research methodology used to answer the main hypothesis of this paper that the IDES provides more accurate interface evaluation in a more timely matter and that users have a positive attitude towards using such a tool. Finally, the results of the experiments are discussed based on the findings of the experiment

With information technology permeating every aspect of our day-to-day life the development of interfaces to accommodate the different users and consequently to improve productivity becomes more and more important. Although it has proven difficult to apply the significant research findings of HCI to interface development, we believe that the use of IDES can help designers apply complex psychological theories to interface design and ensure the inclusion of human factors in software development. It is hoped that one day users will no longer have to adapt to the computers to utilize it effectively- this will happen when HCI theory is completely integrated in the design process. This research paper proposes to take a small step ir [illegible]at direction

II. MOTIVATION FOR RESEARCH

Before proceeding further, it may be appropriate to review in more details the reasons why the study of human-computer interaction is so important The basic premise of HCI research is to help create interfaces that ensures easy and efficient use of the computer. With the widespread use of information technology this has become critical

An appropriate interface can reduce the time and effort required to learn a new system, consequently training cost can be reduced. For example, an airline company that is introducing a new reservation system may be able to eliminate on site training if the interface is self-explanatory and easy to use A software company may decrease the amount of support that is required for its customers during product introduction, if the interface is easy to learn. These examples demonstrate that easy to learn systems can save money

An appropriate interface can also affect productivity Bailey, et al ,(1988) demonstrated that the interface design of their application improved productivity by as much as 77% Gains in productivity can be obtained in several ways The interface can reduce the number of keystrokes that is required to perform a certain task This may seem insignificant, but in the case of a typing department in a law firm, for example, a 10% improvement is important Also, errors occurring when using the system can be reduced An interface that reduces errors can increase the user satisfaction and Benbasat and Dexter (1981) demonstrated that this can influence the use of the systems options and encourage exploration. Assuming that the system has the proper functionality this can only be positive. Furthermore, there are certain situations, such as an airline traffic control system, where there is no room for error. Finally, a good interface increases retention of how to use the system This ensures that users do not have to waist time re-learning the application. An effective interface can also reduce cognitive effort This means that and interface the requires little cognitive effort can have the user concentrate on the actual task. These factors all affect productivity and any organization that wants its information technology to be used effectively and efficiently cannot ignored the user interface

Hardware and software developers now realize that a good interface is critical to the survival of their product A product that is aimed at the mass market cannot hope to sell if it is not easy to use and easy to learn Functionality alone is no longer sufficient The development of walk-up-and-use systems such as automatic bank tellers make it necessary to develop applications that require minimum training. Nickerson (1981) conducted a study to determine why people did not use a system when it could benefit them. Seven of the ten detected reasons are directly related to the human-computer interface. Consequently, a poorly designed interface can ruin the chances of a product gaining acceptability in the marketplace.

The computer is a tool which was developed to help people do there job. The ease with which one communicates with the machine determines the extent to which it will be used and whether or not it is used efficiently depends upon the man-machine language that is available The ideal system is one where the users concentrate solely on the task that must be performed and is not hindered by the communication protocol between him/her and the computer. The interface is critical to ensure that computers are used to their fullest.

The importance of the human factors has led to the development of many tools and methodologies to ensure that human factors are properly integrated in software development. Unfortunately, many of the research findings in HCI are often not applied. The next section reviews some of these tools and provide explanations as to why interface designers often do not use the tools that are available to them.

Two important parameters of an interface are its ease of use and its ease of learning. Tools to measure these human factors attributes are cumbersome and difficult to integrate in the software development process (some of the reasons are discussed in a later section) This research paper proposes to test a method to facilitate the evaluation of the ease of learning and ease of use of an interface. This would greatly help the interface desigers in their goal of developing efffective interfaces. The use of automated tools has not yet been empirically tested.

III. TOOLS AND METHODOLOGIES IN HUMAN-COMPUTER INTERACTION

This section presents a review of design guidelines and principles as well as design methodologies. A section presents some of the reasons why some of these design approaches have not been fully adopted by interface designers. Finally, a hybrid design methodology is presented along with an explanation of how task-theoretic analytic models(methods to quantitatively measure human factor parameters of an interface such as ease of learning) can be integrated in the development of an interfaces.

A. Interface Design Guidelines

Historically, intuition guided the development of an interface and the designer relied on empirical studies with the actual users to determine the appropriateness of a design. This is difficult since the approach is completely theoretical and consequently, difficulties arise when determining what exactly needs to be tested The inefficiency of this approach led to the development of general design guidelines that could help the designer during the initial development of the interface.

Design guidelines were suggested to eliminate the need to test every possible alternative of an interface. The basic premise of this approach is that psychologists perform experiments in the laboratory to determine the general design features that affect the user behavior. The laboratory findings are then applied in the design process For example, an experiment may be conducted to determine if semantic organization of menu items or alphabetic organization is better

The development of human-computer interface can be very complex and consequently guidelines can be applied to many different parameters of an interface. Decision must be made about the best interaction style- menu selection system, command languages, or direct manipulation. The most appropriate way to display the data. The use of colors, graphs, and systems messages. Are icons more appropriate? When should animation be introduced? The list goes on This led to the publication of guideline books that attempt to provide guidance on all these different issues. Needless to say these books were voluminous and consequently became cumbersome to use.

Maguire (1982) published a study evaluating the different design guidelines offered up to that time. He found that many of the guideline books were too large to be used effectively, many of the recommendations did not provide empirical proofs. Furthermore, there were many guidelines that contradicted one another. Another problem associated with the proposed guidelines are that they do not add up to a coherent psychological picture (Moran, 1981). This is due to the fact that many of the guidelines suggested are dependent on contextual variables The types of users and tasks can have a significant impact on the findings and hence, the guidelines are not always applicable This is not to suggest that guidelines should be completely ignored but that they should be applied with care. Anderson and Olson (1985) believe that design guidelines provide suggestions to the designer that can be better then those based solely on intuition

Moran (1981) suggests that the only way to progress in the field of HCI is to move beyond the computer system and consider the user on his/her own psychological term. For the field of HCI to progress it has to move beyond design guidelines and attempt to develop a psychology of the user. Efforts to develop a psychology of the user lead to the introduction of design principles.

B. Design Principles

Design principles are distinguished from design guidelines in that they are based on sound theories that have been empirically tested. By being able to model the underlying user behavior, design principles that are not contextually constrained can be developed. By understanding the cognitive processes that influence human-computer interaction, the interface designer is no longer tied to a set of design guidelines. This is a promising approach to interface design and it stems from the suggestion by Card, et al. (1983) that the user's *recognition-action cycle* can be used as the basic behavior for understanding the psychology of HCI. Card, et al. (1983) extended psychological theory so as to account for interfaces in speed of use, ease of learning and other attributes that are relevant to the design of effective interfaces.

The recognition-action cycle includes three stages that explains user behavior. The first stage is perception. This is where the user perceives the computer presentation and encodes the information. In the second stage the user searches long and short-term memory to determine an appropriate response. Finally, in the execution stage the response is activated by sending out the signals for the motor processors to act A more elaborate model is proposed by Norman (1986) where the recognition-action cycle is expanded to include seven stages In Norman's model the memory stage includes mental activities such as interpretation, evaluation of possible system response, the formulation of goals and intentions, and finally the specification of the action sequence Olson and Olson (1990) suggest that there are four cognitive processes that are employed by the user motor movement, perception, cognition, and memory Each of these cognitive processes have limited capacity and consequently limit the user's behavior and the HCI design

The recognition-action framework provides a model to understand the psychology of the user behavior This can lead to the development of design principles Norman (1986) suggests that there is a gap between the designer system's model and the user's mental model The underlying principle is that the system model must be compatible with the user's understanding of the specific process that must be accomplished The idea is to match the system model to that of the user's mental model The user's mental model is the user's conceptualization of the system's components and the interaction of these components as well as how to change these components (Carrol and Olson 1988) Basing the system's model on the user's mental model ensures that the human computer interface is more effective. The recognition-action framework helps understand the way the user interacts with the system's model.

Understanding the stages of user activities in human-computer interaction has important implications for the development of design principles For example, the user's task at the intention formation stage can be facilitated by providing information about the current state of the system and explicit possible action (Norman 1984) The execution stage can be simplified by memory aids, menus, and pointing devices (Norman 1984) Norman (1986) suggests design principles based on his general theory of action He views the design of an interface as that of building two bridges, one across the gulf of execution, which separates the user's conception of a goal from the actions needed to accomplish it and a second, across the gulf of evaluation, which separates the results provided by the system to the user's understanding of the progress accomplished Norman's design principles are based on this analogy of building bridges

Some examples of his design principles are (Quoted from Norman, 1988) 1) Use both Knowledge in the world and Knowledge in the head, 2) Simplify the structure of the task, 3) Make things visible: bridge the gulfs of execution and evaluation, 4) Get the mapping right Make sure that the user can determine the relationships between (a) intentions and possible actions, (b) actions and their effects on the system, (c) the actual system state and what is perceivable by sight, sound, or feel, and (d) the perceived system state and the goals and intentions of the user, 5) Exploit the power of constraints, both natural and artificial, 6) Design for error, 7) When all else fails standardize The strength of these principles is that they are based on sound theories They are developed using an information processing model of the user and consequently can be generalized to different contexts

The problem with the principles approach is that it is often difficult to provide the designer with specific ways to improve the interface Molich and Nielsen (1990) have suggested more specific principles that are based on the recognition-action framework Some of these principles are 1) Use simple and natural languages, 2) Speak the user's language, 3) Minimize the user's memory load, 4) be consistent, 5) provide feedback, 6) Provide clearly marked exit, 7) provide shortcuts and, 8) provide good error messages Some of these principles can leave room for interpretation and many designers feel that they are too general Grudin (1989) emphasizes the problem by analyzing the consistency principle He states that there is a problem with this principle in that consistency must be defined It must be determined what is "good" consistency, and there must be a method to determine when other design considerations overshadow consistency in importance He believes that in certain instances striving for consistency may sacrifice ease of use for ease of learning which may not always be appropriate This provides an example of the difficulty of applying design principles

Useful principles are difficult to formulate Norman (1986) suggest that design principles have to be general enough so as to outlast the technological development that are taking place and at the same time they must be precise enough to provide the designer with usable tools to develop an effective interface Principles that declare that the designer must consider the user are valid but they may not be useful to help the designer in day-to-day development problems

Understanding what is required to develop a good interface further emphasizes why design principles are not sufficient As mentioned before, the central problem in HCI is the design of the system model-the conceptual description of how the system works Developing a system model is no easy task. First, an analysis of the user tasks is required in order to match the system model to the user understanding of these tasks (Carroll and

Thomas, 1982, Halasz and Moran, 1982, Moran, 1981) Second, metaphors and abstract models must be developed to adequately portray the system functionality (Carrol, et al., 1988). Design principles are of limited help to adequately perform these activities

Card, et al, (1983) and Norman (1986) suggest that during computer interaction, the user's mental activities revolve around goal determination and action planning Consequently, the interface must adequately support these activities Detailed task analysis must determine user goals and the methods and objects employed to achieve these goals (Grudin, 1989, Phillips, et al, 1988) The analysis of work activities and work related scenarios can be used to discover goals and methods (Young, et al, 1989) A scenario is a summary of a user interacting with a particular device responding to an event that is constructed so that the user performs a certain action (like deleting words in a document for example) Scenario analysis produces records of user actions and from those records specific user goals and, methods to achieve these goals are identified The detailed task analysis is performed by determining the cognitive processes that are involved in handling the events As mentioned before, design principles are of little help to the interface developer performing task analysis

Also, metaphors and abstract models can be extremely useful in developing the system model A metaphor can help the user relate the system model to concepts already known The user can consequently infer what actions are possible and what effect they may have on the system state Tools and systems that are used in the task domain is help discover appropriate metaphors (Carrol, et al, 1988) An example of a metaphor can be a typewriter for the use of a word processor Unfortunately, systems are often complex and one metaphor may not be sufficient The analogy of the typewriter breaks down when we refer to moving a whole paragraph to another section for example Abstract models are useful when it is difficult to find a real world representation of the system The purpose of the abstract model is to provide a semantic organization of the system for the user Again principles are of little help to develop an effective system's model

There has been much research effort devoted to develop a psychology of the user This could account for ease of use, ease of learning, error, satisfaction, and retention The basic premise of the principles approach to designing interface is that by understanding the user behaviour, principles that have a strong theorethical foundation can be suggested Consequently, these principles can be applied to a wide variety of situations The discussion about Norman's framework provide an example of the the type of principles that can be applied to interface development. Unfortunately, the task of developing an effective interface is complex and tools such as metaphors, abstract models, and detailed task analysis are necessary. Design priciples are not adequate in these instances. This has lead researchers to believe that task and metaphor analysis must be user centered and iterative (Gould, et al., Anderson and Olson, 1985; Buxton and Schneiderman, 1980, Gould, 1988; and Gould and Lewis, 1988). Iterative design methodologies are based on the premise that it is not possible to get it right the first time when developing an interface The following section presents some of the design methodologies that have been suggested and discusses their drawbacks.

C. Iterative Design Methodologies

This section describes iterative design methodology and explains why it is not widely accepted. As mentioned before theoretical knowledge does not yet provide the necessary tools to guide the designer in the development of an effective interface. Iterative design and experimentation has been suggested as a solution to this. This section reviews the design process offered by Gould, et al.,(1988). They suggest that the design of the interface can be divided-early focus on users, integrated design, early-and continual-user testing and, iterative design. Their design process has proven to be a worthwhile approach. Although this design process has much support some studies have demonstrated that it is often not applied by interface designers. This lack of acceptability is due to difficulties in implementing the design methodology to the development of interfaces.

Gould and Lewis (1985) argue that the quality of an interface is determined largely by the process used to design it and suggest four activities to guide the designer in this development process:

Early focus on users. It is imperative that the developer understands who the user will be. This can be accomplished by having direct contacts with the intended users. Interviews, observations, surveys and, participatory design are all ways to gain a better understanding of the user. The aim is for the designer to be familiar with the cognitive, behavioral, attitudinal, and anthropometric characteristics of the intended user. furthermore, an understanding of the nature of the work that will be accomplished must be acquired by the designer.

Integrated design. All aspects of interface development should be seen as a whole. Items such as the user interface, help system, training plan, and documentation must be developed in parallel and under one management. These aspects must not be developed sequentially.

Early and continual user testing. Early in the development process prototypes should be used to test the actual users doing the intended tasks This is based on the premise that the only feasible approach to successful design is an empirical one Measurements of the user's performance and reactions must be recorded and analyzed.

Iterative design. When problems are discovered during user testing they must be fixed. Consequently, the design must be iterative-there must be a cycle of design, user testing and measure, and redesign as often as possible The basic idea is that the system will be improved after each iteration.

There is much evidence to prove that this design process helps develop enjoyable interfaces that are easier to learn, easier to use and, provide an appropriate functionality Several researchers have proposed similar ideas (Bennett, 1984; Damodaran, et al., 1980; Meister, 1986; Reitman and Olson, 1985; Rubinstein and Hersh, 1984; Shackel, 1984; and Good, et al., 1986.). Good, et al.,(1986) have demonstrated how the use of such a methodology has lead to successful interface design at Digital Equipment Corporation. Many products that have been successful in the marketplace have been developed using this methodology (Xerox's Star system, 1982; Apple Computer, 1983, and Lotus 1-2-3, 1989). Finally, case studies have tested these ideas and concluded that this process is valuable (Bury, 1985; Good, et al., 1986; Gould, et al., 1983; Hewett, et al , 1986) Despite all this evidence, this design methodology is still not used widely

Gould, et al.,(1987) conducted a survey among interface designers to determine the suitability of their methodology to the development of interfaces The designers responded that the process was obvious and that everyone followed this type of methodology. They then asked the designer to name the design principles that should be applied and 26% could not even name one activity while 35% could only name one. This suggests that designers may not be properly equipped to apply this design methodology and that their orientation to design may be the reason that iterative design methodologies are not used. The authors state that four orientations to design can be applied: 1) Milestone oriented (e.g. project reviews), 2) Specification oriented (functional description), 3) Characteristic oriented and, 4) Process oriented (Gould's design principles). The orientation to design can more often be characterized as falling in on of the first three categories.

Why iterative design methodologies are not used. Gould, et al.,(1991) explained that there are several reasons why their design principles are often not applied. There are many designers who believe that usability cannot be measured. They think that only programmer productivity is measurable. This belief is wrong Several methods have been suggested to measure ease of use and ease of learning. For example, the GOMS model (described later) can measure ease of use, TAMPEL can measure ease of learning (Card, et al., 1983; Khalifa, 1991). These models are accurate enough to measure human factors attributes of an interface.

Another factor hindering inclusion of human factors principles in the development of the interface is the reward structure and apparent conflict between meeting deadlines and achieving usability factors. Project managers are reluctant to apply principles that may not provide tangible and measurable advantages. There is a belief that software development is not organized to carry out this process. Gould, et al.,(1991) state that designers feel that iterative design is too risky, too difficult, and too time consuming Often the interface code is intermixed with the functional code and a change at the interface may have an impact on the underlying code. With the development of methods to separate the interface code from that of the program, this problem will diminish over time. With methods to facilitate the use of HCI tools to interface development such as IDES (described in a later section), applying human factors methods may become feasible.

In addition to being expensive and time consuming the designers feel that they do not have the proper tool and training to carry out usability testing. User testing is expensive and complex. It often requires knowledge of psychological experimental techniques. Most interface designers have a formal background in systems analysis and design but not in experimental psychology. The user experiments often need the intervention of a human factors expert. This expert may not be available or may be too expensive for a small development project.

Even with the overwhelming evidence that including human factors is beneficial and even necessary, designers will be reluctant to work in that direction until some of the issues mentioned above are resolved. There has been much work to provide the interface designer with appropriate tools. The last section discussed design guidelines and design principles which can guide the designer and ensure that obvious design errors are avoided Design methodologies that integrates human factors in the development life cycle have been suggested (Grudin, 1988; Grudin, et al., 1987; Hartson and Hix, 1989, Mantei and Teorey, 1988). These methodologies may resolve the issue that the software design environment is not conducive to the inclusion of human factors Design guidelines, principles, and development methodologies are a step in the right direction

Two of the main reasons mentioned by designers as explanation of why human factors methods are not always included in the development effort are 1) it is too expansive and, 2) it is too difficult. Mantei, et al.,(1988) mention that eight distinct costs are added to a project by including human factor 1) The cost of running focus groups, 2) The cost of building product mockup, 3) The expense of the initial design of the prototype, 4) The expense of changing the prototype, 5) The expense of purchasing the prototyping software (UIMS), 6) The cost of running the users studies, 7) The cost of creating a user study environment (laboratory) and, 8) The cost of conducting user survey They provide an estimate of the total cost of adding the necessary activities to ensure the development of an appropriate interface Of this amount 50% is directly linked to prototype construction, user testing, and user evaluation. Furthermore, this assumes only four iterative changes which, in many cases, may not be enough This means that if there was a way to reduce the number of iteration and testing of prototype designers may be encouraged to include human factors in the design process

This leaves one difficulty, mainly that human-computer interaction is complex and the designers may not be able to apply the techniques available. Once a design prototype is completed there are many ways that it can be evaluated. The most appropriate and the one yielding the most accurate result is to perform actual user tests. As mentioned above, this is difficult and expensive. Other methods to evaluate an interface include Heuristic evaluation, Software Guidelines, Structured Walkthroughs, Decomposition, and Task-theoretic Analytic Models. Each of these evaluation methods have their pros and cons (this is discussed in the next section). The commonalty of these methods is that they are difficult to apply for the designer who is not appropriately trained With the development of easy to use evaluation techniques designers may be willing to include human factors in the development of their project.

Applying a proven interface evaluation technique on potential design is an approach which can reduce substantially the number of revisions to the prototypes that must be built. Paper-and-pencil interface could be evaluated by using one of the evaluation techniques mentioned above. Effective methodologies to integrate human factors issues in the traditional software life cycle is another approach that is necessary. Norman,(1986) suggests that there are four methods which can be applied in parallel to ensure the development of effective interfaces. 1) It must be impressed on the designer that the user has special needs and that the design must take this into account, 2) Quantitative methods (such as TAMPEL) must be provided to the designer to aid him/her in interface development, 3) Software tools for interface development must be provided and, 4) The interface design must be separate from other programming tasks Designers generally agree that the interface must be designed with the user in mind and UIMS tools provide mechanism which can help the interface development process and which can ensure that the interface code is separate from the function code. This leaves item number two which states that quantitative methods are necessary for the design of effective interfaces. This is addressed in the next section which reviews a hybrid design methodology for the development of interfaces. Evaluation techniques can be integrated into this hybrid methodology Appropriate evaluation techniques have the potential to resolve the issue of cost and difficulty of applying human factors to interface development.

D. Hybrid design methodology

The development of a human-computer interface is a complex undertaking. As mentioned in the previous section there may be difficulties in applying traditional software design methodologies For this reason several researchers have proposed hybrid methodologies to develop computer interfaces (Anderson and Olson, 1985; Guindon, 1990). These methodologies take into account the fact that the design of interfaces must be accompanied by certain activities such as user and task analysis, specification of an initial design, iterative revision of the design, development of a prototype and, iterative revision of the prototype This section provides a brief description of Anderson and Olson, (1985) design methodology An explanation of where these techniques can fit in this design methodology is provided

Traditional software development methodologies divide the development process into three general stages.

ANALYSIS. At this stage the product's functionality and initial hardware/software constraints are determined. A feasibility study is performed and a cost/benefit analysis of the system is proposed Finally, a development schedule is projected.

DESIGN. The product is actually designed at this stage First, only the functional specifications are proposed but later the complete details are coded and tested to terminate with a working system.

IMPLEMENTATION. The final product is installed in the actual work environment. All the appropriate training is performed with the users and with the support personnel.

The development of an interface can follow this general framework except that analytic models and empirical testing should be taken into account Anderson and Olson, (1985) refer to this as a hybrid design methodology. This methodology an be roughly divided into six distinct steps:

1. USER AND TASK ANALYSIS

This activity consists of determining the user's needs and capabilities. The functional details of the system that the interface must support are specified Details about the user and the tasks can be obtained through reports from users and natural observations. Questionnaires and interviews to understand the relevant details about the user can be administered, diaries of the users activities can be analyzed and, focus groups studies can be performed. This provides data which can be used analyze how the user performs a task.

2. DESIGN: THE INITIAL DESIGN

At this stage the initial interface is specified This design is based on the user and task analysis performed previously The interface or the system's details are derived from three sources- design guidelines or principles, the designer's intuition, or theory based judgment. Guidelines may be useful at this point to provide general guidance to the designer as to what makes a good interface. The same caveat mentioned previously should be remembered in that guidelines may be contextually constrained

3. DESIGN: FORMAL ANALYSIS OF THE INITIAL DESIGN

The goal of this stage is to carefully scrutinize the initial design and determine the possible improvements that could be made before the prototype is built. The purpose is to specify the best interface in order to minimize the number of iterations that will have to be done. This is a critical activity and the designer needs good support tools to improve the interface design. Paper-and-pencil interface design can be used at this stage.

4. DESIGN: BUILDING A PROTOTYPE

After the initial design has been evaluated and revised several times a prototype is built. Prototypes can be constructed with methods such as facading, Wizard of Oz and rapid prototyping (Anderson and Olson, 1985 provide a description of these methods).

5. PROTOTYPE TESTING AND ITERATIVE REVISION

At this stage the prototype is tested with the users. This user testing can be done through laboratory or field experiments. Data such as the time it takes to perform a task, the time it takes to learn a task, the frequency and kinds of errors, the goals and intentions of the user, and the attitude of the user can be collected. This data can be compared to human factors goals about the interface design that may have been agreed upon before the construction of the prototype. Revisions to the prototype are completed and further tests are performed until the design is deemed to be satisfactory.

6. IMPLEMENTATION

The final interface design is implemented and monitoring of the users is performed.

Other interface design methodologies do not include an additional activity of iterative revision of the initial design (Mantei and Teorey, 1988) The purpose of this extra step is to improve the initial design to the extent where the number of iterations of the prototype is be minimized. This is sensible considering the fact that prototype testing and building is a high cost activity Moving some of the revisions from stage 5 to stage 3 can shorten the development cycle and reduce cost substantially. Khalifa and Kira (1992) suggest two reasons for this:

1) It is less expensive and time consuming to formally analyze paper design then to perform actual user testing

2) It is much simpler to make revisions on a paper design then to make it on the actual prototype.

For these two reasons the hybrid design methodology presented be Anderson and Olson,(1985) is better then traditional software development approach. This hybrid design methodology takes into account two distinct features of interface design, mainly that the development must be iterative and that user testing must be performed. The following section provides a discussion on the different interface evaluation techniques.

IV. USER INTERFACE EVALUATION TECHNIQUES

Iterative design methodology can be considered a pre-requisite for the development of an effective interface Each prototype must be evaluated to determine the changes that must be made. This is done until a final, satisfactory interface is developed Several evaluation techniques have been suggested to analyze an interface design· heuristic evaluation, software guidelines, structured walkthroughs, decomposition, task-theoretic analytic models, and user testing. This section provides a description of each of these methods along with their strengths and weaknesses. An effective and easy to apply evaluation technique has the potential to reduce substantially the number of iterations necessary to develop an a satisfactory interface.

HEURISTIC EVALUATION. Nielsen and Molich, (1990) define heuristic evaluation as an informal method of usabiiity analysis where a number of evaluators are presented with an interface design and asked to comment on it. These User Interface experts rely heavily on past experiences to determine if there are any properties of the interface that may cause some usability problems.

ADVANTAGES. Nielsen and Molich (1990) state that there are four advantages to the use of heuristic evaluation (they assume that the designer performs the evaluation): 1) the method is inexpensive, 2) it is intuitive and it is easy to motivate people to do it, 3) it does not require advance planning, 4) it can be used early in the development process Jeffrey, et al.,(1991) conducted a study comparing evaluation techniques and found the heuristic evaluation to be adequate.

DISADVANTAGES. There are limitations to the heuristic evaluation method that makes it inappropriate to use. Nielsen and Molich (1990) conducted four experiments to test the effectiveness of this method. They found that at best the interface designer could identify an average of 34% of the potential problems. Furthermore, the method does not offer a framework to provide suggestions when problems are actually found. The evaluation is obviously biased by the mindset of the evaluator and this can be extremely limiting. The method is completely devoid of theory and based on subjective rules. It could not provide, for example, ways to improve the ease of learning and the ease of use. Interestingly enough Nielsen and Molich, (1990) suggest that this is the most popular method despite its major drawback-this may be due to its low cost.

SOFTWARE GUIDELINES. This technique makes use of published guidelines which provide evaluators with specific recommendation about the design of an interface Examples of such guidelines are how the content of a screen should be organized or how items should be arranged in a menu (Polson, et al , 1990; Smith and Mosier, 1986, Shneiderman, 1986).

ADVANTAGES. There are several advantages to the use of software guidelines to evaluate an interface. First, the method is inexpensive and does not requir. UI experts. The interface designer only needs a document which provide general guidelines that can be applied across a wide range of computer and instrument systems. Several guideline books have been published in the last few years (Apple Computer Inc , 1989; HP Guidelines, 1990; Smith and Mosier, 1986). Jeffries, et al., (1991) conducted a study to determine the strengths and weaknesses of each evaluation techniques and found that a strength of software guideline evaluation techniques is that the evaluator is able to identify serious and recurring problems.

DISADVANTAGES. Jeffries, et al., (1991) also found several problems with this technique. In their study the evaluators using guidelines were only able to identify 35% of the problems present in the interface design. Furthermore, the guideline books were cumbersome to use and did not address all the issues of interface evaluation. As mentioned previously guidelines are not based on sound theoretical foundation. Guidelines are developed through empirical testing and consequently cannot address all the issues in the interface development Again, this technique is solely a descriptive techniques it does not provide ways to correct the problems with the interface This makes it inadequate as a stand-alone method.

USER TESTING. In usability testing the evaluator studies the interface under real world or controlled situations. The evaluators gather data on problems that arise during the use of the interface.

ADVANTAGES. Jeffries, et al., (1991) found that most serious problems were found using this method. The technique provides an opportunity to test the users in a real-life work environment. Given that the interface is studied properly, this method ensures that most usability problems are found .

DISADVANTAGES. There are several reasons why usability testing may be the least popular evaluation method First, UI expertise is necessary. Second, it is very expensive (refer to the last section for cost estimates). The cost of setting up a laboratory and the expense of building a prototype to carry out the testing ensures that this method is prohibitive for many software projects. Finally, this is a descriptive technique and it does not provide ways to improve the design.

STRUCTURED WALKTHROUGH. Yourdon (1989) suggests the use of structured walkthroughs to evaluate software This technique was adapted by Lewis, et al., (1990) to include a theoretically structured process called a Cognitive Walkthrough. Structured Walkthroughs involve the construction of tasks that the user actually carries out on a simulated system. The user tries out the system by going through the task, step by step, screen by screen, command by command. This can be done through experimental simulation with the aid of a prototype or through the evaluation of paper and pencil design. The purpose of a structured walkthrough is to identify confusing, unclear, or incomplete instructions, illogical or inefficient operations, unnatural or difficult procedures, and procedural steps that may have been overlooked in the design of the interface (Anderson and Olson, 1985).

Structured Walkthroughs are inherently atheoritical in that they only provide a method to go through a software This makes the technique less useful for the evaluation of an interface. For this reason the Cognitive Walkthrough was suggested be Lewis, et al., (1990). The Cognitive Walkthrough is a theoretically structured evaluation process that takes the form of a list of questions The questions are derived from the Cognitive Complexity model developed by Polson and Kieras, (1985) and they focus the designer's attertion on individual aspects of the interface that the underlying theory claims are important in facilitating the problem solving and learning processes of an interface.

ADVANTAGES. Cognitive Walkthroughs have several strengths that make it a desirable evaluation techniques for certain cases. First, the technique helps the evaluator define the user's goals and assumptions. This helps the designer understand why certain features of the interface may cause problems and may help determine how a better interface could be designed Second, the method is based on theory and is not contextually constrained. Third, this technique can be used by software developers.

DISADVANTAGES The evaluator must be familiar with a task definition methodology and this is difficult to learn. Testing an interface with this method can be lengthy and tedious. It often requires pages of analysis Jeffries, et al , (1991) state that the evaluators using the Cognitive Walkthrough technique were not able to identify general and recurring problems. Rieman, et al , (1991) proposed an automated version of the Cognitive Walkthrough This makes the method less tedious to use Lewis, et al , (1990) do not claim that the cognitive walkthrough methodology will eliminate the need for evaluating prototypes of the interface Their argument is that the walkthrough, with a very limited investment in resources, approximately one hour per task per interface, can detect almost 50% of the problems encountered by users of the design Jeffries et al , (1991) found that only 35% of the problems were identified. This may not be high enough to replace usability testing as an evaluation technique

DECOMPOSITION. This technique is proposed by Reitman, et al , (1985) It consists of separating the major components of the interface design and analyzing these components for their impact on cognition. A picture displayed on a screen, for example, is evaluated by determining the user's ability to perceive meaningful relationships of the system model. The commands, for example, are assessed by evaluating their load on long term memory, how easy they are to remember, and how confusable they are among each other Reitman, et al., (1985) suggest that once the first evaluation has been performed a second design alternative should be constructed and analyzed again. They also propose that each alternatives should be debated among the design team to determine the best approaches for each components. This technique can be performed on paper-and-pencil before a prototype is constructed.

ADVANTAGES. The method encourages careful scrutiny of the proposed design and also encourages designers to specify better interfaces before the first prototype is built. This makes decomposition inexpensive to perform.

DISADVANTAGES. Some of the drawbacks are that UI expertise is required Reitman, et al., (1985) state that the component should be evaluated based on theoretical interface guidelines and principles. They do not mention that guidelines and principles are often not adequate to evaluate a design. If task-theoretic analytic methods are used their complexity can be a problem .

TASK-THEORETIC ANALYTIC MODELS. These techniques are a promising approach to interface design evaluation The models provide thc evaluator with representations and analysis that assess different aspect of an interface. Some models evaluate, for example, which part of a metaphor aid performance and which do not (Douglas and Moran, 1983) or how the user's short term memory load is affected at each stage of performing a task (Kieras and Polson, 1985) According to Anderson and Olson. (1985) examples of these models are metaphor analysis (Carrol and Thomas, 1982, Carrol and Mack, 1982), assessment of mental models (DeKleer and Brown, 1983), development of production rule systems that represent the user's knowledge of the task (Kieras and Polson,1985, Khalifa, 1991), object/action analysis (called "external/external task mapping" by Moran, 1983), the GOMS model (Card, et al , 1980, 1983), and formal grammar notation (Reisner, 1981)

ADVANTAGES All of these models have the advantage of being based on sound theory of human behavior. They can be applied before any coding is performed and help the designer structure the development to include principles that are based on theories of human behavior Furthermore, these tools provide quantitative evaluations For example, ease of use of an interface can be measured by applying the GOMS model. This enables the evaluator to compare two design alternatives quantitatively It is hoped that analytic tools will develop to the point where the designer will include all the theoretical knowledge of the user behavior in the interface design This could render actual user testing unnecessary The goal is to have user testing as the final check before the interface is actually built. Although there has been much progress with this approach understanding of the user behavior has not reached a level where it is possible to eliminate prototype testing.

DISADVANTAGES. There are several reasons why Task-theoretic analytic models are often not used in the evaluation of an interface First, none of the models encompass all of the cognitive aspects of human-computer interaction; each focuses on one important aspect For example, the GOMS model permits the evaluator to test the ease of use of an expert user performing a specific task. Second, these methods require the understanding of complex psychological theories which is often beyond the designer's expertise. Third, the intervention of a human factors expert is necessary. The difficulty in applying these models have led Kira and Khalifa (1992) to develop an automated tool (IDES) which helps the designer in measuring ease of use (Keystroke-Level Model) and ease of learning (TAMPEL) of an interface.

Although there are severa problems with applying task-theoretic analytic models to evaluate an interface it is still the most promising approach since it is based on sound theory. The purpose of this paper is to demonstrate that models such as the Keystroke-level Model to measure ease of use and TAMPEL to measure the ease of learning can be effectively and efficiently used with tools such as IDES which was developed by Khalifa and Kira (1992). According to Khalifa and Kira (1992) there are several ways that task-theoretic analytic models can help the designer develop better interfaces

1. The use of formal models requires the designer to describe formally the interface specifications. This ensures that the designer is precise and consciously thinks of human factor issues

2. Formal models helps the designer formulate clear testable hypothesis concerning the design alternatives This is very helpful in that determining what to test is often a very difficult activity Such models can provide helpful guidance.

3. Formal models can helps the designer learn about what design properties make for effective interface. Formal models, when used repeatedly, can help the designer develop heuristics of good interface design.

It is important to note that the best approach to designing and evaluating an interface may involve the combination of several techniques. Guidelines and principles can provide guidance to ensure that gross errors are not committed at the initial interface design. Heuristic evaluation, based on user behavior principles may then be used to evaluate an interface roughly. Task-theoretic Analytic tools can measure quantitatively parameters such as ease of use and ease of learning for example The designer should make use of guidelines, principles, different evaluation techniques, and iterative design methodologies to ensure the development of an effective interface

V. DESCRIPTION OF COGNITIVE MODELS

Khalifa and Kira, (1992)'s IDES incorporates Task-theoretic Analytic Models based on the Goals, Operators, Methods, and Selection rules (GOMS) model of Card, et al., (1983) formalism to describe interfaces. Several predictive techniques such as the Keystroke-Level Model and TAMPEL are based on the GOMS formalism. These models provide the designer with the ability to predict how users will interact with a proposed interface and derive quantitative predictions of, for examples, human factor goals such as ease of learning and ease of use. The following sections describe these models and reviews some of their limitations

A. The GOMS Model

The framework suggested by Card, et al., (1983) has two components. The first one provides a general characteristic of the human information processing system, in terms of both a system's architecture and quantitative parameters of components performance. This is called the Model Human Processor (MHP). The second component (GOMS) provides a method to describe what the user needs to know in order to perform computer-based tasks. Furthermore, the GOMS model describes the four cognitive components of skilled performance in their tasks: goals, operators, methods, and selection rules.

The MHP model was described in the section on design principles. From this model the authors are able to derive quantitative values for the cognitive processes that are involved in human-computer interaction. Time parameters from these processes are estimated from empirical data obtained from people using text-editors and graphics systems (Card, et al, (1983)). There are four activities that represent the human interaction with the computer· keystroke (k), mental operator (M), pointing (P), and moving the hand (H). A keystroke consists of the time it takes for an average typist to press each key. A keystroke is estimated at .28 seconds. A mental operator is interpreted as the time to retrieve the next chunk of information from long-term memory into working memory and is estimated to be 1.35 seconds. Pointing refers to pointing to a target on a small display with a mouse and is estimated to take on average 1.1 seconds (the time is variable according to Fitts's law) The time to move the hand from the keyboard to the mouse is estimated to be .400 seconds.

The GOMS Model consists of four cognitive components 1) goals and subgoals for the task; 2) operators, including both overt operators (like key presses) and internal operators (like memory retrievals), 3) methods composed of a series of operators for achieving the goals; and 4) selection rules for choosing among competing methods to achieve the same goals. Card, et al., (1983) suggest that the GOMS Model can be formulated at several levels of analysis: the Unit-Task Level, the Functional Level, the Argument Level, and the Keystroke Level. Each of these levels represent a different grain of analysis A GOMS analysis is a formal and structured way to describe the knowledge that a user needs to perform a specific task on a device Describing the goals, operators, methods, and selection rules in a formal way is the goal of performing a GOMS analysis Once the GOMS model has been developed predictions about human factor parameters such as ease of learning and ease of use can be derived

Olson and Olson (1990) state that there are several reasons why cognitive models are useful for the interface designer First, such models constrain the design space These constraints can, for example, ensure that the interface is not designed in such a way that the working memory is overloaded Second, cognitive models can help answer specific design decisions so that the designer can decide between having an interface that requires few keystrokes but is difficult to remember or one that involves more keystrokes but is easy to recall. Third, the time to learn a system or the time it takes to perform a specific task can be estimated and the staffing requirements can consequently be derived Fourth, the building of the training material can be based on such cognitive models Finally, the understanding of the human-computer interface can lead the designer towards interface features that have strong implications for performance

Often, the design team sets certain human factor goals that must be achieved through the interface. The designers might state that a specific task should take one minute to complete and five minutes to learn. Before the introduction of Task-theoretic Analytic Models the only way to determine if the human factor goals of ease of learning or ease of use had been met was to built a prototype and perform user testing As mentioned in the section on evaluation techniques user testing is difficult and expensive Cognitive models can predict some parameters of the system and can avoid user testing at the beginning of development. Several authors (Card, et al., 1983, Reisner, 1981, Anderson and Olson, 1985, Kieras and Polson, 1985; Bovair, et al, 1990, Khalifa, 1991) propose the use of GOMS analysis to evaluate and improve the design of an interface on paper before performing actual user testing. There is strong support for the use of Cognitive Models in the development of effective interfaces.

The predictive accuracy of a GOMS analysis is supported by several authors (Card, et al., 1983; Olson and Nilsen, 1988; John and Newell, 1989; Walker, et al., 1988). These researchers performed GOMS analysis on interfaces and then conducted user testing experiments to determine the accuracy of the model. These studies provide evidence that GOMS analysis is accurate enough to provide reliable measures.

Although there is strong support for the use of GOMS analysis in the design of interface there are still some limitations to the model itself. The section on evaluation methods discussed the drawbacks to the use of Task-theoretic Analytic Models without alluding to the areas where GOMS analysis is not appropriate. Card, et al., (1980) propose a detailed list of some of the shortcomings of the model:

1. The model only applies to skilled users, not to beginners or intermediates. Non-expert users spend much time in problem solving activities. The original model assumes that users simply retrieve and execute plans automatically.

2. The original GOMS does not provide an account of ease learning or of recall after a period of time

3. The model focuses on errorless performance of the task. No account of errors that may take place even during skilled performance is provided.

4. The model is explicit about perceptual and motor components of skilled behavior but is not explicit about the actual cognitive processes that takes place during interaction.

5. The model is adequate to analyze tasks that can be assumed to be serial in nature while there are many tasks that occur in parallel at some level.

6. The model does not assess the functionality of a task but concentrates solely on the usability of the interface.

7. The model does not address how much must be kept in mind while using the system- mental workload.

8. The amount and kind of fatigue that a user may experience while using the system is not part of the model.

9. Individual differences among users is not accounted for. This stems from the fact that the system was developed to only take account of skilled users.

10. The model does not provide support to determine the satisfaction or acceptance of an interface.

These limitations are well documented and for these reasons the GOMS Model has been extended to be less restrictive. The original model is well suited to predict the performance of a skilled user carrying out routine tasks. Polson and Kieras, (1985) extend the model to include production rules and propose the Cognitive Complexity Theory to account for learning of an interface. Khalifa, (1991) suggest a model to measure ease of learning and provides several useful extensions. Smelcer, (1989) extend GOMS analysis to predict the number of errors that occurs with a given interface. Kieras and Polson, (1985) suggest that the number of production rules in there CCT approach can help determine where errors will occur. Khalifa and Kiras (1992)'s IDES includes the Keystroke Level Model and TAMPEL to assess both the ease of use and ease of learning. The following section provides a brief description the Keystroke-Level Model and TAMPEL.

B. Keystroke-Level Model

The Keystroke-Level Model provides the interface designer with the capabilities to predict how much time a user would take to accomplish a given task with a given interactive computer system. Given the task (may involve several subtasks), the command language of a system, the motor skill parameters of a user, the response time parameters of the system, and the method used for the task, the model can predict the time an expert user will take to execute a task using the system providing the method is used without error.

The Keystroke Level Model is different from the GOMS in that the methods are not predicted by this model- the assumption is that the user is an expert and uses the same method consistently. What is provided is a set of heuristics which determine where the mental operations occur. In this model each task is assumed to consist of a set of keystrokes, hand movements, and mental retrievals. The same parameters as the GOMS model are used: 1) Keystroke (K)= .280 seconds; 2) Mental operator (M)= 1 35 seconds; 3) Pointing with a mouse (P)= 1.1 seconds; and 4) Moving hand from keyboard to mouse (H)= .400 seconds. An analyst can use these parameters to predict the time it will take for a task to be completed given heuristics of where to put the M's and when to insert the amount of time the system takes to respond (with the advent of inexpensive processing power this is becoming negligible) A prediction of adding the contents of cell B22 to B29 could be done in the following manner:

@ s u m (12345678 : 654321)<ret>
MK MK K K MK MKKKKKKKK MK MKKKKKK MK MK

Which ads up to:

8 M's and 22 K's

8(1.35) + 22(.280)= 16.96 seconds

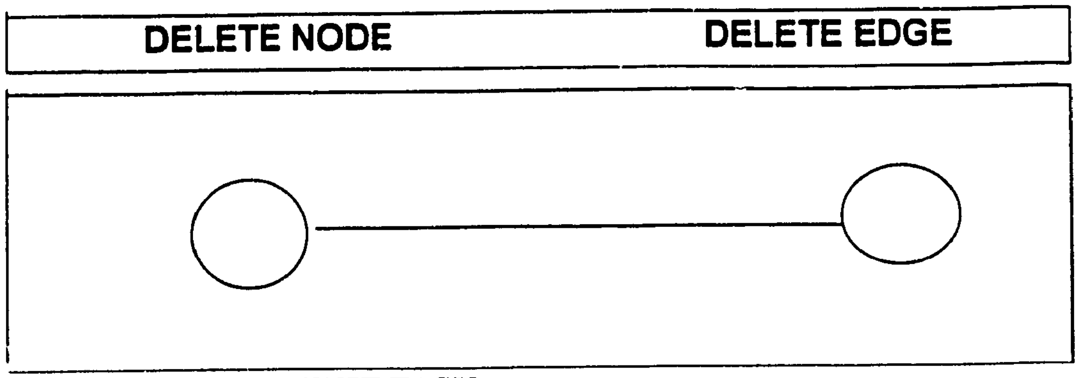
By calculating the parameters and applying consistent M's using formal heuristics this model can predict the time it takes to perform a routine task without errors. Card, et al., (1983) were able to calculate unit task time from a large set of tasks with a 90% accuracy.

C. TAMPEL

Khalifa, (1991) proposed a model to evaluate an interface in terms of ease of learning. This approach is the most comprehensive so far Khalifa (1991) calls his methodology TAMPEL (Task Analytic Methodology for Predicting Ease of Learning) TAMPEL is different from the other models proposed for ease of learning in that it identifies different classes of rules and assumes that there are different types of cognitive skills that have different learning times. The methodology takes into account the way that these skills are retrieved from long-term memory Skills that can be recognized (recognition) from the interface are easier to learn then those that must be recalled TAMPEL measures ease of learning differently Previous studies assumed that ease of learning could be measured with training time This may not be a reasonable assumption and for this reason Khalifa, (1991) suggests that cognitive learning time can be empirically measured by the difference between the total time spend learning how to perform a routine task efficiently (total learning time) and the time spent on performing the same tasks after learning (usage time) Furthermore his study allows the subjects to follow their preferred learning method (previous study restrained the methods that the subjects could use)

Khalifa (1991) performed an experiment where users were asked to complete certain text-editing tasks of a computer system. Accurate measurements were taken of the time to perform each tasks. Based on these detailed measurements a regression analysis was performed to develop a predictive model Khalifa evaluated the cognitive learning time to be based on the number of recalled procedures and recognized procedures that are necessary to complete a particular task. Khalifa (1991) estimated that the total learning time can be evaluated with the following equation **Learning Time = -58.15 + 43.63 (recall) + 25.53 (recognition) + 5.99 (additional rules)** The number of additional rules is estimated to be the rule recalls minus the procedure recall plus the procedure recognized.

Empirical experiments support the TAMPEL model (Khalifa,1991) The methodology accurately predicts the ease of learning of 54 text-editing tasks and accounts for more the 77% of the variance among the observed learning times The experiments performed are rigorous and statistically significant. The study asked participants to learn how to use a specific interface TAMPEL is the most complete Task-theoretic Analytic model to measure ease of learning and for this reason this approach is part of the automated tool that is used in this study. The following is an example of using TAMPEL measure ease of learning. This system deletes nodes and edges. The task is to delete two nodes and one edge.



```
GOAL:  ACQUIRE-UNIT TASK
IF (NODES TO BE DELETED)  THEN                       RULE_RECL
   UNIT-TASK = DELETE-NODE                           PROC_REGZ
ELSE IF  (EDGE TO BE DELETED) THEN                   RULE_REC
   UNIT TASK = DELETE-EDGE                           PROC_REGZ

GOAL:  EXECUTE-DELETE NODE
IF ("DELETE-NODE" IS NOT SELECTED)  THEN             RULE_RECL
   UNIT-TASK =  SELECT ("DELETE -NODE")              PROC_RECL
ELSE IF  ("DELETE-NODE" IS SELECTED)  THEN           RULE_RECL
   UNIT TASK = SELECT (LOCATION)                     PROC_RECL
GOAL:  EXECUTE-DELETE EDGE
IF ("DELETE-EDGE" IS NOT SELECTED)  THEN             RULE_RECL
   UNIT-TASK = SELECT ("DELETE-EDGE")                TRANSFER
ELSE IF  ("DELETE-EDGE" IS SELECTED) THEN            RULE_RECL
   UNIT TASK = SELECT (EDGE)                         PROC_RECL

GOAL:  EXECUTE-SELECT (OBJECT)
IF (CURSOR IS NOT POINTING TO OBJECT)  THEN          RULE_RECL
   UNIT-TASK =  POINT-TO (OBJECT)                    PROC_RECL
ELSE IF (CURSOR IS POINTING TO OBJECT) THEN          RULE_RECL
   UNIT TASK = PRESS (ANY BUTTON)                    PROC_RECL
```

RULE_RECL = 8

PROC_RECL = 5

PROC_REGZ = 2

ADD _RULES = 8 + (5 - 2) = 11

CLT = -58.15 + 43.63(5) + 25.53(2) + 5.99(1) = 217.05

TIME TO LEARN = 276.95 SECONDS

Figure 2: TAMPEL example

Explanation of example.

Goal: acquire unit task The first action that the user must complete to perform the task is to "acquire the unit task". This effectivelly means that the user must determine what he/she wants to do. In the case of our example the user must remember the condition for applying the procedure "delete node" or "delete edge" Knowing the condition for applying a procedure is considered a rule recall In the interface depicted the functionality of the procedure is displayed on the screen. For this reason we consider the procedure of "delete node" and "delete edge" to be recognition procedure The user needs less cognitive effort since the functionality of the system is displayed on the screen-the user can recognize the procedures.

Goal: execute delete node. In order to perform the task the user must actually choose the necessary procedure. The user must know the condition for applying the following operator. In our example the user must know if the function "delete node" has been selected or not. This is a rule recall Once the decision is made if the function has been selected or not the user must know what action to take in order to perform the procedure. In the case of our example there are no memory aides that are displayed on the screen. For this reason the "unit task select-delete node" is considered a procedure recall. After "delete node" is selected the user must select the object to delete. In the case of our example the first object to delete is a node

Goal: execute delete edge. Once the node is deleted the next step is to learn how to delete an edge. Again, the user must know the condition for applying the "delete edge" function. Again, this is a rule recall. Once the decision is made that the function has been selected or not the user must know the appropriate action to take in order to perform the goal The user must now select the delete edge function This action has already been performed (select delete node). For this reason this selection is considered a transfer Once the function is selected the user must acutally perform the action. In this case the user has to delete and edge. This is the first time that an edge is deleted-for this reason this action is a procedure recall. It is not a procedure recognition since there are not visual cues that can help the user in his/her selection.

Goal: execute select object. In this example there are two instances when the user selects an object-node and edge. The action of selecting these nodes and edges must be learned. First the user must determine if the cursor points to the object. This is a rule recall. Then the user must actually perform the action of pointing to the object. This is considered a procedure recall If the cursor is pointing to the object then the user must perform the action is pressing any button to activate the full function. This is a procedure recall.

Predicting ease of learning. Once the interface has been represented using the TAMPEL formalism one must use the equation derived by Khalifa (1991) to predict the ease of learning of such an interface The first step is to count all the rule recall, procedure recall, procedure recognized, and calculate the additional rules. Once these values are obtained they must be used in the regression equation that predicts ease of learning. In our example the time to learn the interface isestimated to be approximatelly 276.95 seconds.

The Keystroke Level Model and TAMPEL are GOMS based models that enable the interface designer to formally predict ease of use and ease of learning. Although there are many limitations (some were mentioned in the last section) to GOMS analysis there are distinct advantages that make them useful in interface design. GOMS analysis can formally evaluate interfaces before the prototype is built. The empirical tests demonstrate that these models are accurate enough to be reliable in paper-and-pencil evaluation. In this way the interface can be iteratively improved until the interface is effective enough for a prototype to be built. IDES enables the application of these formal models by the designer directly without the help of human factors professional who are not always available The next section provide a description of Khalifa and Kira (1992) 's IDES.

VI. DESCRIPTION OF GTAL METHOD TO REPRESENT INTERFACES

As mentioned previously the use of Task-theoretic Analytic models are helpful in the evaluation of interfaces but that unfortunately their complexity has hindered their acceptance among developers. For this reason Khalifa and Kira, (1992) have developed an automated tool that designers can use to describe an interface design and evaluate these designs in terms of ease of use and ease of learning. Their tool consists of an Interface Description and Evaluation System (IDES) that performs the following functions: 1) create a formal description of user interfaces, 2) delete, add, and modify parts of an interface description, 3) query an interface description, and 4) evaluate an interface description in terms of human factors A designer can thus apply complex psychological theory to evaluate an interface in terms of ease of use and ease of learning.

Khalifa and Kira, (1992) describe their IDES in terms of a Data Base Management System (DBMS). There are two important differences between their IDES and a DBMS: 1) an IDES manages formal descriptions of user interfaces as opposed to regular d .a, 2) the relationship between differen. interface description is represented with the use of different formalism. An IDES has four components: an Interface Description System (IDS), an Interface Description Manipulation System (IDMS), a Query System (QS), and an Evaluation System (ES). These components allows a user of the IDES to create, edit, query, evaluate, and store interface descriptions in an Interface Description Base (IDB)

The following description of these components is based on Khalifa and Kira (1992):

1. Interface Description System (IDS)

The IDS is a tool that describes the interface in terms of user actions. This description is translated into a formal description using one of the formalism that the IDES supports. There are many formalism that have been proposed to describe user interfaces (Command Language Grammar by Moran, 1981; GOMS formalism by Card, et al., 1980; Task Action Grammar by Payne and Green, 1986 and; Formal Grammar by Reisner, 1981). These formalism are often difficult and time consuming and for this reason there are not easily adopted by interface designers. For this reason the automated tool uses a graphic representation of the interface that is based on GOMS. This is done to simplify the formal representation of the interfaces. These descriptions are stored in the IDB.

2. Interface Description Manipulation System (IDMS)

The IDMS is a set of procedures that enables the user to add, delete, or modify the interface description stored in the IDB.

3. Query System (QS)

This component consists of a set of user friendly commands that allows the user to retrieve the formal description of the interfaces stored in the IDB.

4. Evaluation System (ES)

The ES is where the formal Task-theoretic Analytic Models are stored. With these models the user can formally evaluate interfaces. IDES uses two different models to predict both ease of use and ease of learning. The Keystroke Level Model (Card, et al., 1983) is used to predict the ease of use and TAMPEL (Khalifa, 1991) predicts ease of learning. This tool has the potential to enable the interface designer to easily apply complex psychological theory to the development of interfaces. As mentioned previously one of the problems with using models such as TAMPEL is that the representational formalism is quite complex. For this reason Khalifa (1991) has developed a graphical representational formalism. The following provides an example of how the ease of learning of a task can be measured.

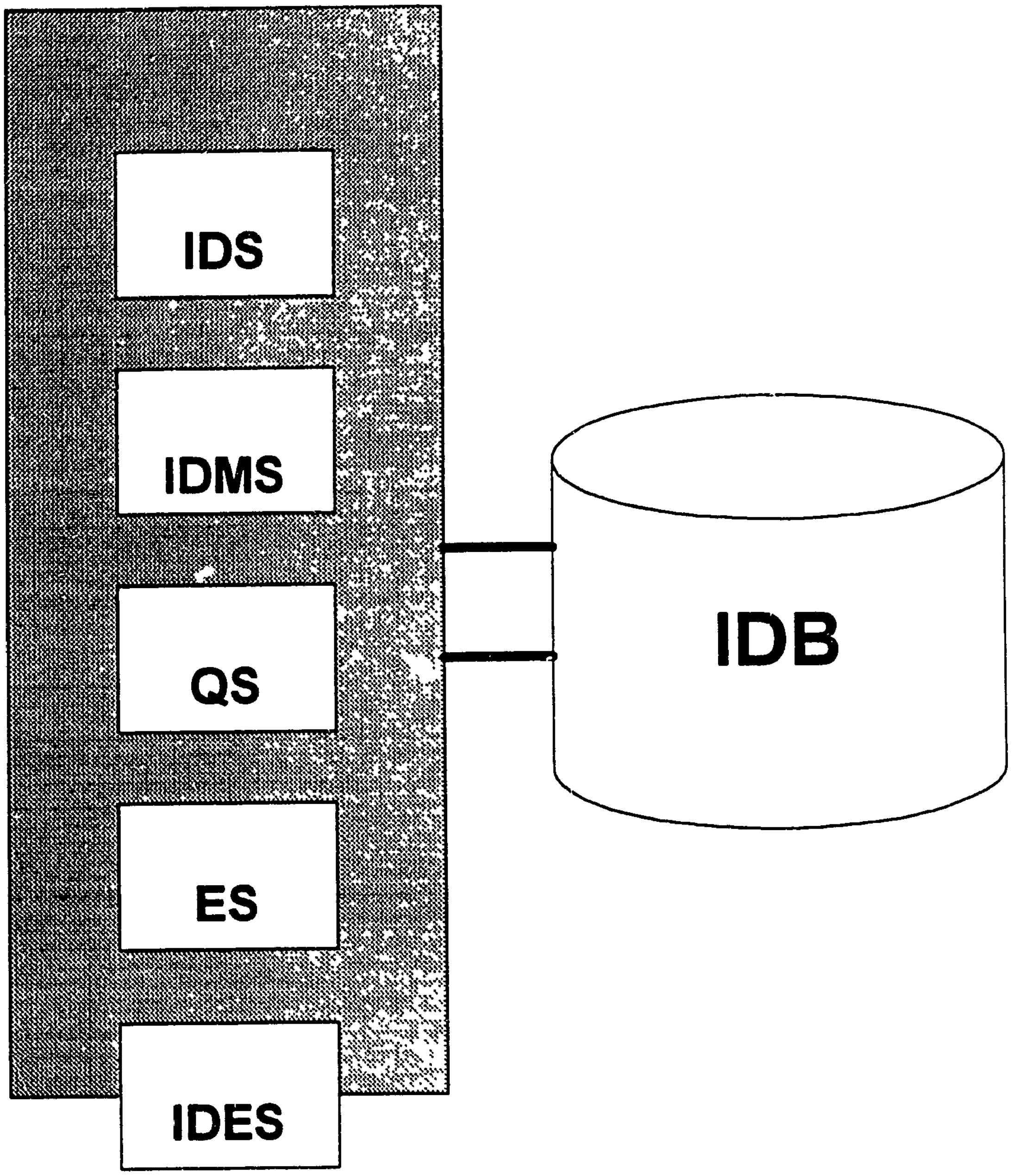


Figure 3: Component of IDES

EXAMPLE OF USING IDES

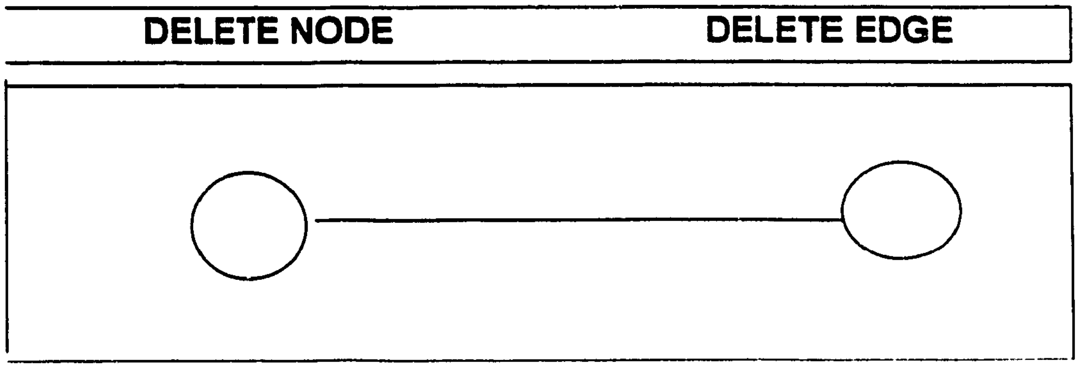


Figure 4: Example of IDES

The section on the description of TAMPEL provided the method to measure the ease of learning manually. Solving the same problem using IDES is quite different A graphical representation is first completed and the system consequently calculates the ease of learning automatically. At this stage of the tool's development the users are able to present only part of an interface at at time. Consequently, the system only calculate the ease of use and ease of learning of a particular task of the interface. The user has to add all the results of the "subtasks" in order to evaluate the whole interface In the future the whole

interface will be represented with the system.

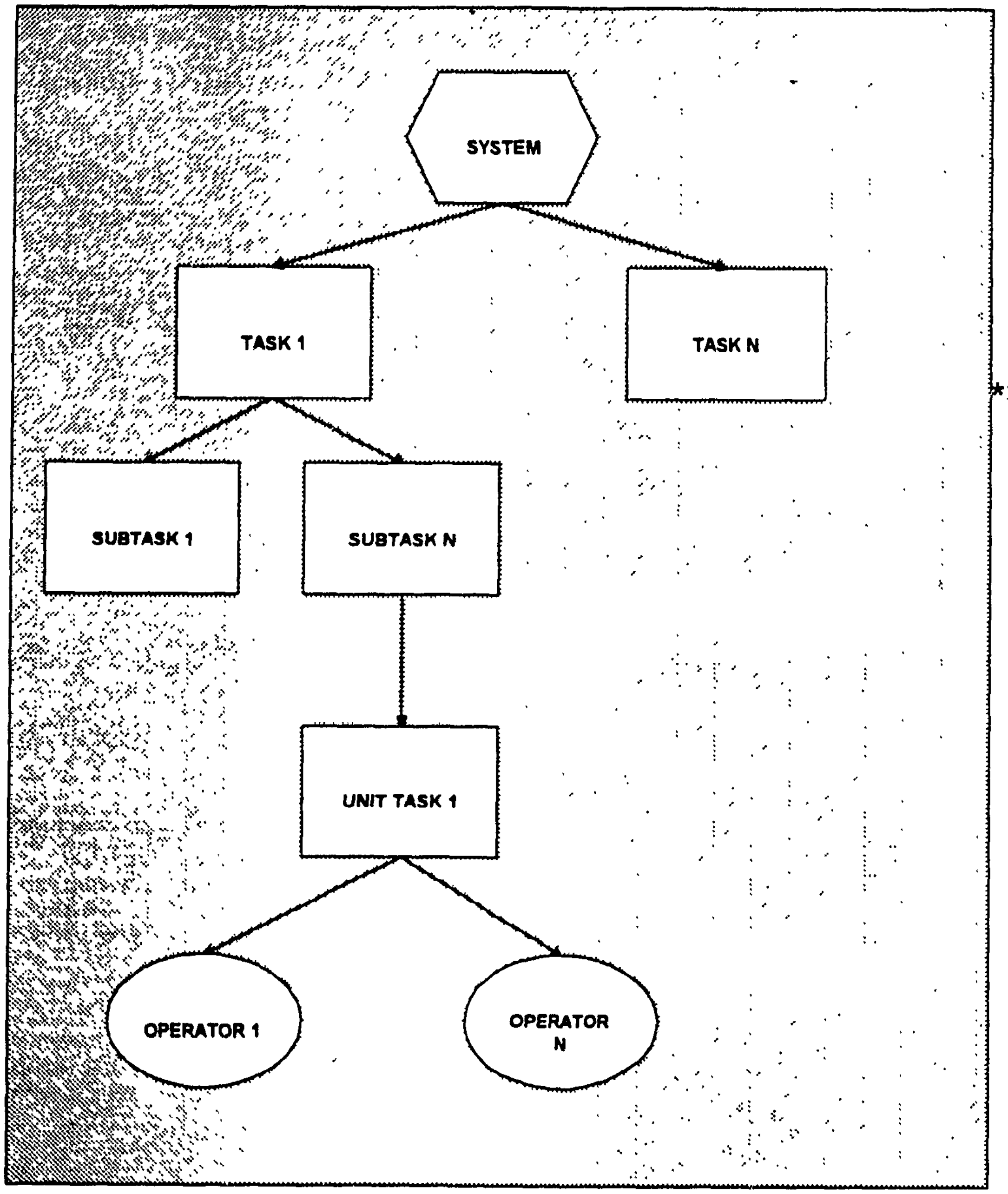


Figure 5: GTAL Description of the Task Hierarchy

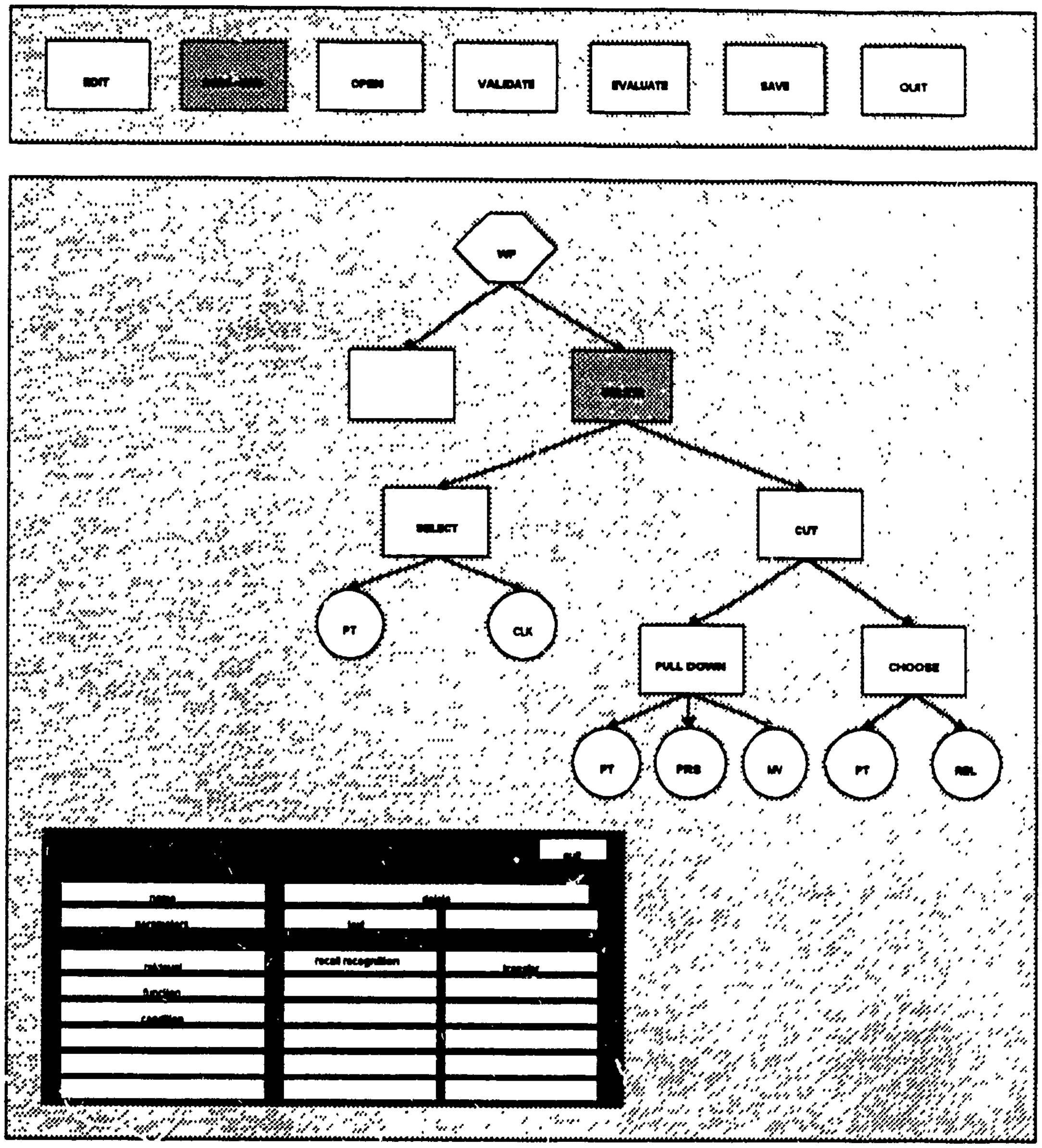
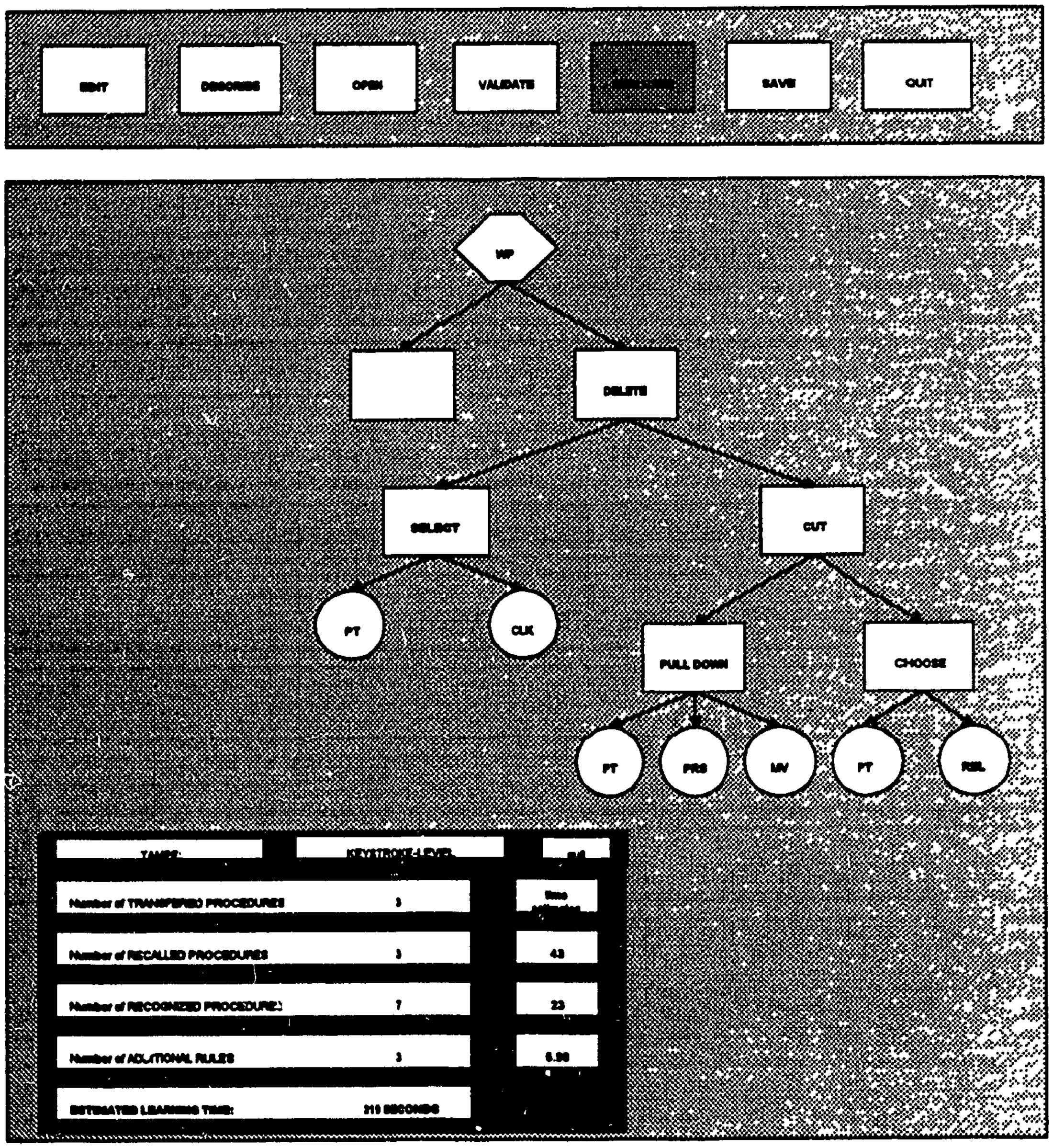


Figure 6: GTAL Representation of a Task



TAMPE:	KEYSTROKE-LEVEL	[illegible]
Number of TRANSFERRED PROCEDURES	3	time estimates
Number of RECALLED PROCEDURES	3	43
Number of RECOGNIZED PROCEDURES	7	23
Number of ADDITIONAL RULES	3	6.30
ESTIMATED LEARNING TIME:	210 SECONDS	

Figure 7: Evaluation of a Task

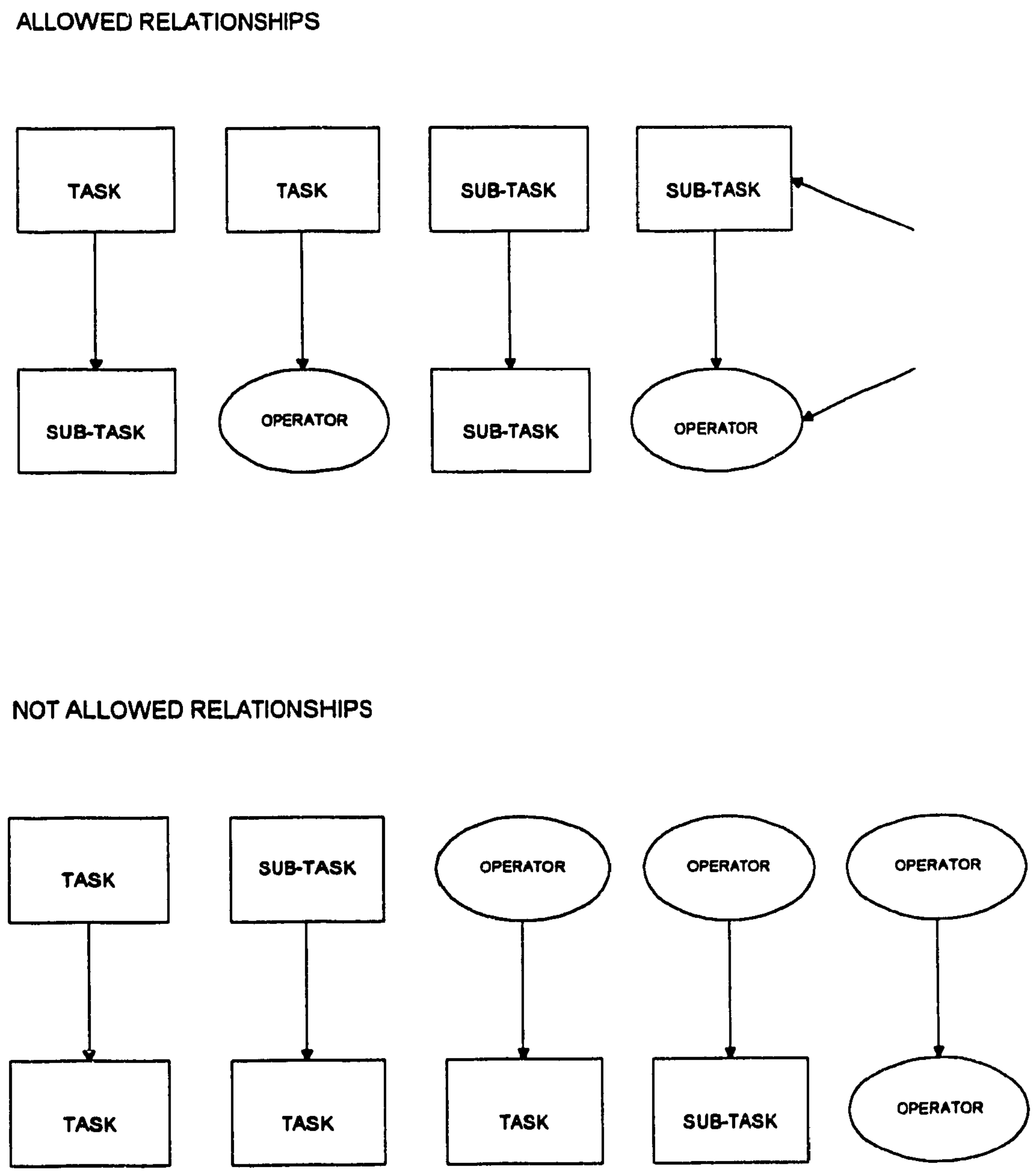


Figure 8: Possible Relationships

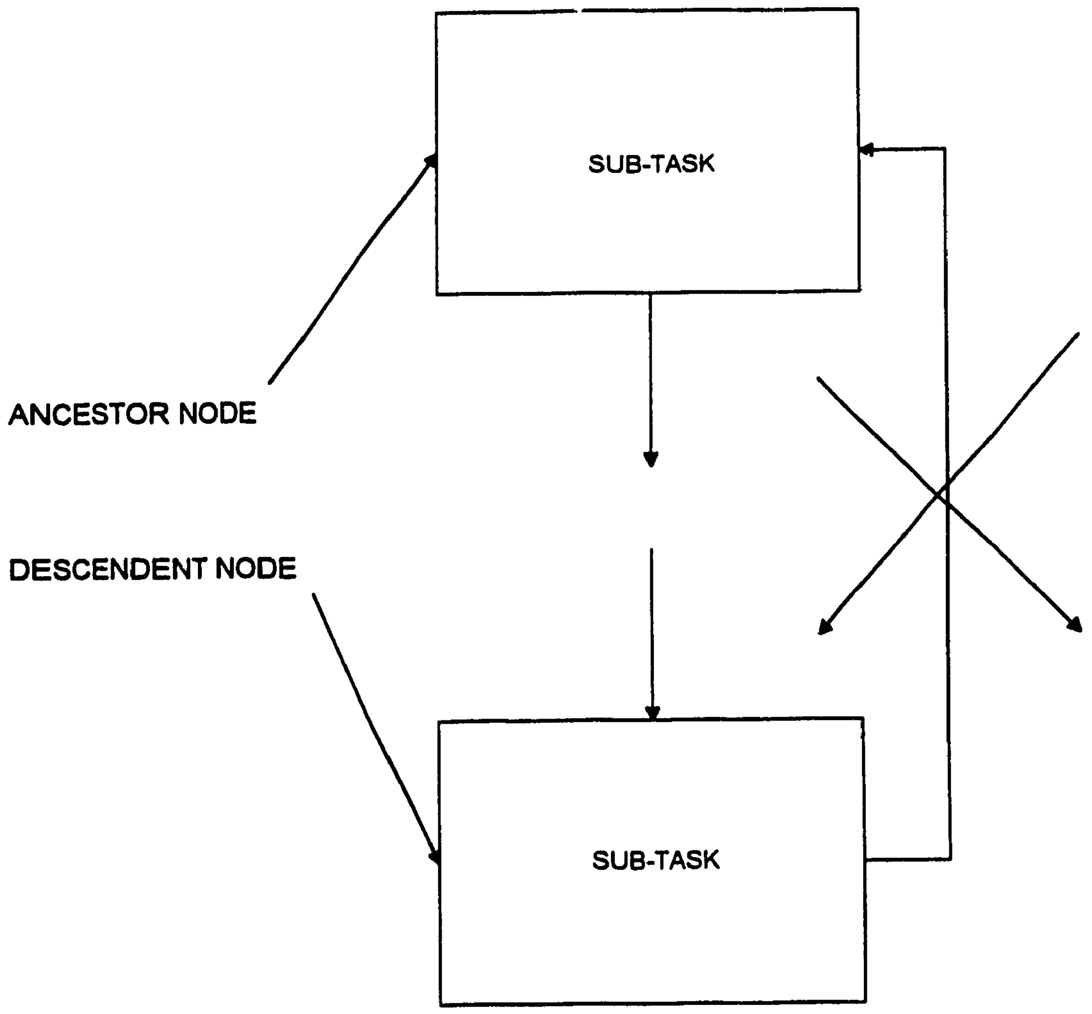
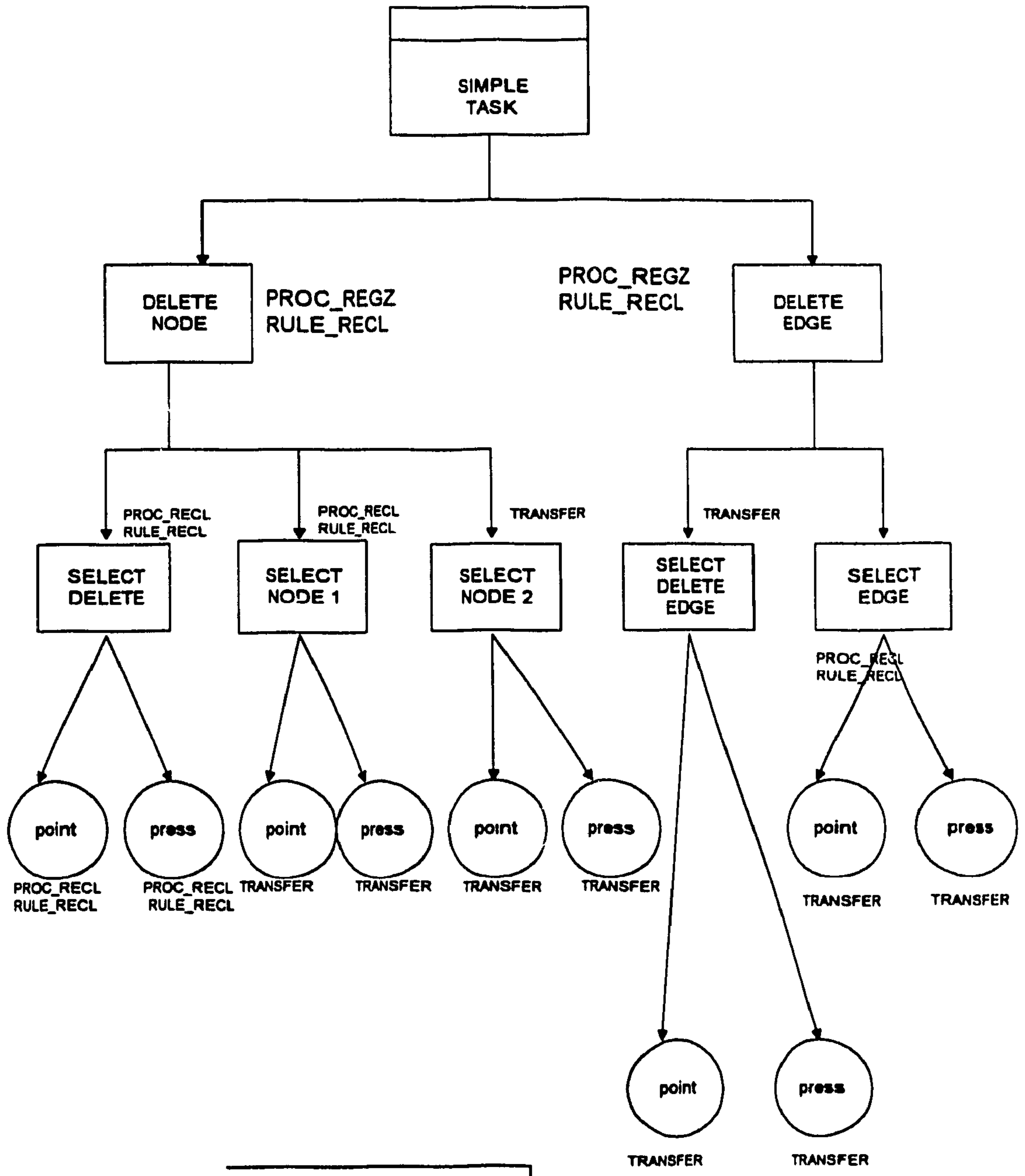


Figure 9: Non-recursion Rule



RULE_RECL = 8

PROC_RECL = 5

PROC_REGZ = 2

ADD _RULES = 8-(5 + 2) = 1

These values are obtained by counting the number of rule recall, procedure recall and recognized from the GTAL representation of the interface. The system then uses the prediction regression formula to measure ease of learning.

TIME TO LEARN = 217.05 SECONDS

Figure 10: Example of GTAL

As mentioned before the an advantage of using this graphical formalism is that is it easy to use and easy to learn Once this graphical representation is in IDES the user can easily make modification and quickly determine the impact on the ease of use and ease of learning of an interface IDES can also be used as a learning tool. The user can easily test many different interfaces This accelerates the learning process of understanding what affects ease of use and ease of learning.

The next section describes the hypotheses that are developed to test the main premise of the paper that using IDES to evaluate ease of use and ease of learning of an interface is a feasible approach

VII. RESEARCH QUESTIONS

A. INTRODUCTION

The main proposition of this paper is that using an automated tool such as IDES to evaluate interfaces is both more accurate and faster then using a manual method Also, the advantages of using IDES should be more significant the more complex the interface is Furthermore, users of IDES will have a more positive attitude toward the use of models, such as Keystroke and TAMPEL, to evalute interfaces

Accuracy is defined as evaluating ease of use and ease of learning of an interface as close as possible to the model's accurate results If, for example, an interface's ease of learning is evaluated by the using the TAMPEL model to me 230 seconds the participant's evaluation will be compared to this result

Speed of evaluation is defined as the time it takes to evaluate an interface The participants evaluation completion time are compared to each other

General attitude trends are measured for both the users using IDES and the users using the manual method. The attitudes measured are usefulness of models, ease of learning and satisfaction Also, the ease of use of IDES is measured

The complexity of the interface is measured as the number of functions that are available to the users. A two function interface is considered simple while a four function interface is considered complex.

H1: The useres of IDES are more accurate when evaluating the ease of use and the ease of learning of an interface

H2: The users of IDES are faster when evaluating the ease of use and the ease of learning of an interface

H3: The users of IDES have a more positive attitude towards the use of the Keystroke-level Model and TAMPEL

B. HYPOTHESIS 1: ACCURACY

The two evaluation parameters used in this research is the ease of use (Keystroke-level Model) and ease of learning (TAMPEL). Accuracy of measurement is tested in terms of ease of use and ease of learning. The first statistical test is to determine if the complexity of an interface impacts the results. If interface complexity does not have significance on the results the Simple and Complex tasks are combined together and the t-test is performed for the aggregate results.

H.	PARAMETERS	DESCRIPTION
		Ease of Use
H10	Simple vs Complex	Complexity of interface significantly impact the results
H11	Simple Task	The group using IDES will be more accurate
H12	Complex Task	The group using IDES will be more accurate
	Simple vs. Complex	
H13	Manual	The simple task is more accurate
H14	IDES	No difference between simple and complex task
		Ease of Learning
H15	Simple vs. Complex	Complexity of interface significantly impacts the results
H16	Simple Task	The group using IDES will be more accurate
H17	Complex Task	The group using IDES will be more accurate
	Simple vs. Complex	
H18	Manual	The simple task is more accurate
H19	IDES	No difference between simple and complex task

Table 1: Description of Hypothesis-Accuracy

1) Description of Expected Results-Ease of Use

Simple vs. Complex (H10). The Keystroke-level Model has a specific representational grammar for describing computer interfaces. The "syntax" of the model must be mastered to accurately evaluate interface in terms of ease of use. With IDES this grammar is hidden from the user. Because of this it is expected that the evaluation of the interface will be more accurate for the group using the automated tool. Furthermore, this grammar becomes more complex to apply the more complex the interface. This suggests that the difference between the accuracy of the simple and complex task evaluation is greater for the manual users.

Simple Task (H11). Since the representational grammar is hidden from tne user of IDES it is expected that the participants using this tool will have more accurate results.

Complex Task (H12). Here the results are expected to be the same as for the simple task. The users of IDES will be more accurate in their evaluation of the interface. The more complex the task, the more chances that there are to make mistakes in applying the Keystroke-level Model.

Simple vs. Complex-Manual (H13). As mentioned previously the more complex an interface the more difficult it is to apply models such as the Keystroke-level Model. Because of this it is expected that the simple task's results will be more accurate then the complex task's results

Simple vs. Complex-IDES (H14). Since the users of IDES do not have to concern themselves with the syntax of the models it is expected that the accuracy of the evaluation will not be affected by the complexity of the task.

2) Description of Expected Results-Ease of Learning

Simple vs. Complex (H15). Similar to using the Keystroke-level Model syntax TAMPEL has its own representational grammar. This grammar is quite complex and cumbersome. It is expected that the complexity of the task will lead the manual users to be less accurate then the users of IDES as the complexity of the task increases. This is due to the fact that the users of IDES are shielded from having to use the "Syntax" of TAMPEL.

Simple Task (H16). The results for the evaluation of the ease of learning of an interface is expected to be similar then the results for the ease of use. For a simple task the users of IDES are expected to evaluate the interface more accurately Again, TAMPEL's representational grammar is complex and cumbersome. IDES hides the user fromm this complexity.

Complex Task (H17). Again, for the same reasons as for the simple task, the users of IDES will be more accurate when measuring ease of learning of the interface.

Simple Task vs. Complex Task-Manual (H18). As the complexity of the interface increases the application of TAMPEL becomes more complex. Consequently, the simple interface will be evaluated more accurately then the complex interface.

Simple Task vs. Complex Task-IDES (H19). Since the users of IDES are not affected by the syntax of the model it is expected that the simple task evaluation will be more accurate then the complex task.

C. HYPOTHESIS 2: SPEED OF EVALUATION

H.	PARAMETERS	DESCRIPTION
H20	Simple vs Complex	Complexity of interface significantly impact the results
H21	Simple Task	The group using IDES will be faster
H22	Complex Task	The group using IDES will be faster
	Simple vs. Complex	
H23	Manual	The simple task is evaluated faster
H24	IDES	No difference between simple and complex task

Table 2: Description of Hypothesis-Speed of Evaluation

Simple vs. Complex (H20). As mentioned previously the more complex an interface the more complex the application of the models that measure ease of use and ease of learning. Since the users of IDES do not have to use the model's syntax they should be able to measure ease of use and ease of learning of simple and complex interface in a similar way. The complexity of the interface should not impact the evaluation speed. It is expected that the difference between the speed of evaluation for the simple and complex task will be greater for the manual users.

Simple Task (H21). Since the representational grammar of the models is hidden form the users of IDES it is expected that the group using the automated tool will evaluate the interfaces faster.

Complex Task (H22). For the same reason as for the simple task the users of IDES are expected to take less time to evaluate the ease of use and the ease of learing of the interface.

Simple vs. Complex-Manual (H23). As mentioned previously the models become increasingly complex as the interface becomes more complex. Consequently, the evaluation of the simple interface should be faster.

Simple vs. Complex-IDES (H24). Since IDES makes it more simple to represent interfaces it is expected that there will be no significant differences between the simple and the complex interface evaluation speed.

D. HYPOTHESIS 3: ATTITUDE TOWARDS MODEL

The attitude towards the use of the models is measured in terms of usefulness, ease of learning, and satisfaction. The manual group is compared to the IDES group. Also, the ease of use of the IDES is tested to ensure the adequacy of using such a tool. The following table provides a summary of the expected results:

		Attitude Towards Model
H31	Usefulness	Users of IDES will find the models more useful
H32	Learning	Users of IDES will find the models easier to learn
H33	Satisfaction	Users of IDES will be more satisfied
		Attitude Towards IDES (User Friendliness)
H35	Ease of Use	Users will find IDES easy to use

Table 3: Hypothesis 3-Attitude Towards the use of the Models

1) Attitude Towards the Use of the Model

Usefulness (H31). Because of the complexity of using the models it is expected that the users of the manual method will find the models less useful. Something that is too complex to use loses its usefulness. Since the users of IDES are not required to apply the complex syntax of the models to the evaluation of the interface they will find the models more useful.

Learning (H32). Learning to measure the interfaces is expected to be easier for the users of IDES.

Satisfaction (H33). The users of IDES are expected to be more satisfied with using the models to evaluate interfaces. The automated tool makes it much easier to measure ease of use and ease of learning. Since it is much easier the results should demonstrate that the users of IDES are more satisfied.

2) Attitude Towards IDES (Ease of Use)

User Friendliness (H34). The users are expected to find the system easy to use.

VIII. RESEARCH METHODOLOGY

A. Experimental Procedure

The purpose of this research paper is to assess the suitability of applying complex Task-theoretic Analytic Models such as the Keystroke-level Model and TAMPEL using IDES. Two groups are asked to evaluate a different set of interfaces One group evaluates both a simple and a complex interface by using IDES while the other group performs the measurements manually. The participants are compared as to the accuracy and the time taken to perform their evaluation. Also, their attitudes towards the technique (manual vs. IDES) used to evaluate the interfaces is compared

The experiment can be divided in two parts. The first part is a class lecture given to the participants to teach them how to measure the ease of use and the ease of learning of an interface using the Keystroke-level Model and TAMPEL. The second part is the actual experiment where the participants apply the Keystroke-level Model and TAMPEL to evaluate different interfaces.

CLASS LECTURE. The purpose of the class lecture is to ensure that the participants have an understanding of the Task-theoretic Analytic models used to evaluate the ease of use and the ease of learning of an interface. Most of the participants do not have formal training in HCI and for this reason it is necessary to introduce them to the various topics to make them proficient with the Keystroke-level Model and TAMPEL

The lecture is divided into five sections (refer to **appendix A**) for the agenda used in the lecture). First a brief introduction of the study is given to the subjects followed by a description of the tools and methodologies used in human computer interaction. The participants are introduced to concepts such as design guidelines, design principles, and iterative design methodologies. The advantages and the disadvantages of each approach is discussed. The students are then introduced to the various interface evaluation techniques that are available. Heuristic evaluation, software guidelines, user testing, and task-theoretic analytic models are reviewed along with their strengths and weaknesses. The students are then taught how to measure the ease of use and the ease of learning of an interface by applying the Keystroke-level Model and TAMPEL.

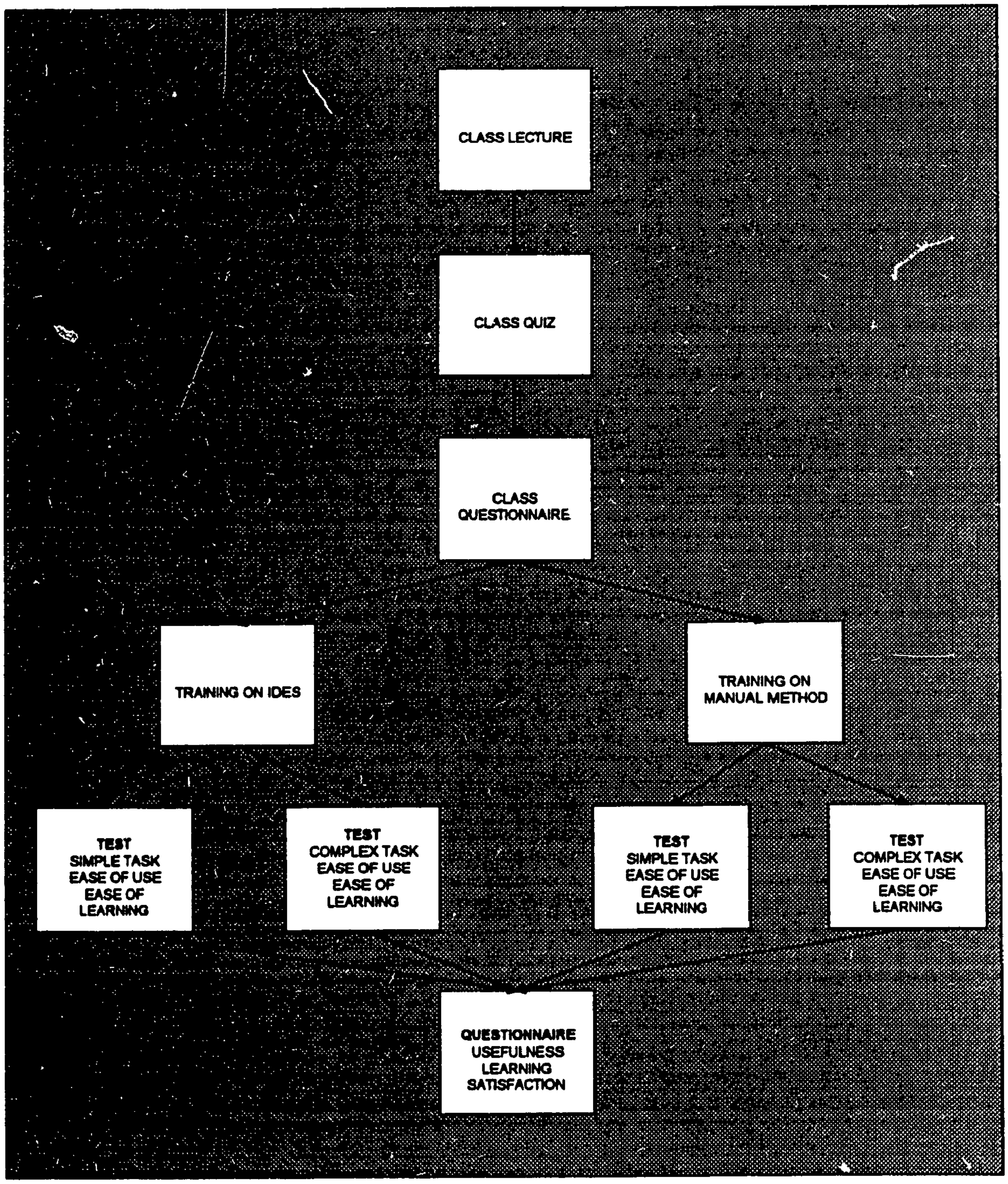

CLASS LECTURE
CLASS QUIZ
CLASS QUESTIONNAIRE
TRAINING ON IDES
TRAINING ON MANUAL METHOD
TEST SIMPLE TASK EASE OF USE EASE OF LEARNING
TEST COMPLEX TASK EASE OF USE EASE OF LEARNING
TEST SIMPLE TASK EASE OF USE EASE OF LEARNING
TEST COMPLEX TASK EASE OF USE EASE OF LEARNING
QUESTIONNAIRE USEFULNESS LEARNING SATISFACTION

Figure 11: Research Methodology

In order to facilitate the learning of the Task-theoretic Analytic Models the remaining part of the lecture is devoted to in-class examples and exercises. For the Keystroke-level Model a simple example is provided along with a set of heuristics to explain where the M operators need to be inserted. A second example is given but this time the students are given a few minutes to solve the problem on their own. The solution is then reviewed in class and questions are answered to clarify areas of confusion. The students are asked to perform a total of four measurement of the ease of use using the Keystroke-level Model. The exercises are designed to be increasingly complex.

TAMPEL is taught in a similar way. First a brief description of the model is given along with a simple example to reinforce the principles. Measuring ease of learning is a more complex undertaking then measuring ease of use. For this reason a greater portion of the class lecture is devoted to clearly understand the model. The participants are asked to measure the ease of learning of two interfaces (one more complex then the other) and the solutions are reviewed in class. Again, a question period is provided to ensure that the students understand the mechanics of TAMPEL. A chart describing the general rules of the TAMPEL model is given to the participants to be used as reference material.

The purpose of the class lecture is to teach the participants how to apply the models used to measure ease of use and ease of learning. To ensure that the students internalize the techniques learned a class quiz (refer to Appendix B) is given. There are two reasons why a class quiz is administered. One reason is that the students must be able to apply the models effectively to ensure the success of the experiment. Another reason is to verify that the members of the different test groups are similar in their understanding of the models.

There are two ways to ensure that the students are motivated to participate in the experiment. The whole exercise is worth 5% of the total mark of a Systems Design course. Furthermore, a price of $100 is to be given to the member of each group that evaluates the interface the most accurately and the most rapidly.

At the end of the class lecture each student is systematically assigned to one of the two groups. This was done by giving a number to each participants starting from the front row and designate the odd numbers to be in one group while the even numbers are in the other. In order to take the full benefit of the class lecture the experiment is performed during the following two days after the lecture. Each participant is given a time slot of when to perform the actual experiment. The group using IDES must sign up for two time slots since they must first be taught how to use the automated tool. These training session are given in groups of three. The lecture is given on a Wednesday night between 5:40 PM and 8:20 PM. This is the regular time for this Systems Design class.

EXPERIMENT. The test consists of evaluating the ease of use and the ease of learning of both a simple interface and complex interface (Refer to **appendix C**). The two groups are given the same instruction set along with a detailed description of the interfaces. The simple interface consists of measuring the ease of use and the ease of learning of a two command system. The complex interface has four different commands. The more commands an interface the more complex it is.

Group 1 applies the Keystroke-level Model and TAMPEL manually. A special study room with cubicles is used for the participants to complete the experiment. This study room provides a quiet area and ensures that there are no interferences that can distract the students. Each student is greeted by the researcher and given the actual assignment. The assignment is self-explanatory. The participants are also given a copy of the agenda that is used in the class lecture along with the solutions to the class exercises. Each participants is asked to spent 15 minutes to review these exercise. This is done to ensure that both group get approximately the same training time. Once the two interfaces are evaluated the participants complete a brief questionnaire.

The second group is treated a little different. The participants need to learn how to use IDES. For this reason a 15 minutes training session is provided to groups of three students. The researcher briefly explains the use of the Graphical Method to describe interfaces and the class exercises are evaluated using IDES. Each student is then asked to perform an analysis using IDES. This training session ensures that the participants are familiar with the mechanics of using IDES. This period is equivalent to the fifteen minutes review that the other group is asked to perform.

The participants using IDES then come back to do the actual interface evaluation at their allotted time. Again, the researcher greets them and provides them with the actual assignment along with a copy of the class agenda with the solutions to the class exercises. The students perform the evaluation and then complete the questionnaire.

There is an attempt to have both groups subjected to similar circumstances. The researcher treats each participants the same way. The timing of the evaluations is done by having the students report to the researcher that they are finished with a particular portion of the assignment. Internal validity is maximized by having both groups subjected to similar circumstances with just one difference in that one group evaluates the interfaces manually while the other group uses IDES. This ensures if the results are different it is due to the method of evaluation.

B. Sample

In order to emphasize that the results are due to the treatments of the experiment an attempt is made to have a sample that has similar attributes. For this reason all of the participants are third year M.I.S. students from a Systems Design class. The sample consists of 27 students- 14 using IDES and 13 using the manual method. Each participant is asked to fill out a questionnaire to assess their background (Refer to **appendix D**). The following information is collected: 1) Years of university education, 2) University program presently enrolled in (Major and Minor), 3) M.I.S. or Computer Sciences courses taken so far, 4) Formal M.I.S. work experience obtained so far, and information on M.I.S. knowledge in general-a)operating system, b) computer hardware, c) defining information requirements d) designing inputs and outputs for application software, e) developing file structures, f) use of office automation.

Based on the results of the questionnaire the two groups are very similar (refer to **appendix E** for results). Furthermore, all the participants are given the same class lecture and the same treatment during the experiment. The results from the class exercise showed that both groups obtained similar scores when evaluating the interface during the class lecture (refer to **appendix F** for result of class quiz) This ensures that the results obtained are not due to differences between the two groups.

It may be difficult to generalize the results to the community of interface designers since the sample consists of M. I. S. sudents. These MIS students may have different attributes then the general population. Nevetherless, an attempt was made to maximize external validity by using students that will eventually be part of the target community.

C. Data Analysis

1. Operationalization of the variables

Speed of Evaluation. Each participant is measured on how fast the evaluation of the ease of use and ease of learning is performed. A time is obtained for both the simple and complex interface.

Accuracy of Results. Ease of use and ease of learning is measured with the use of the Keystroke-level Model and TAMPEL. As discussed in a previous section these models have been tested repeatedly and have been found to be accurate. The participants are asked to measure the ease of use and the ease of learning for each interface. Their evaluation is compared to the actual accurate time for the ease of use and the ease of learning (refer to **appendix G** or the solutions to the class exercise) .

Complexity of the Interface The complexity of an interface is deemed to be determined by the number of commands available to the user. For this reason the simple interface is a two-command system while the complex interface is a four-command system.

Attitude towards the evaluation techniques. In the class lecture the participants were introduced to different approaches to evaluating interfaces. (refer to **appendix G** for questionnaire on attitude). The questions relate to the usefulness, ease of learning and satisfaction of using the Keystroke-level Model and TAMPEL for the users of IDES and for the manual users. Second, the questions relate to how IDES is as a tool. IDES must be easy to use in order to be an effective tool. Several questions relate to this ease of use. The questionnaire use a five point Likert scale with "0" as a neutral. This is an effective scale to evaluate variables such as usefulness, ease of use, ease of learning, and satisfaction.

2. Statistical Tests

A t-test analysis is performed to test for the differences in accuracy of interface evaluation between the group using IDES and the group using the manual method. Also, the difference in the time to evaluate ease of use and ease of learning between the two

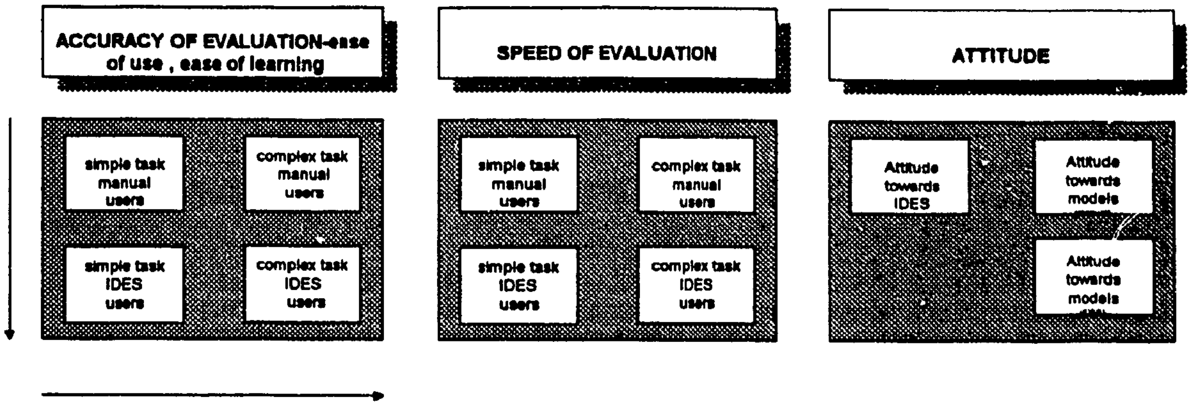


Figure 12: Variables Tested

groups is tested using a t-test. 95% confidence intervals are calculated to show the direction of the differences if their are any. The attitude was tested using similar statistical procedures. It is important to not that absolute values are used to compare both groups for the accuracy of measurements.

The first t-test is performed to determine if the interface complexity affects the results obtained by the participants. If there is an effect t-tests are calculated for the simple and complex interface separately. If complexity does not significantly impact the results t-tests are calculated for both the the simple and complex results together. The aggregate results are used to test for the impact of IDES on the accuracy of the evaluation and on the speed of the evaluation.

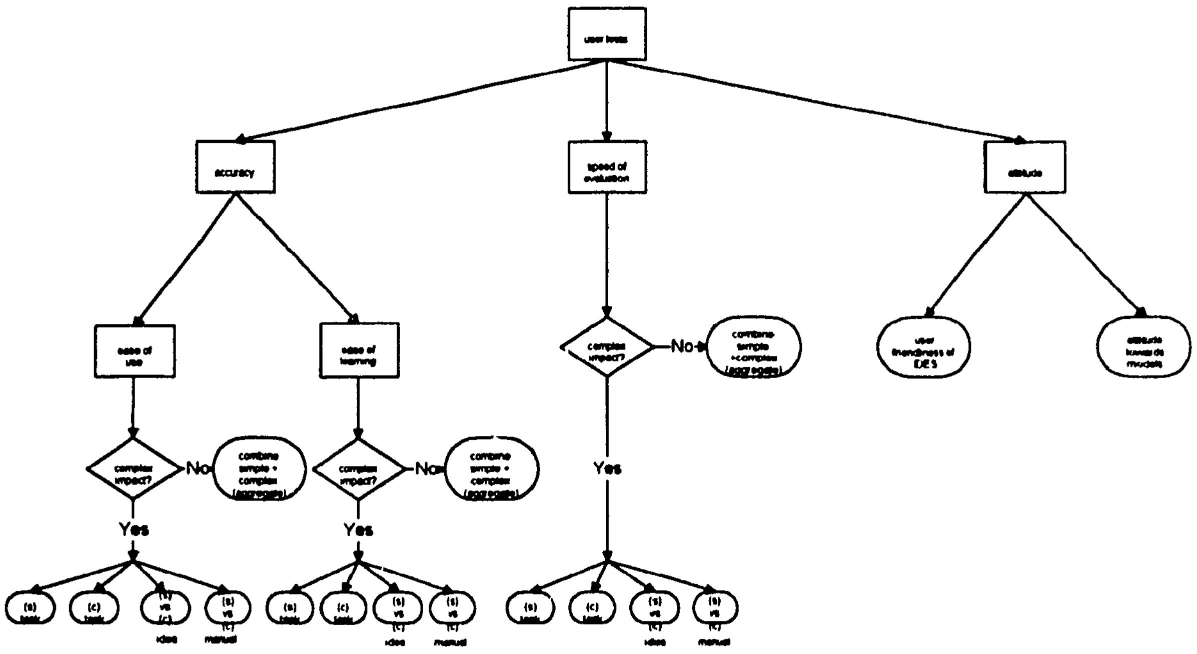


Figure 13: Statistical Procedures

D. LIMITATION OF METHOD

CONSTRUCT VALIDITY. The extent to which the constructs of theoretical interests are successfully operationalized determines the validity of the research. The speed of evaluation is easily measured using a stop watch The accuracy of the results are based on the evaluation of the interfaces obtained by applying accurately the Keystroke-level Model and TAMPEL. These models have been empirically tested and provide accurate results of the ease of use and the ease of learning. The attitude toward the tool is more difficult to measure. For this reason multiple operationalization of the variables of interest is used. The purpose of the attitude questionnaire is to get a general idea of the usefulness, ease of learning and satisfaction of the Keystroke-level Model, TAMPEL, and IDES

INTERNAL VALIDITY. The background questionnaires ensures that there are no differences between the two groups. Randomization is tested Motivation for the study was ensured by giving $100 to the participant with the most accurate score obtained in the least amount of time. Furthermore, 5% of the total mark of the class is assigned to the experiment. The time between the class lecture and the actual experiment is minimized. These precautions increase the internal validity.

EXTERNAL VALIDITY. It is difficult to state with certainty that the results of this study can be generalized to the population of interface designers. There are important differences between the participants and the rest of the designer population One of the important difference is that the participants have little work experience and are all from the same university. The study confirms the feasibility of using IDES to apply complex theories to the evaluation of interface. There are no reasons why these results could not be obtained with software developers.

IX. DISCUSSION

The results of this study demonstrate the advantages of an automated graphical representational tool such as IDES to evaluate an interface in terms of ease of use and ease of learning. This section reviews the statistical tests performed along with an explanation of the results. The first section is about the effect of IDES on interface on the accuracy of evaluation. The second section assesses the impact of IDES on the time it takes to measure the ease of use and the ease of learning of an interface. The impact of interface complexity is covered in these two sections. Finally, The last section is about users attitudes towards the application of Task-theoretic Analytic Models The participants using the manual system are compared to the users of IDES.

H.	**PARAMETER**	**EXPECTED RESULTS**	**RESULTS**
H1	Accuracy of Evaluation	Users of IDES will be more accurate	Users of IDES are more accurate for ease of learning evaluation
H2	Speed of evaluation	Users of IDES will be faster	Users of IDES are faster
H3	Attitude	Users of IDES will have a more positive attitude towards the models	Users of IDES have a more positive attitude towards the models

Table 4: Results of main tests

The results also suggest that the complexity of the interface does not significantly impact the accuracy of the evaluation. Complexity of the interface does seem to affect the results of the speed of evaluation-the simple interfaces take less time to evaluate. Also, the users of IDES seem to find the tool to be user friendly.

A. HYPOTHESIS 1: ACCURACY

1. Complex-Simple (Manual vs. IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H10	Manual	13	7.58	Complexity	-1.42	(-8.2 , 1.5)
	IDES	14	10.90			

Table 5: Impact of interface complexity on accuracy of evaluation for ease of use

Refer to Table 5. Since T = -1.42 (p=0.017) we accept H_0 This result suggests that there is no difference between the results of the users of the manual method and the users of IDES in terms of complexity of interface.

The Keystroke-level Model has a representational grammar that could be considered easy to learn and to apply. This would suggest that the complexity of the interface should not impact the accuracy of evaluation. The results demonstrate that using IDES for both a simple and complex interface does not significantly impact the accuracy of the results. For measuring ease of use with the Keystroke-level Model the complexity of the interface does not impact the accuracy of the measurements.

Since complexity of the interface has no significance on the results the next t-test combines the results for both the simple and complex interface and only tests for the impact of IDES vs. the manual method on the accuracy of the ease of use measurements.

2. Ease of Use-Simple-Complex Combined (Manual vs. IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H10a	Manual	13	8.33	Manual vs. IDES	0.55	(-3.5 , 5.9)
	IDES	14	7 11			

Table 6: Accuracy of Results for Ease of Use-combined groups (simple-complex)

Refer to Table 6. Since T = 0.55 (p=0.59) we accept H_0. <u>These results suggest that when evaluating the ease of use of an interface in terms of ease of use with the Keystroke-level Model the use of IDES does not impact the accuracy of the evaluation</u>

As mentioned previously the Keystroke-level Model is easy to learn and to apply. Once the users have evaluated a few interfaces with the model their effectiveness increases. The users of the manual method had the chance to use the Keystroke-level model during the class lecture and during the preparation period just before the experiment. Because of this they were able to apply the model accurately This is demonstrated by the statistical test obtained. These results suggest that there seems to be no advantage in using a tool like IDES to evaluate the ease of use of an interface with the Keystroke-level Model.

3. Complex-Simple (Manual vs. IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H15	Manual	13	249	Manual vs. IDES	1.42	(-41 , 205)
	IDES	14	167.1			

Table 7: Impact of interface complexity on accuracy of evaluation for ease of learning

Refer to Table 7 Since T = 1.42(p=0.18) we accept H_0. <u>The result of this statistical test suggest that the complexity of the interface does not significantly impact the accuracy of the ease of learning evaluation for both the manual users and the users of IDES.</u>

Intuitively, one would believe that the more complex the interface is the more difficult it will be to evaluate the ease of learning by using the manual application of TAMPEL. The results suggest otherwise. The complex interface could be not "complex" enough. As mentioned previously complexity was defined as having two more functions on the screen. It is possible that the difference in complexity between the simple and the complex interface was not great enough Also, one could argue that once the application of the model is begun manually the task of measuring the interface becomes quite mechanical. It could be that it just take longer to evaluate a complex interface but it is not necessarily more difficult.

Since complexity of the interface has no significance on the results the next t-test combines the results for both the simple and complex interface and only tests for the impact of IDES vs. the manual method on the accuracy of the ease of learning measurements.

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H15a	Manual	13	273	Manual vs. IDES	2.33	(11 , 280)
	IDES	14	127.8			

Table 8: Accuracy of Results for ease of learning-combined groups (simple-complex)

Refer to Table 8. Since T = 2.33(p=0.037) we reject H_0. <u>There is enough evidence to suggest that the users of IDES were more accurate when evaluating the ease of learning with the TAMPEL model.</u>

Learning how to apply TAMPEL to measure ease of learning is a difficult process. There are many rules that must be clearly understood and the syntax of the representational grammar can be confusing. Using IDES the interface evaluator is only required to understand how to represent an interface graphically and to know when to apply transfers. The user does not need to know the mechanics ot TAMPEL. The results suggest that using IDES is beneficial when evaluating the ease of learning of an interface.

B. HYPOTHESIS 2-SPEED OF EVALUATION

1. Complex-Simple (Manual vs IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H20	Manual	13	8.54	Manual vs. IDES	3.54	(3.4 , 13.2)
	IDES	14	0.21			

Table 9: Impact of interface complexity on speed of evaluation

Refer to Table 9. Since T = 3.54(p=0.002) we reject H_0 <u>There seems to be enough evidence to suggest that the complexity of the task will affect the time it takes to evaluate an interface in terms of ease of use and ease of learning.</u> These results demonstrate that IDES becomes more useful the more complex the task is

This is consistent with expected results. The users of IDES do not have to measure ease of use and ease of learning of the interface with the cumbersome manual method. The group using IDES is only required to represent the interface using GTAL. This is easier then applying the Keystroke-level Model and TAMPEL.

2. Simple Task (Manual vs. IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H21	Manual	13	28.46	Manual vs. IDES	2.47	(1.0 , 11.60)
	IDES	14	22.14			

Table 10: Speed of evaluation for Simple Task

Refer to Table 10. Since T = 2.47(p=0.02) we reject H_0. <u>This would suggest that the manual users take more time to evaluate the ease of use and ease of learning of a simple interface with the Keystroke-level Model and TAMPEL then the users of IDES.</u>

As mentioned previously the users of IDES to not have to apply the "syntax" of the evaluation models. This makes it much simpler and faster to evaluate the interface. For this reason the users of IDES take less time to complete their measurements.

3. Complex Task (Manual vs. IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H22	Manual	13	37.00	Manual vs. IDES	4.39	(7.70 , 21.60)
	IDES	14	22.36			

Table 11: Speed of evaluation for Complex Task

Refer to Table 11. Since T = 4.39(p=0.0002) we reject H_0. <u>There seems to be enough evidence to suggest that the manual users take more time to evaluate the ease of use and the ease of learning of a complex interface then the users of IDES.</u> Again, these results are expected since IDES simplifies the the process of evlauating ease of use and ease of learning of an interface.

4. Simple vs.Complex Task (Manual)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H23	Simple	13	28.46	Simple-Complex	-2.72	(-15.1 , -2.0)
	Complex	13	37.00			

Table 12: Speed of evaluation for Manual-Simple vs. Complex

Refer to Table 12. Since T = -2.72(p=0.013) we reject H_0. <u>There seems to be enough evidence to suggest that the more complex the interface the longer it takes the manual users to evaluate the ease of use and the ease of learning of that interface.</u>

When measuring ease of use and ease of learning every function of the interface must be accounted for. This means that the user must apply the "syntax" of the models used and must calculate the impact of this function. This takes time. The more function you have the more time this will take. The results presented in Table 12 supports this.

5. Simple vs.Complex Task (IDES)

H.	GROUP	N	MEAN	DESCRIPTION	T-TEST	95% C. I.
H24	Simple	14	22.14	Simple-Complex	-0.08	(-6.0 , 5.5)
	Complex	14	22.36			

Table 13: Speed of evaluation for IDES-Simple vs. Complex

Refer to Table 13. Since T = -0.08(p=0.94) we accept H_0. The results suggest that for the users of IDES there is no time difference between evaluating a simple interface and a complex interface.

As the user becomes more familiar with representing interfaces with GTAL he/she becomes faster. This is a possible explanation for the reason as to why the results show that there is no difference in the speed of evaluation between the simple interface and the complex interface. Also, some of the functions between the complex interface and the simple interface were similar. The users of IDES had already created the GTAL for these functions. Therefore, it took them less time when evaluating these functions in the complex interface. Familiarity with the use of GTAL and previous evaluation of functions may account for the fact that there are no difference between the simple and complex interface.

C. HYPOTHESIS 3: Attitude Towards Keystroke-level Model and TAMPEL

1. Usability Test of IDES

For IDES to be an effective tool to evaluate ease of use and ease of learning the useres must find it easy to use. The underlying assumption is that if IDES is complex and cumbersome the users will not find it a useful tool to apply the Keystroke-level Model and TAMPEL. The following statistical test provide support to believe that IDES was considered to be user friendly.

QUESTIONS	N	MEAN	STDEV	T-VALUE	P-VALUE	95 % C. I.
Easy/Difficult (7)	14	1.286	0.611	7.87	0.0000	(0.933,1.639)
Clear (9)	14	0.786	0.893	3.29	0.0058	(0.270,1.301)
Flexible (10)	14	1.071	0.730	5.49	0.0001	(0.650, 1.493)
Use (12)	14	1.214	0.426	10.67	0.0000	(0.968, 1.460)
Format (13)	14	1.143	0.770	5.55	0.0001	(0.698, 1.588)
Clear (15)	14	1.286	0.469	10.26	0.0000	(1.015, 1.556)

Table 14: User Friendliness of IDES

Refer to Table 14. All the T-values suggest that IDES is user friendly. For IDES to be en effective tool to evaluate interfaces it must be easy to use. Six indicators are used to evaluate the ease of use: Is IDES easy/difficult to use, is the information from IDES clear, is IDES flexible, how easy is IDES to use, how useful is the presentation format, and finally, is the interaction with the tool clear and understanble. According to Table 14 the results demonstrate that there is enough evidence to believe that the users of IDES found the tool to be user friendly. These results were to be expected . IDES is developed in such a way simplify its use. Graphical interfaces are used and the information is presented in a simple easy to use format.

2. Attitude Towards Models

H31	Usefulness	GROUP	N	MEAN	T-TEST	95% C. I.
	Faster/Slower (1)	Manual	13	0.54	-2.87	(-1.80 , -0.27)
		IDES	14	1.571	p=0.011	
	Performance (2)	Manual	13	0.008	-2.89	(-1.83 , -0.30)
		IDES	14	1.143	p=0.003	
	Productive (3)	Manual	13	0.538	-2.40	(-1 39 , -0.10)
		IDES	14	1.286	p=0.025	
	Effective (4)	Manual	13	0.08	-2,72	(-1.76 , -0.23)
		IDES	14	1.071	p=0.013	
	Easier/Harder (5)	Manual	13	0.38	-2.43	(-1.94 , -0 15)
		IDES	14	1.429	p=0.024	
	Useful (6)	Manual	13	0.62	-2.76	(-1.56 , -0.21)
		IDES	14	1.50	p=0.014	

Table 15: Usefulness of Models-Manual vs. IDES

Usefulness. Refer to Table 15. All the T-values suggest that the users of IDES have a more positive attitude towards the usefulness of the models for evaluating the ease of use and the ease of learning of the interfaces. The P-values range from 0.001-0.025).

One of the premise of this research is that models such as the Keystroke-level Model and TAMPEL are not used by interface evaluators because of the difficulty of understanding the model's application. Because of the complex nature of these models the perception of the usefulness of the models should be negative. If there is a tool such as IDES that simplifies interface evaluation the attitude towards the usefulness of the Keystroke-level Model and TAMPEL should be more positive. The results in Table 15 supports this by demonstrating that the users of IDES are more positive towards the usefulness of the models.

H32	Learning	GROUP	N	MEAN	T-TEST	95% C. I.
	Skillful (8)	Manual	13	-0.62	-5.37	(-2.66 , -1.15)
	(11)	IDES	14	1.286	p=0.0001	

Table 16: Ease of Learning of the Application of the Models-Manual vs. IDES

Learning. Refer to Table 16. The T-value suggest that the users of IDES found the application of the Keystroke-level Model and TAMPEL to be easier to learn.

As mentioned in a previous section one of the reasons evaluation approaches such as the Keystroke-level Model and TAMPEL are not widely used is the fact that they are complex and difficult to apply. IDES simplifies the application of these models. It is therefore expected that the users of IDES will find it easier to become skillful at measuring the ease of use and the ease of learning of interfaces. The results in Table 16 supports this proposition.

H33	Satisfaction	GROUP	N	MEAN	T-TEST	95% C. I.
	Satisfaction (10)	Manual	13	-0.54	-3.78	(-2.38 , -0.69)
	(15)	IDES	14	1.000	p=0.001	

Table 17: Satisfaction with the Models-Manual vs. IDES

Satisfaction. Refer to Table 17. The T-value suggests that the users of IDES are more satisfied with the Keystroke-level Model and TAMPEL then the manual users.

Because of the difficulty to use and to learn how to use the Keystroke-level Model and TAMPEL (Refer to Table 15 and Table 16) it is expected that the satisfaction with the manual application of the models will be more negative for the manual users. The results from Table 17 demonstrates that the users of IDES are more satisfied.

XI. CONCLUSION

Table 18 provide a summary of the main tests of the experiment. These results demonstrate the advantage of using a tool such as IDES to evaluate the ease of use and the ease of learning of an interface.

EVALUATION	RESULT
Ease of Use (H1)	
Complex-Simple (Manual vs. IDES)	Complexity does not affect the results
Manual vs. IDES (Simple + Complex)	IDES is more accurate
Ease of Learning (H1)	
Complex-Simple (Manual vs. IDES)	Complexity does not affect the results
Manual vs. IDES (Simple + Complex)	IDES is more accurate
Speed of Evaluation (H2)	
Complex-Simple (Manual vs. IDES)	Complexity affects the results
Simple Task	Manual group takes longer
Complex Task	Manual group takes longer
Simple Task vs. Complex Task (Manual)	Complex task takes longer
Simple Task vs. Complex Task (IDES)	No difference
Attitude Towards Model (H3)	
Usefulness	IDES find the models more useful
Learning	IDES find models easier to learn
Satisfaction	IDES are more satisfied with the models
User Friendliness of IDES (H3)	
Ease of Use	Users find IDES user friendly

Table 18: Summary Results

One of the premise of this paper is that Task-theoretic Analytic Models such as the Keystroke-level Model and TAMPEL are not used by interface evaluators because of their complexity. Other methods to evaluate interfaces such as user testing, design guidelines and principles, walkthroughs, decomposition, and heuristics, all have disadvantage thata renders them, to a certain, ineffectual when evaluating an interface (refer to the section on evaluation methods). Task-theoretic Analytic Models provide a method to measure human factor parameters such ease of use and ease of learning accurately. A tool that would simplify their use and ease of learning without Human Factors experts. The results from this study suggest that IDES could be such a tool.

It can be concluded that evaluating interfaces with IDES is faster and provide more accurate evaluations of ease of use and ease of learning. Furthermore, users of IDES had a more positive attitude towards using the Keystroke-level Model and TAMPEL. IDES provide a mean for the interface developer to apply Task-theoretic Analytic Models such as the Keystroke-level Model and TAMPEL. This approach ensures that interface designer can incorporate formal theory into the evaluation of the interface.

As mentioned at the beginning of this study interface designers have been reluctant to integrate formal human-computer interaction theory into the software development process. Some of the reasons for this are that many of the models are too complicated and cumbersome to use effectivelly. Developers have not been trained to understand complex task-theoretic analytic models. With an automated tool like IDES the interface designer can apply models such as the Keystroke-level Model and TAMPEL without having to understand their intricate representational formalism. Their are many advantages to this. First, formal human factor theory can be integrated into the software development process. Second, designers can evaluate an interface without the help of human factors experts. Third, evaluating many alternatives of an interface can be an effective learning tool. Fourth, interface designs can be evaluated on paper before a prototype is built. This has the potential to decrease the number of iteration necessary to develop an effective interface. Tools such as IDES can make the development of interfaces less expensive, less time consuming, and less complex. IDES offers the possibility to apply the substantial research findings of the Human-Computer Interaction field to interface development.

This study only touches upon one aspect of using automated tools to evaluate interfaces-how IDES can improve evaluation in terms of ease of use and ease of learning. the next step could be to determine how such a tool can impact the actual development of an interface. The next step could be to determine how such a tool can impact the actual development of an interface. This could further demonstrate the potential of graphical automated tools to improve interface development. An experimental study could be designed to determine how much more effective interfaces are when they are developed with a tool such as IDES. Automated tools have the potential to significantly impact interface development

REFERENCES

1 Anderson, N S and Olson, J R (1985) *"Methods for Designing Software to Fit Human Needs and Capabilities"* Proceedings of the Workshop on Software Human Factor, Commission on Behavioral and Social Sciences and Education, National Research Council, Washington, D C National Academy Press

2 Bailey, W A., Knox, S T and Lynch, E F (1988). *"Effects of Interface Design Upon User Productivity"* in CHI'88 Conference Proceedings on Human Factors in Computer Systems, p 207-212

3 Benbasat, J and Dexter, A. (1981). *" An Experimental Study of the Human Computer Interface"* Communications of the ACM, Volume 24, p 752-762.

4 Benbasat, J. and Wand, Y (1984) *" A Structured Approach to Designing Human-Computer Dialogue"* International Journal of Man-Machine Studies, 21, p 105-126.

5 Bennett, J (1983) *" Analysis and Design of the User Interface for Decision Support Systems"* in Building Decision Support Systems, J. Bennett (ed.), Addison-Wesley Readings, MA, p 41-64

6 Bovair, S, Kieras, D E., and Polson, P.G. (1990). *" The Acquisition and Performance of Text-Editing Skill: A Cognitive Complexity Analysis"* Human Computer Interaction, 5, p 1-48

7. Bury, K F (1985) *" The Iterative Development of Usable Computer Interfaces"* in Human Computer Interaction, INTERACT' 84, p 343-348.

8. Buxton, W and Shneiderman, R. (1980). *" Iteration and the Design of the Human Computer Interface"* in Proceedings of the 13th Annual Meeting of the Human Factors Association of Canada, p 300-311.

9 Card, S K., Moran, T P , and Newell, A (1980). *" The Keystroke-Level Model for User Performance Time with Interactive Systems"* Communications of the ACM, Volume 31, p 429-439

10 Card, S K., Moran, T P , and Newell, A (1983) *The Psychology of Human Computer Interaction* Hillsdale, NJ Lawrence Erlbaum Associates, Inc

11. Carrol, J M and Thomas, J C (1982). " *Metaphors and the Cognitive Representation of Computing Systems*" IEEE Transactions on Systems, Man, and Cybernetics, (12:2), p 107-116

12. Carrol, J M. and Olson, J R. (1988) " *Mental Models*" *in Human Computer Interaction*, M. Helander (ed.), Elsevier Science Publishers, Amsterdam, p 45-65

13. Carrol, J.M., Mack, R.L , and Kellog, W A (1988) " *Interface, Metaphors, and User Interface Design*" in Handbook of Human Computer Interaction, M Helander (ed.), Elsevier Science Publishers, Amsterdam, p 67-86

14. DeKleer, J and Brown, J.S (1983). " *Assumptions and Ambiguities in Mechanistic Mental Models*" in Mental Models, D Gentner and A S Stevens (eds.), Hillsdale, NJ Lawrence Erlbaum, p 95-115

15. Douglas, S A and Moran, T P (1983) " *Learning Text Editing Semantics by Analogy*" in CHI' 83, Proceedings of the Conference on Human Factors in Computing Systems, p. 207-211.

16 Good, M.D., Whiteside, J.A , Wixon, D R., and Jones, S J (1984) " *Building a User Derived Interface*" Communications of the ACM, Volume 26, p 1032-1043

17. Gould, J.D and Boies, S J (1983) " *Human Factors Challenges in Creating a Principal Support Office System-The Speech Filing Approach*" ACM Transactions on Information Systems, (1.4), p 273-298

18. Gould, J D. and Lewis, C (1985) " *Designing for Usability: Key Principles and What Designers Think*" Communications of the ACM, Volume 28, p 300-311

19. Gould, J.D., Boies, S.J , Levy, S., Richard, J , and Schoonard, J (1987) " *The 1984 Olympic Message System: A Test of Behavioral Principles of System Design*" Communications of the ACM, Volume 30, p 758-769

20 Gould, J.D., Boies, S.J., and Lewis, C. (1991). " *Making Usable, Useful, Productivity-Enhancing Computer Applications*" Communications of the ACM, Volume 34, p.75-85.

21. Grudin, J. and Welty, C. (1988). " *Experimentation in Computer Science: An Empirical View*" International Journal of Man-Machine Studies, 29, p.613-624.

22. Grudin, J. (1988). " *Intergrating Human Factors in Software Development*" in CHI'88 Conference Proceedings on Human Factors in Computing Systems, p 157-159.

23. Grudin, J. (1990). " *The Computer Reaches Out: The Historical Continuity of Interface Design*" in CHI'90 Conference Proceedings on Human Factors in Computing Systems, p. 261-268.

24. Grudin, J. (1989). " *The Case Against User Interface Consistency*" Communications of the ACM, Volume 32, p. 1164-1173.

25. Guindon, R. (1990). " *Knowledge Exploited by Experts During Software System Design*" International Journal of Man-Machine Studies, 33, p. 279-304.

26. Halasz, F.G. and Moran, T.P. (1982). " *Analogy Considered Harmful*" in CHI'82 Proceedings of the Conference on Human Factors in Computing Systems, p. 383-386.

27. Hartson, H.R. and Hix, D. (1989). " *Towards Empirically Derived Methodologies and Tools for Human Computer Interface Development*" International Journal of Man-Machine Studies, 31, p. 477-494.

28. Hartson, H R. and Hix, D. (1989). " *Human Computer Interface Development Concepts and Systems for its Management*" ACM Computing Surveys, Volume 2, p 5-92.

29. Hewett, T T and Meadow, C.T. (1986). " *On Designing for Usability: An Application of Four Key Principles*" in CHI'86 Conference Proceedings on Human Factors in Computing Systems, p. 247-251.

30. Jeffries, R., Miller, J.R., Wharton, C , and Uyeda, K.M (1991) " *User Interface Evaluation in the Real World: A Comparison of Four Techniques*" in CHI'91 Conference Proceedings on Human Factors in Computing Systems, p 119-124.

31. John, B.E. and Newell, A (1991). " *Towards an Engineering Model of Stimulus-Response Compatibility" in Stimulus Response Compatibility: An Integrated Approach*, R.W. Gilmore and T.G. Reeve (eds.). Amsterdam North Holland

32. Khalifa, M. and Kira, D. (1992) " *An Automated Tool for Describing and Evaluating User Interfaces*" In Press

33. Khalifa, M. (1991) " *A Task Analytic Methodology for Predicting Ease of Learning of Interactive Computer System: TAMPEL" Doctoral Thesis*, The Wharton School, University of Pennsylvania.

34. Kieras, D. and Polson, P.G (1985) " *An Approach to the Formal Analysis of User Complexity*" International Journal of Man-Machine Studies, 22, p. 365-394.

35. Kieras, D. (1988). " *Towards a Practical GOMS Model Methodology for User Interface Design*" in Handbook of Human Computer Interaction, M Helander (ed.), Elsevier Science Publishers, North Holland, p 135-157

36. Laird, J.E., Newell, A., and Rosenbloom, P.S. (1987). " *SOAR: An Architecture of General Intelligence* " Artificial Intelligence, 33, p. 1-64.

37. Lewis, C., Polson, P G , Wharton, C., and Rieman, J (1990). " *Testing a Walkthrough Methodology of Theory Based Design of Walk-Up-And-Use Interfaces*" in CHI'90 Conference Proceedings on Human Factors in Computing Systems, p. 235-242

38. Maguire, M (1982). " *An Evaluation of Published Recommendations on the Design of Man-Computer Dialogue*" International Journal of Man-Machine Studies, 16, p 236-261

39. Mantei, M.M. and Teorey, T.J. (1988). *" Cost/Benefit Analysis for Incorporating Human Factors in the Software Life cycle"* Communications of the ACM, Volume 31, p. 429-439

40. Meister, D. (1986). *Human Factors Testing and Evaluation.* North Holland Science Publishers, Amsterdam, Netherlands.

41. Molich, R. and Nielsen, J. (1990). *" Improving a Human Computer Dialogue"* Communications of the ACM, Volume 33, p. 338-349.

42. Moran, T P. (1981). *" An Applied Psychology of the User"* Computing Surveys, Volume 13, p. 1-11.

43. Moran, T.P. (1981). *" The Command Language Grammar: A Representation for the User Interface of Interactive Computing Systems"* International Journal of Man-Machine Studies, 15, p. 3-50.

44. Nielsen, J. and Molich R. (1990). *" Heuristic Evaluation of User Interfaces"* in CHI'90 Conference Proceedings on Human Factors in Computing Systems, p. 249-255.

45 Nickerson, R.S. (1981) *" Why Interactive Computer Systems Are Sometimes not used by People who Might Benefit From Them"* International Journal of Man-Machine Studies, 15, p. 469-483.

46. Norman, D A. (1984). *" Stages and Levels in Human Machine Interaction"* International Journal of Man-Machine Studies, 21, p. 365-375.

47. Norman, D.A. (1986). *" Design Principles for Human Computer Interfaces"* in User Centered System Design, Norman D.A. and Draper (eds.), Lawrence Erlbaum Associates, Hillsdale , NJ.

48. Norman, D.A. (1981). *" Design Rules Based on Analysis of Human Errors"* Communications of the ACM, Volume 26. p. 254-258.

49. Phillips, M.D., Mack, R.L., and Kellog, W A. (1988). *" Interface Metaphors and User Interface Design"* in Handbook of Human Computer Interaction, M Helander (ed.), Elsevier Science Publishers, Amsterdam, p 67-86

50. Polson, P.G. and Kieras, D E (1985) *" A Quantitative Model of the Learning and Performance of Text Editing Knowledge"* in CHI'85 Conference Proceedings on Human Factors in Computing Systems, p. 207-212.

51. Polson, P.G. and Lewis, H.L (1990). *" Theory-Based Design for Easily Learned Interface"* Human Computer Interaction, Volume 5, p. 191-220.

52. Reisner, P. (1981). *" Formal Grammar and Human Factors Design of an Interactive Graphics System"* IEEE Transactions on Software Engineering, SE-7(2), p. 229-240.

53. Reitman, J.S., Whitten, W.B., and Gruenenfelder, T.M (1985). *"A General User Interface for Entering and Changing Tree Structures, Nested Menus, and Decision Trees"* in Proceedings of NYU Symposium on User Interface Design, Norwood, NH.

54. Rieman, J., Davies, S., Hair, C., Esemplare, M., Polson, P.G., and Lewis, C. (1991). *" An Automated Cognitive Walkthrough"* in CHI'91 Conference Proceedings on Human Factors in Computing Systems, p. 427-428

55. Rubinstein, R. and Hersh, H. (1984). *" The Human Factor: Designing Computer Systems for People"* Digital Press, Burlington, MA.

56. Shackel, B. (1984). *" Information Technology-A Challenge to Ergonomics and Design"* Behavioral and Information Technology, Volume 3, p 263-275

57. Shneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley Publishing Company

58. Smelcer, J.B. (1989). *Understanding Human Errors in Database Query.* Unpublished Doctoral Dissertation, University of Michigan, Ann Arbor

59 Smith, S.L. and Mosier, J.N. (1986). " *Guidelines for Designing the User Interface Software*" Report 7, MTR-10090, ESD-TR-86-278 Mitre Corporation, Bedford, MA.

60. Young, R.M., Bernard, P., Simon, T., and Whittington, J. (1989). " *How Would Your Favorite User Model Cope With These Scenarios?*" SIGCHI Bulletin (20:4), p. 51-55.

61. Yourdon, E. (1989). *Structured Walkthroughs.* (4th ed.) Yourdon Press, Englewood Cliff, NJ.

APPENDIX A
CLASS LECTURE

AGENDA

- INTRODUCTION
- TOOLS AND METHODOLOGIES IN HUMAN COMPUTER INTERACTION (HCI)
- USER INTERFACE EVALUATION TECHNIQUES
- DESCRIPTION OF COGNITIVE MODELS
- CLASS EXERCISE TO MEASURE EASE OF USE AND EASE OF LEARNING

INTRODUCTION TO HCI

- THE WAY IT WAS!
  - ONLY SCIENTISTS AND ENGINEERS WERE INVOLVED WITH THE USE OF COMPUTERS
  - USING THE COMPUTER EFFICIENTLY WAS MORE INPORTANT THEN THE USER INTERFACE
- THE REALITIES OF TODAY!
  - THE USE OF COMPUTERS IS NOW WIDESPREAD AND MANY “NON-EXPERTS” USE INFORMATION TECHNOLOGY
  - SOFTWARE PRODUCTS ARE AIMED AT THE MASS MARKET AND THE EASE OF USE AND EASE OF LEARNING IS OFTEN A SUCCESS FACTOR (EX. APPLE)

INTRODUCTION TO HCI

- WHAT'S THE BIG DEAL ABOUT A GOOD INTERFACE
  - INCREASES THE EASE OF USE
  - REDUCES THE LEARNING TIME
  - REDUCES THE NUMBER OF ERRORS
  - INCREASES USER SATISFACTION
  - IMPROVES RETENTION OF FUNCTIONALITIES
- DIFFERENT TYPES OF INTERACTION
  - COMMAND LANGUAGE
  - MENU SYSTEM
  - DIRECT MANIPULATION

INTRODUCTION TO HCI

- DIFFICULTY OF DESIGNING A GOOD INTERFACE
  - DIFFICULT TO BE COMPATIBLE WITH THE USER'S MENTAL MODEL AND EXPECTATIONS
  - TIME CONSUMING AND EXPENSIVE
  - DIFFICULT TO MEASURE HUMAN FACTOR GOALS

DEVELOPMENT TOOLS AND METHODOLOGIES FOR INTERFACES

- USER TESTING
  - WHY IT'S NOT FEASIBLE:
    - » EXPENSIVE
    - » TIME CONSUMING
    - » DIFFICULT-RELIES ON INTUITION
- INTERFACE DESIGN GUIDELINES
  - PSYCHOLOGISTS CONDUCT BEHAVIORAL TESTS IN A LABORATORY
  - WHY IT'S OFTEN NOT ADEQUATE:
    - » TOO MANY GUIDELINES
    - » DEPENDENT ON CONTEXTUAL VARIABLES

DEVELOPMENT TOOLS AND METHODOLOGIES FOR INTERFACES

- DESIGN PRINCIPLES
  - EXPLAIN THE USER BEHAVIOUR ACCORDING TO PSYCHOLOGICAL MODELS
  - WHY IT'S OFTEN NOT ADEQUATE:
    - » PRINCIPLES ARE OFTEN TOO GENERAL(EX. SPEAK THE USER'S LANGUAGE)
- ITERATIVE DESIGN METHODOLOGIES
  - EXAMPLE OF DESIGN METHODOLOGIES
    - » EARLY FOCUS ON USERS
    - » IINTERGRATED DESIGN
    - » EARLY AN CONTINUAL USER TESTING
    - » ITERATIVE DESIGN
  - WHY IT'S DIFFICULT TO APPLY:
    - » EXPENSIVE, TIME CONSUMING, AND DIFFICULT

USER INTERFACE EVALUATION TECHNIQUES

- **HEURISTIC EVALUATION**
  - INFORMAL METHOD WHERE A NUMBER OF EVALUATORS MUST COMMENT ON AN INTERFACE AND DETERMINE IF IT MEETS CERTAIN PRE-SPECIFIED HUMAN FACTOR GOALS
  - WHY IT'S OFFTEN NOT ADEQUATE:
    - » COMPLETLY SUBJECTIVE
    - » STUDIES DEMONSTRATE THAT IT IS NOT ALWAYS ACCURATE
- **SOFTWARE GUIDELINES**
  - EVALUATE THE INTERFACE ACCORDING TO A SET OF GUIDELINES
  - WHY IT'S OFTEN NOT ADEQUATE:
    - » GUIDELINES ARE CUMBERSOME TO USE
    - » CONTEXTUALLY CONSTRAINED

USER INTERFACE EVALUATION TECHNIQUES

- USER TESTING
  - CONDUCTING ACTUAL USER TESTING
  - WHAT'S THE PROBLEM WITH THIS:
    - » EXPENSIVE , TIME CONSUMING, AND DIFFICULT
- TASK-THEORETIC ANALYTIC MODELS
  - MODELS THE HUMAN BEHAVIOUR AND PROVIDES QUANTITATIVE ANALYSIS OF INTERFACES. MEASURES PARAMETERS SUCH AS EASE OF USE AND EASE OF LEARNING
    - » EXAMPLE-THE TASK ERASE (DELETE, BLOCK, BACKSPACE)
  - WHY THEY ARE OFTEN NOT USED:
    - » DIFFICULT TO APPLY
    - » CAN ONLY EVALUATE CERTAIN HUMAN FACTORS

THE GOMS MODEL

- GOALS, OPERATORS, METHODS, AND SELECTION RULES (GOMS)
  - GOALS-GOALS FOR THE TASK
  - OPERATORS-OVERT OPERATORS (KEYPRESSES), INTERNAL OPERATORS (MEMORY RETRIEVAL)
  - METHODS-SERIES OF OPERATORS FOR ACHIEVING THE GOAL
  - SELECTION RULES-CHOOSING AMONG COMPETING METHODS TO ACHIEVE THE SAME GOAL
- A GOMS ANALYSIS IS A FORMAL AND STRUCTURES WAY TO DESCRIBE THE KNOWLEDGE THAT A USER NEEDS TO PERFORM A SPECIFIC TASK ON A DEVICE

THE GOMS MODEL

- DESCRIBING THE GOALS, OPERATORS, METHODS, AND SELECTION RULES IN A FORMAL WAY IS THE GOAL OF PERFORMING A GOMS ANALYSIS
- ONCE THE GOMS MODEL HAS BEEN DEVELOPED PREDICTION ABOUT HUMAN FACTORS SUCH EASE OF USE AND EASE OF LEARNING CAN BE DERIVED
- EXAMPLE:
  - GOAL: ERASE A WORD
  - OPERATORS: KEYPRESSES, POINTING, MENTAL (MEMORY RETRIEVAL)
  - METHOD: BACKSPACE, DELETE, BLOC
  - SELECTION RULES: DEPENDS ON WHAT NEEDS TO BE DELETED AND ON THE POSITION OF THE CURSOR

KEYSTROKE-LEVEL MODEL

- THE MODEL PROVIDES THE INTERFACE DESIGNER WITH THE CAPABILITY TO PREDICT HOW MUCH TIME A USER WOULD TAKE TO ACCOMPLISH A GIVEN TASK WITH A GIVEN INTERACTIVE COMPUTER SYSTEM
  - GIVEN THE FOLLOWING
    - » THE TASK (MAY INVOLVE SEVERAL SUBTASKS)
    - » THE COMMAND LANGUAGE OF A SYSTEM
    - » THE MOTOR SKILLS PARAMETERS OF THE USER
    - » THE RESPONSE TIME PARAMETERS OF THE SYSTEM
    - » THE METHOD USED TO PERFORM THE TASK
  - THE MODEL PREDICTS
    - » THE TIME AN EXPERT USER WILL TAKE TO EXECUTE THE TASK WITHOUT ANY ERROR

KEYSTROKE-LEVEL MODEL

- HOW THE TASK IS DECOMPOSED

– KEYSTROKE (K)	.08 SECONDS
– MENTAL OPERATOR (M)	1.35 SECONDS
– POINTING WITH A MOUSE	1.10 SECONDS
– MOVING HAND TO STARTING POSITION	.40 SECONDS

- TIME TO EXECUTE A TASK
  - EXECUTE= K + P + H + M

APPLICATION KEYSTROKE LEVEL MODEL

- EXAMPLE 1
  - @ SUM (12345678 : 654321) < RET >

    MK MK K K MKMKKKKKKKK MKMKKKKKK MK MK

    » ADDS UP TO: 8 M'S AND 22 K'S

    » 8(1.35) + 22 (.08) = 12.56 SECONDS
- HEURISTICS FOR "M" OPERATORS
  - PLACE M'S IN FRONT OF ALL P'S THAT SELECT COMMANDS (NOT ARGUMENTS) [@ => MK]
  - IF AN OPERATOR FOLLOWING AN M IS FULLY ANTICIPATED IN AN OPERATOR JUST PREVIOUS TO M, THEN DELETE THE M [PMK => PK]
  - IF A STRING OF MK'S BELONGS TO A COGNITIVE UNIT (EX. THE NAME OF A COMMAND), THEN DELETE ALL M'S BUT THE FIRST [SUM => MKKK...]

KEYSTROKE-LEVEL EXERCISE

- EXERCISE 1
  - TASK => UNDERLINE THE WORD "EASY" (ASSUME THE CURSOR IS AT THE BEGINNING OF THE WORD)
  - PROCEDURE
    - » PRESS ALT
    - » PRESS F4
    - » PRESS => TO END OF THE WORD EASY
    - » PRESS F8

SOLUTION-EXERCISE 1

1) PRESS ALT	MK
2) PRESS F4	MK
3) PRESS => TO END OF WORD	M, 4K
4) PRESS F8	MK

$$\begin{aligned} T_{\text{EXECUTE}} &= 4M\text{'S} + 8K\text{'S} \\ &= 4(1.35) + 7(.08) \\ &= \underline{5.96 \text{ SECONDS}} \end{aligned}$$

KEYSTROKE-LEVEL EXERCISE

- EXERCISE 2
  - TASK => MOVE THE FIRST SENTENCE TO THE END OF THE ARGUMENT (ASSUME THE CURSOR IS AT THE BEGINNING OF THE SENTENCE)
  - SENTENCE
    - » THIS IS AN EXERCISE TO TEACH HOW TO APPLY THE KEYSRTOKE-LEVEL MODEL. DON'T WORRY THE NEXT EXAMPLE WILL BE MORE CHALLENGING
  - PROCEDURE
    - » PRESS ALT
    - » PRESS F4
    - » PRESS => TO THE END OF THE FIRST SENTENCE
    - » PRESS CTLR
    - » PRESS F4
    - » PRESS MENU CHOICE 1 (BLOCK)
    - » PRESS MENU CHOICE 2 (MOVE)
    - » PRESS => TO THE END OF THE SECOND SENTENCE
    - » PRESS RETURN

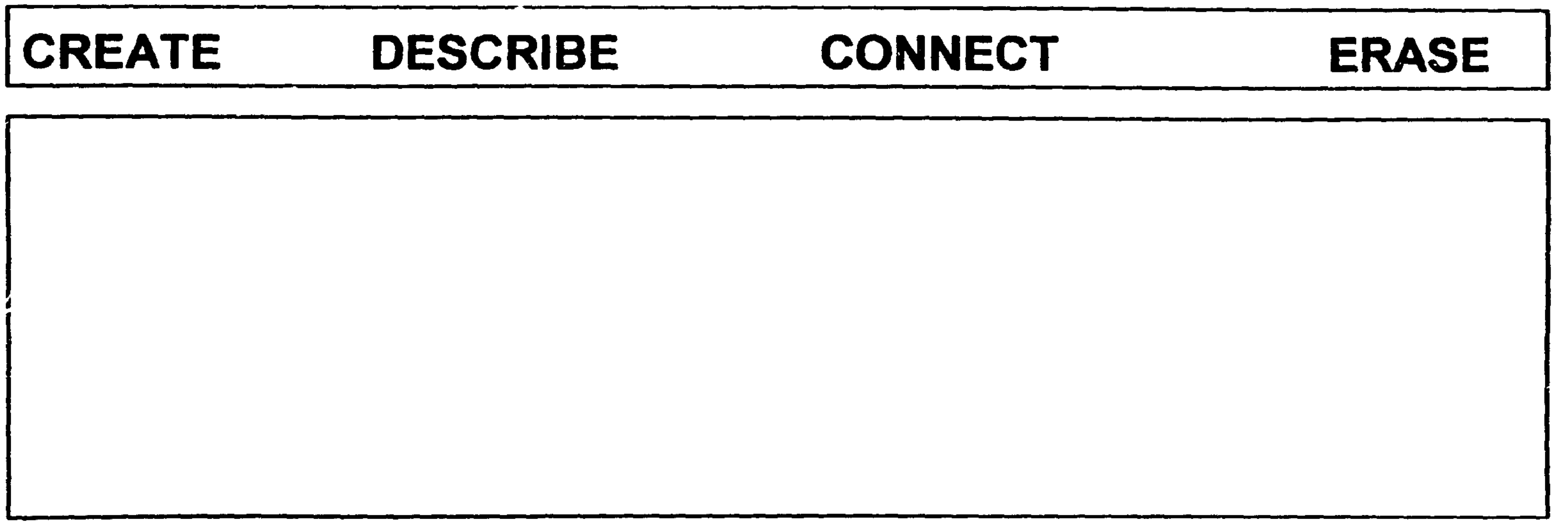
SOLUTION-EXERCISE 2

1) PRESS ALT	MK
2) PRESS F4	MK
3) PRESS => TO END OF SENTENCE	68K, M
4) PRESS CLTR	MK
5) PRESS F4	MK
6) PRESS MENU CHOICE 1	MK
7) PRESS MENU CHOICE 2	MK
8) PRESS => TO END OF SENTENCE	54K, M
9) PRESS <RETURN>	MK

$T_{EXECUTE}$ = 9M'S + 129K = 22.47 SECONDS

KEYSTROKE-LEVEL EXERCISE

- TO PERFORM A FUNCTION YOU POINT TO THE MENU ITEM AND CLICK ANY BUTTON (EX. "CREATE"). YOU THEN POINT TO THE LOCATION WHERE YOU WANT THE DOCUMENT "CREATED". ONCE YOU ARE IN THE "CREATE" MODE YOU DO NOT NEED TO GO BACK.



EXERCISE 3 => "CREATE" TWO DOCUMENTS

EXERCISE 4 => "CONNECT" TWO DOCUMENTS

SOLUTION EXERCISE 3

1) SELECT “CREATE”	
REACH FOR THE PUCK	H
POINT TO CREATE	MP
PRESS ANY BUTTON ON MOUSE	K
2) “CREATE” DOCUMENT 1	
POINT TO LOCATION	MP
PRESS ANU BUTTON ON MOUSE	K
3) “CREATE” DOCUMENT 2	
POINT TO LOCATION	MP
PRESS ANY BUTTON ON MOUSE	K

$T_{EXECUTE} = H + 3M + 3P + 3K = 7.99$ SECONDS

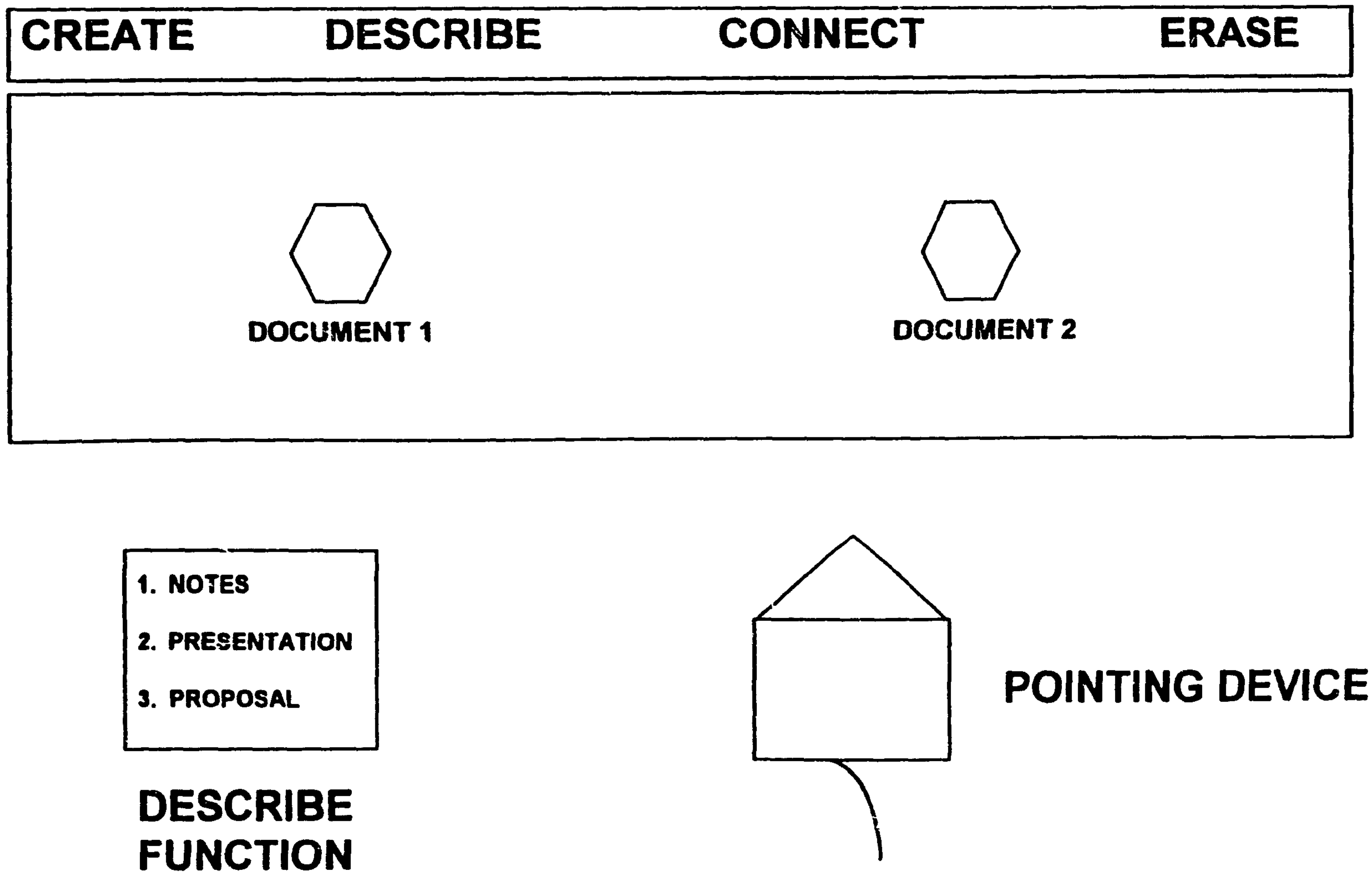
SOLUTION EXERCISE 4

1) **SELECT "CONNECT"**	H
REACH FOR THE MOUSE	MP
POINT TO "CONNECT"	K
PRESS ANY BUTTON ON MOUSE	
2) **SELECT DOCUMENT 1**	MP
POINT TO DOCUMENT	K
PRESS ANY BUTTON ON MOUSE	
3) **SELECT DOCUMENT 2**	MP
POINT TO DOCUMENT 2	K
PRESS ANY BUTTON ON MOUSE	

$$T_{EXECUTE} = H + 3M + 3P + 3K = 7.99 \text{ SECONDS}$$

KEYSTROKE-LEVEL MODEL

- EXERCISE 5: DESCRIBE THE TWO DOCUMENTS
- EXERCISE 6: ERASE DOCUMENT 2



SOLUTION EXERCISE 5

1) SELECT "DESCRIBE"	
REACH FOR THE MOUSE	H
POINT TO "DESCRIBE"	MP
PRESS ANY BUTTON ON MOUSE	K
2) SELECT DOCUMENT 1	
POINT TO DOCUMENT 1	MP
PRESS ANY BUTTON ON MOUSE	K
3) "DESCRIBE" DOCUMENT 1	
POINT TO THE APPROPRIATE CHOICE	MP
PRESS ANY BUTTON ON MOUSE	K
4) SELECT DOCUMENT 2	
POINT TO DOCUMENT 2	MP
PRESS ANY BUTTON ON MOUSE	K
5) "DESCRIBE" DOCUMENT 2	
POINT TO THE APPROPRIATE CHOICE	MP
PRESS ANY BUTTON ON MOUSE	K

$T_{\text{EXECUTE}} = H + 5M + 5K + 5P = \underline{13.05 \text{ SECONDS}}$

SOLUTION EXERCISE 6

1) SELECT “ERASE”	
REACH FOR THE MOUSE	H
POINT TO “ERASE”	MP
PRESS ANY BUTTON ON MOUSE	K
2) “ERASE” DOCUMENT 2	
POINT TO THE LOCATION OF DOCUMENT 2	MP
PRESS ANY BUTTON ON THE MOUSE	K

$T_{EXECUTE} = H + 2M + 2P + 2K =$ 5.46 SECONDS

TAMPEL

- THE MODEL PROVIDES THE INTERFACE DESIGNER WITH THE CAPABILITY TO PREDICT HOW MUCH TIME A USER WOULD TAKE TO LEARN HOW TO USE A SYSTEM EFFICIENTLY WITH A GIVEN COMPUTER INTERFACE
- EFFICIENTLY MEANS
  - THE DIFFERENT TASKS THAT CAN BE PERFORMED WITH THE SOFTWARE, THE PROCEDURES THAT COULD BE USED TO PERFORM THESE TASKS, AND THE FUNCTIONALITY OF EACH PROCEDURES
  - THE USER KNOWS HOW TO EXECUTE EACH PROCEDURES
  - THE CONDITIONS UNDER WHICH EACH PROCEDURES APPLY, AND AN EFFICIENT METHOD FOR COMBINING DIFFERENT PROCEDURES TO ACCOMPLISH A CERTAIN TASK

TAMPEL

- WITH TAMPEL THE LEARNING TIME IS A FUNCTION OF THE COGNITIVE SKILLS THAT ARE ASSOCIATED WITH LEARNING THE SYSTEM. THESE COGNITIVE SKILLS ARE:
  - COGNITIVE SKILLS OF TYPE “PROCEDURES” ARE DEFIINED AS THOSE THAT INVOLVE KNOWING THE AVAILABLE PROCEDURES (OR OPERATORS), THEIR FUNCTIONALITY AND THE CONDITIONS UNDER WHICH THEY APPLY
  - COGNITIVE SKILLS OF TYPE “RULES” ARE THOSE THAT INVOLVE KNOWING ADDITIONAL CONDITIONS FOR EXECUTING PROCEDURES AND OPERATORS IN DIFFERENT SITUATIONS

TAMPEL

- A GIVEN COGNITIVE SKILL CAN BE CATEGORIZED AS EITHER A RECALL TASK OR A RECOGNITION TASK, DEPENDING ON WETHER IT INVOLVES RECALL MEMORY OR RECOGNITION MEMORY
- A GOMS FORMALISM IS USED TO DECOMPOSE THE DIFFERENT FUNCTIONS OF A TASK OF A GIVEN SYSTEM

TAMPEL

- THE COGNITIVE LEARNING TIME (CLT) IS A FUNCTION OF THE NUMBER OF PROCEDURES THAT MUST BE RECALLED (PROC_RECL), THE NUMBER OF PROCEDURES THAT MUST ME RECOGNIZED (PROC_REGZ), AND THE NUMBER OF ADDITIONAL RULES (ADD_RULES = RULES-PROCEDURES)

CLT = -58.15 + 43.63 PROC_RECL + 25.53
25.53 PROC_REGZ + 5.99 ADD_RULE

- YOU USE THE GOMS FORMALISM TO DETERMINE THE NUMBER OF PROC_RECL, PROC_REGZ, AND RULE_RECL

RULES TO APPLY TAMPEL

- DECOMPOSE THE TASK USING THE GOMS FORMALISM
- WRITE DOWN IF THE STATEMENTS ARE "RULE_RECL", "PROC_RECL", OR "PROC_REGZ"
- DETERMINE WHICH PROCEDURES ARE TRANSFERED
- THE "ADDITIONAL RULES" ARE THE TOTAL NUMBER OF RULES MINUS THE TOTAL NUMBER OF PROCEDURES
- CLT = -58.15 + 43.63 PROC_RECL + 25.53
  25.53 PROC_REGZ + 5.99 ADD_RULE
- IN THE CASE OF DIRECT MANIPULATION SYSTEMS DO NOT FORGET TO COUNT THE "SELECT (OBJECT)" PROCEDURE ALL THE TIME IT OCCURS

TAMPEL

- **EXERCISE 1**

  THIS IS THE SAME FICTITIOUS SYSTEM THAT WAS USED IN THE LAST SECTION TO MEASURE THE EASE OF USE. THIS SYSTEM WORKS WITH A MOUSE AND THE MENU ITEMS CAN BE ACTIVATED BY POINTING THE CURSOR TO THE ITEM AND CLICKING ANY BUTTON. USING GOMS FORMALISM AND TAMPEL MEASURES THE TIME IT WOULD TAKE TO LEARN THIS SYSTEM

CREATE	CONNECT

SOLUTION EXERCISE 1

```
GOAL:  ACQUIRE  UNIT TASK
IF (MORE DOCUMENT NEED TO BE ADDED) THEN                RULE_RECL
        UNIT TASK = CREATE (LOCATION)                   PROC_REGZ
ELSEIF (DOCUMENT NEED TO BE CONNECTED)                  RULE_RECL
        UNIT TASK = CONNECT (DOCUMENT X, DOCUMENT Y)    PROC_REGZ


GOAL:  EXECUTE-CREATE (LOCATION)
IF ("CREATE" IS NOT SELECTED) THEN                      RULE_RECL
        UNIT TASK = SELECT "CREATE"                     PROC_RECL
ELSEIF ("CREATE" IS SELECTED) THEN                      RULE_RECL
        UNIT TASK = SELECT (LOCATION)                   PROC_RECL

GOAL:  EXECUTE- CONNECT (DOCUMENTX, DOCUMENT Y)
IF ("CONNECT" IS NOT SELECTED) THEN                     RULE_RECL
        UNIT TASK = SELECT ("CONNECT")                  TRANSFER
ELSEIF ("CONNECT" IS SELECTED) THEN                     RULE_RECL
        UNIT TASK = SELECT (DOCUMENT X)                 PROC_RECL
ELSEIF (DOCUMENT X IS SELECTED) THEN                    RULE_RECL
        UNIT TASK = SELECT (DOCUMENT Y)                 TRANSFER

GOAL:  EXECUTE SELECT (OBJECT)
IF (CURSOR IS NOT POINTING OBJECT) THEN                 RULE_RECL
        UNIT TASK = POINT TO (OBJECT)                   PROC_RECL
ELSEIF (CURSOR IS POINTING TO OBJECT) THEN              RULE_RECL
        UNIT TASK = PRESS (ANY BUTTON)                  PROC_RECL
```

SOLUTION EXERCISE 1

RULE_RECALL		9
PROCEDURE_RECALL		5
PROCEDURE_RECOGNITION		2
ADDITIONAL_RULES	9-7 =	2

CLT = -58.15 + 43.63 PROC_RECL +
25.53 PROC_REGZ + 5.99 ADD_RULE

CLT = -58.15 + 43.63 (5) + 25.53(2) + 5.99(2)
= <u>223.96 SECONDS</u>

NOTE

THE PROCEDURES ARE DEFINED AS "SIMILAR ENOUGH" IF THEY INVOLVE THE EXECUTION OF THE SAME SEQUENCE OF OPERATORS (SAME RESPONSES), IN ORDER TO ACHIEVE SIMILAR GOALS (SIMILAR STIMULI)-GOALS THAT CAN BE GENERALIZED TO THE SAME GOAL. THERE IS A COMPLETE TRANSFER OF LEARNING BETWEEN PROCEDURES THAT EITHER IDENTICAL OR SIMILAR ENOUGH. THESE "SIMILAR" PROCEDURES ARE TRANSFERRED COGNITIVE SKILLS AND THEY DO NOT ADD TO THE TOTAL LEARNING TIME

TAMPEL

- **EXERCISE 2**

CREATE	CONNECT

1. NOTES
2. PRESENTATION
3. PROPOSAL

DESCRIBE FUNCTION

ESTIMATE THE TIME IT WOULD TAKE TO LEARN HOW USE THIS TWO FUNCTION SYSTEM BY DECOMPOSING THE TASKS USING THE GOMS FORMALISM AND ESTIMATING THE TIME TO LEARN BY USING TAMPEL

SOLUTION EXERCISE 2

```
GOAL:  ACQUIRE  UNIT TASK
IF (MORE DOCUMENT NEED TO BE ADDED) THEN              RULE_RECL
        UNIT TASK = CREATE (LOCATION)                 PROC_REGZ
ELSEIF (DOCUMENT NEED TO BE DESCRIBED)                RULE_RECL
        UNIT TASK = DESCRIBE (DOCUMENT                PROC_REGZ

GOAL:  EXECUTE-CREATE (DOCUMENT)
IF ("CREATE" IS NOT SELECTED) THEN                    RULE_RECL
        UNIT TASK = SELECT "CREATE"                   PROC_RECL
ELSEIF ("CREATE" IS SELECTED) THEN                    RULE_RECL
        UNIT TASK = SELECT (LOCATION)                 PROC_RECL

GOAL:  EXECUTE- DESCRIBE (DOCUMENT)
IF ("DESCRIBE" IS NOT SELECTED) THEN                  RULE_RECL
        UNIT TASK = SELECT ("DESCRIBE")               TRANSFER
ELSEIF ("DESCRIBE" IS SELECTED) THEN                  RULE_RECL
        UNIT TASK = SELECT (DOCUMENT   )              PROC_RECL
ELSEIF (DOCUMENT IS SELECTED) THEN                    RULE_RECL
        UNIT TASK = SELECT (CHOICE 1,2, 3)            PROC_REGZ

GOAL:  EXECUTE SELECT (OBJECT)
IF (CURSOR IS NOT POINTING OBJECT) THEN               RULE_RECL
        UNIT TASK = POINT TO (OBJECT)                 PROC_RECL
ELSEIF (CURSOR IS POINTING TO OBJECT) THEN            RULE_RECL
        UNIT TASK = PRESS (ANY BUTTON)                PROC_RECL
```

SOLUTION EXERCISE 2

RULE_RECALL		9
PROCEDURE_RECALL		5
PROCEDURE_RECOGNITION		3
ADDITIONAL_RULES	9-8 =	1

CLT = -58.15 + 43.63 PROC_RECL +
    25.53 PROC_REGZ + 5.99 ADD_RULE

CLT = -58.15 + 43.63 (5) + 25.53(3) + 5.99(1)
    = <u>242.58 SECONDS</u>

TAMPEL

- **EXERCISE 3**

CREATE	DESCRIBE	CONNECT

ESTIMATE THE TIME IT WOULD TAKE TO LEARN HOW USE THIS THREE FUNCTION SYSTEM BY DECOMPOSING THE TASKS USING THE GOMS FORMALISM AND ESTIMATING THE TIME TO LEARN BY USING TAMPEL

SOLUTION TO EXERCISE 3

```
GOAL:  ACQUIRE  UNIT TASK
IF (MORE DOCUMENT NEED TO BE ADDED) THEN           RULE_RECL
        UNIT TASK = CREATE (DOCUMENT)              PROC_REGZ
ELSEIF (DOCUMENT NEED TO BE DESCRIBE)              RULE_RECL
        UNIT TASK = DESCRIBE (DOCUMENT             PROC_REGZ
ELSEIF (DOCUMENTS ARE TO BE DELETED)               RULE_RECL
        UNIT TASK = ERASE (DOCUMENT)               PROC_REGZ


OAL:  EXECUTE-CREATE (DOCUMENT)
IF ("CREATE" IS NOT SELECTED) THEN                 RULE_RECL
        UNIT TASK = SELECT "CREATE"                PROC_RECL
ELSEIF ("CREATE" IS SELECTED) THEN                 RULE_RECL
        UNIT TASK = SELECT (LOCATION)              PROC_RECL

GOAL:  EXECUTE- DESCRIBE (DOCUMENT)
IF ("DESCRIBE" IS NOT SELECTED) THEN               RULE_RECL
        UNIT TASK = SELECT ("DESCRIBE")            TRANSFER
ELSEIF ("DESCRIBE" IS SELECTED) THEN               RULE_RECL
        UNIT TASK = SELECT (DOCUMENT        )      PROC_RECL
ELSEIF (DOCUMENT IS SELECTED) THEN                 RULE_RECL
        UNIT TASK = SELECT (CHOICE 1,2, 3)         PROC_REGZ

GOAL:  EXECUTE-ERASE (DOCUMENT)
IF ("ERASE" IS NOT SELECTED) THEN                  RULE_RECL
        UNIT TASK = SELECT "ERASE"                 TRANSFER
ELSEIF ("ERASE" IS SELECTED) THEN                  RULE_RECL
        UNIT TASK = SELECT (LOCATION)              TRANSFER

GOAL:  EXECUTE SELECT (OBJECT)
IF (CURSOR IS NOT POINTING OBJECT) THEN            RULE_RECL
        UNIT TASK = POINT TO (OBJECT)              PROC_RECL
ELSEIF (CURSOR IS POINTING TO OBJECT) THEN         RULE_RECL
        UNIT TASK = PRESS (ANY BUTTON)             PROC_RECL
```

SOLUTION EXERCISE 3

RULE_RECALL		12
PROCEDURE_RECALL		4
PROCEDURE_RECOGNITION		4
ADDITIONAL_RULES	12-8 =	4

CLT = -58.15 + 43.63 PROC_RECL +
25.53 PROC_REGZ + 5.99 ADD_RULE

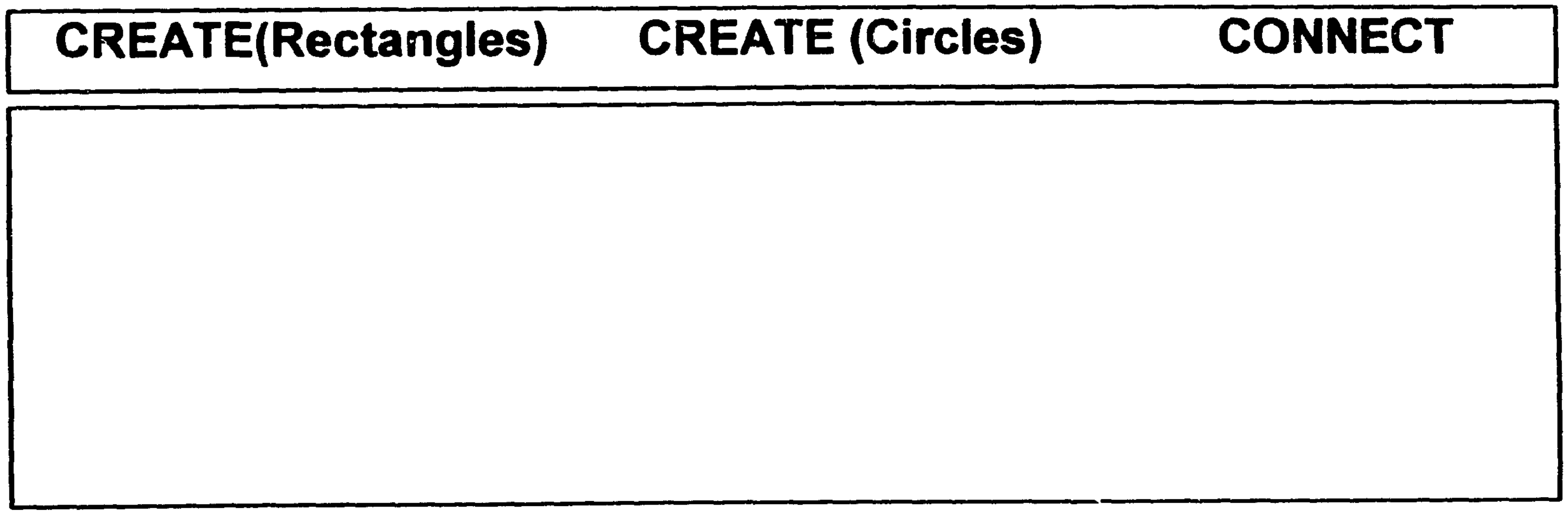
CLT = -58.15 + 43.63 (4) + 25.53(4) + 5.99(4)
= <u>242.45 SECONDS</u>

APPENDIX B
CLASS QUIZ

CLASS QUIZ

FULL NAME________________________

This system provides a way to display "rectangles" and "circles" and connect them. the system has four functions:
CREATE Rectangle
CREATE Circle
CONNECT Figures (rectangles and circles)



The figure above displays the interface used to operate the system. A one button mouse is used as a pointing device. The menu items can be accessed by pointing the cursor to the appropriate item and pressing the button on the mouse once. Once a user has selected a function such as "CREATE" (rectangle) he/she can move to the area in the grid where the figure is to be positioned and press the button on the mouse to "CREATE" the figure. Once the usser is in the "CREATE" mode he/she can create as many figures as is necessary without having to go back to the menu. This function is da-activated when another function is selected.

Measure the ease of use (the time to perform the task) of "CREATING" a rectangle and a circle and "CONNECTING" both figures together. Also, measure the ease of learning (time to learn) of the three functions of the system by decomposing the tasks using the GOMS formalism and applying TAMPEL to measure the time to learn the system. GOOD LUCK!

APPENDIX C
ACTUAL TEST

EVALUATION OF INTERFACES IN TERMS OF HUMAN FACTORS GOALS

INSTRUCTIONS

Bases on the lecture that was given in class on Human Computer Interaction you will be asked to measure Ease of Use and Ease of Learning for different interfaces. The following section describes an interactive network editing system with different alternative interface design. You must measure Ease of Use and Ease of Learning for two tasks based on the Keystroke-level Model and on TAMPEL. You may use the notes that were provided in class to help you evaluate the two tasks. You will be measured on how fast you evaluate each tasks and how accurate your predictions are. Good Luck!

EVALUATION OF INTERFACES IN TERMS OF HUMAN FACTORS GOALS

INSTRUCTIONS

Bases on the lecture that was given in class on Human Computer Interaction you will be asked to measure Ease of Use and Ease of Learning for different interfaces The following section describes an interactive network editing system with different alternative interface design. You must measure Ease of Use and Ease of Learning for two tasks based on the Keystroke-level Model and on TAMPEL. It is imperative that you clearly identify if the tasks and subtasks are recall, recognition, or transfer, and if the operators involve pointing (P), keystroking (K), homing (H), and/or mental operators (M). You may use the notes that were provided in class to help you evaluate the two tasks You will be measured on how fast you evaluate each tasks and how accurate your predictions are Good Luck!

DESCRIPTION OF TASK

In this study, we examine possible designs for the user interface of an interactive computer graphics system for editing networks, where a network is defined to be a collection of nodes connected by edges (Figure 1). There are two basic tasks that the user can perform with the system:

ADD/DELETE a node.
ADD/DELETE an edge

All the commands of the system are listed on a menu that is continuously displayed on the screen along with a grid containing the network being edited and a graphics tablet with a puck is used as a pointing device (Figure 2) To select an object (a node or an edge) or an operation (one of the menu selection) the user points to it with the on-screen cursor by moving the puck on the tablet, and then presses one of the buttons on the puck The user is allowed to perform a network editing operation as many times as he/she wants without having to go back to the menu. For example, the user can add several nodes to the network by selecting the menu item associated with that operation only once, and then every time he/she wants to add a node he/she just has to point to the location where the node is to be added and then press any button on the puck. This reduces hand movement considerably, making the system more efficient and enjoyable to use. The following section provides a description of two interfaces and information on how to perform a task using both system.

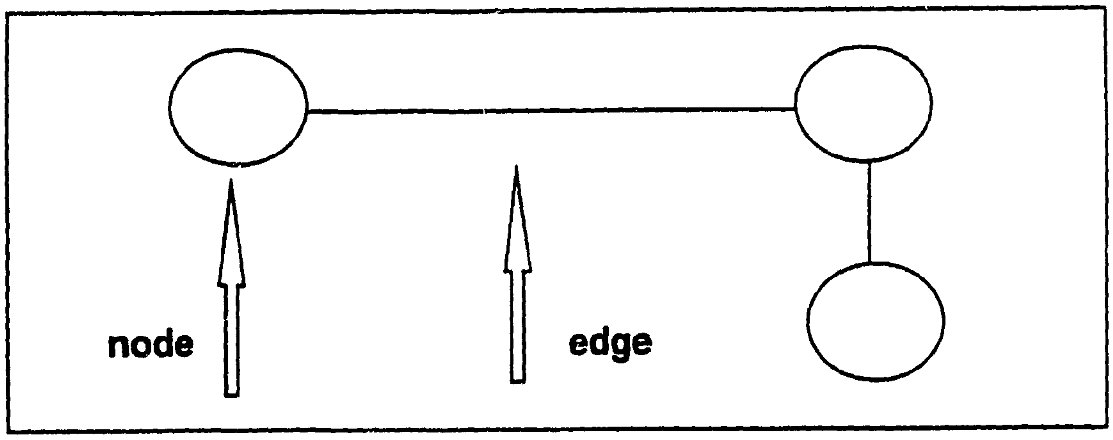


Figure 1: A Network

1. **Interface**

The menu choices of this interface (Figure 3) are ADD-NODE, ADD-EDGE, DELETE-EDGE, and DELETE-NODE This section provides a description of how each of these features can be performed With this interface the user must add-in one node at a time, and then add edges to connect them. A node or an edge can be deleted at any time

To create nodes and to connect them the user must first select the menu item "ADD-NODE", move the puck on the tablet so that the on-screen cursor is positioned where the new node is to be added, and then press any button on the puck to add the node to the diagram. To connect the node with edges, the user must first select the menu item "ADD-EDGE", select the first node to be connected (by pointing to the node and pressing any button on the puck), and then similarly select the second node An edge connecting the two nodes will then appear on the screen

To delete a node or an edge the user must select the menu item "DELETE-NODE" or "DELETE-EDGE" and move the puck on the tablet so that the on-screen cursor is positioned on the node or edge that needs to be deleted. Menu items only need to be selected once For example "ADD-NODE" is selected only once, there is no need to select it every time a node is to be added, unless it is de-selected (by selecting another menu item, ("ADD-EDGE" for instance).

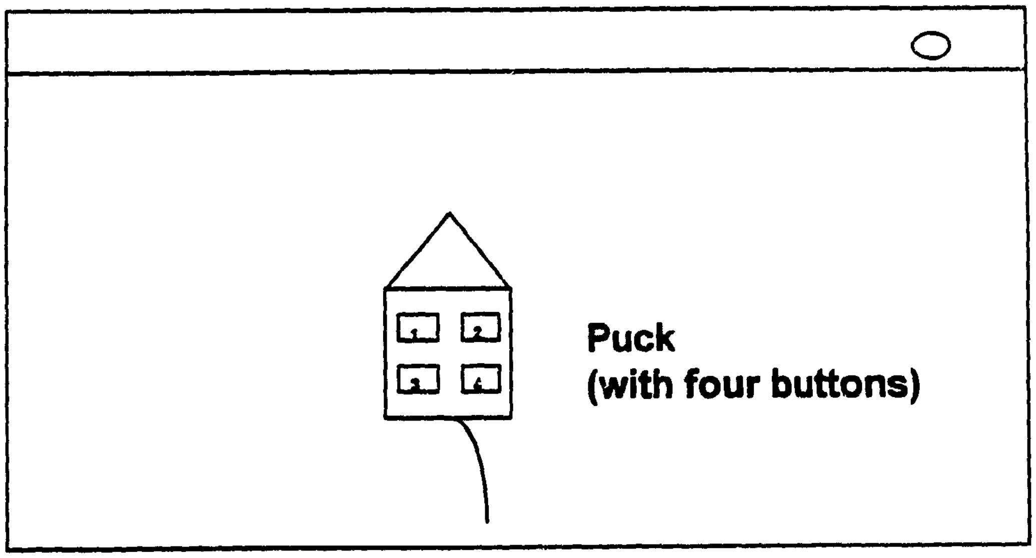


Figure 2: Tablet with Puck

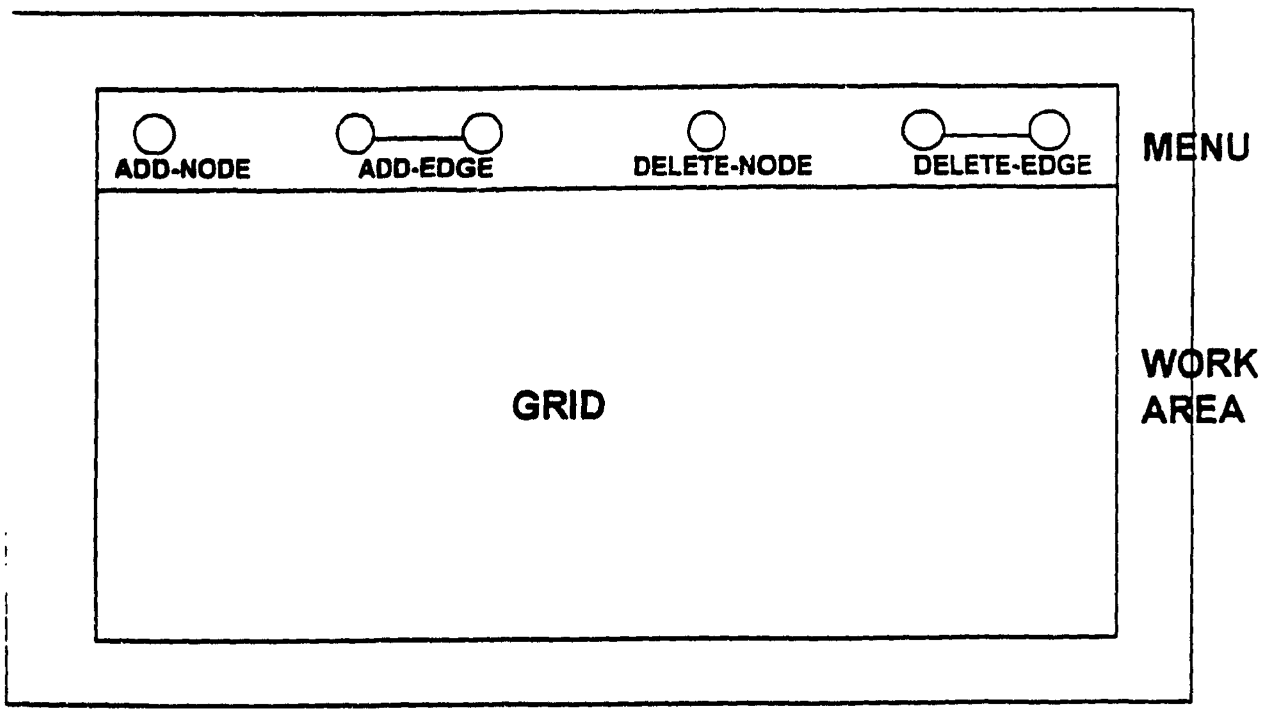
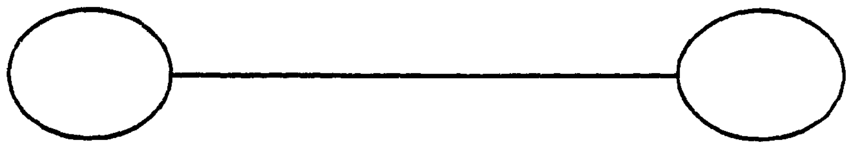


Figure 3: Interface 1

A. EXERCISE 1: Delete two nodes and one edge

This task consists of deleting two nodes connected by an edge. Assume that the user will use the most efficient method which is to delete the two nodes first and then delete the edge.

Given the interface above, the task, and the method for performing the task, predict the ease of use by applying the Keystroke-level Model.



B. EXERCISE 2: Measuring the time to learn two functions

This exercise consists of evaluating the time it takes to learn a system such as the one described. This system, however, only has the function "DELETE-NODE" and "DELETE-EDGE". Using TAMPEL measure the ease of learning

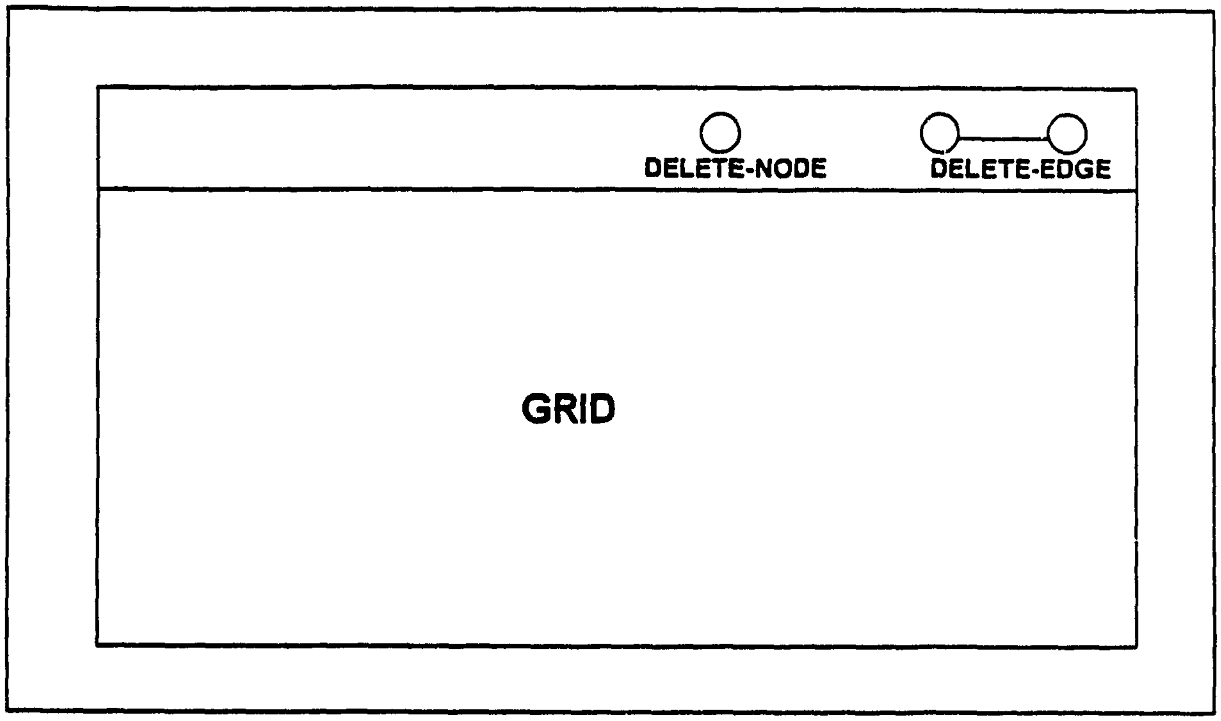


Figure 4: Interface 2

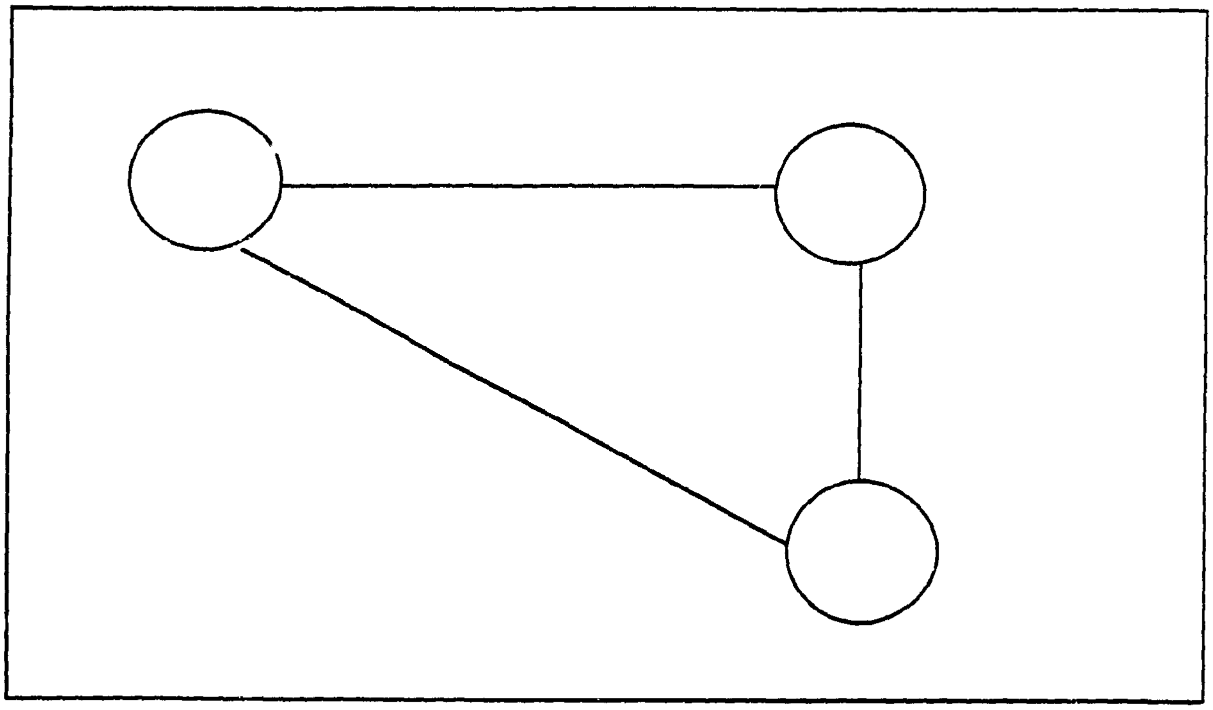
C EXERCISE 3· Create three nodes and connect them

This task consists of creating three nodes and connecting them together Again, we assume that the user will use the most efficient method which consists of following two rules:

1. Create all the nodes first. Since the interface requires two different commands to add and connect nodes (ADD-NODE and ADD-EDGE), it is more efficient to lay out all the nodes first and then connect them by edges. This way the user avoids selecting the menu items "ADD-NODE" and "ADD-EDGE" more then once and hence reduce the physical and cognitive actions needed to change between the two

2 Always move on the the nearest node when adding or connecting nodes this reduces gesture considerably

Using the Keystroke-level Model evaluate the ease of use of performing this task



D EXERCISE 4: Measuring the time to learn four functions

This exercise consists of evaluating the time it takes to learn a system such as the one described. Using TAMPEL measure the ease of learning

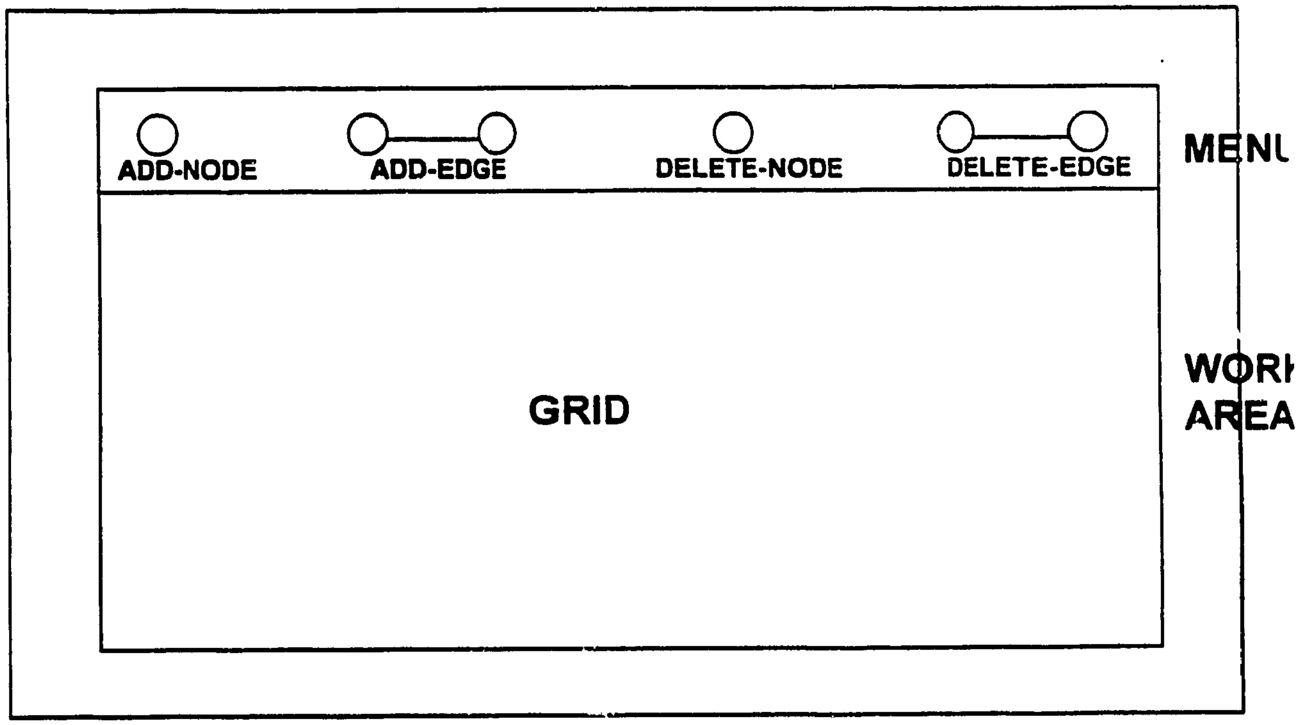


Figure 3: Interface 1

APPENDIX D
BACKGROUND QUESTIONNAIRE

BACKGROUND INFORMATION

1) Full Name:

2) Years of University Education:

3) University program presently enrolled in:
Major:
Minor:

4) How many University level MIS or Computer Science courses have you taken so far:

5) How many years, if any, of formal MIS work experience do you have

6) Using the following scale indicate your agreement with the following statements

1	2	3	4	5	6	7
Very Strongly Agree	Strongly Agree	Agree	Neither Agree Nor Disagree	Disagree	Strongly Disagree	Very Strongly Disagree

* I am knowledgeable with the use of operating systems. (___)

* I am knowledgeable with the use of computer hardware (___)

* I am knowledgeable in defining information requirements (___)

* I am knowledgeable at designing outputs and inputs for application software (___)

* I am knowledgeable at developing file structures (___)

* I am knowledgeable in the use of office automation products such as Wordperfect and LOTUS (___)

APPENDIX E
RESULT OF BACKGROUND QUESTIONNAIRE

BACKGROUNFD INFORMATION

YEARS OF UNIVERSITY EDUCATION

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
YEARS	MANUAL	13	3.154	3.000	0.376	3.000	4 000	(2 927, 3 381)
YEARS	IDES	14	3.357	3.000	1 008	2.000	5 000	(2 775 3 939)

UNIVERSITY M.I.S. COURSES

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
# OF COURSES	MANUAL	13	8.000	9.000	1.780	5.000	10 000	(6 94, 9 076)
# OF COURSES	IDES	14	6.929	7.000	2.303	2.000	11 000	(5 599, 8 258)

FORMAL M.I.S. WORK EXPERIENCE

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
YEARS	MANUAL	13	0.0218	0.000	0.381	0.000	1.000	(-0 013, 0 448)
YEARS	IDES	14	0 0950	0	0.2749	0 000	1 000	(-0 0638, 0 2538)

KNOWLEDGEABLE WITH THE FOLLOWING

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C.I.
OPER. SYSTEM	MANUAL	13	3.385	4.000	1.502	1.000	6.000	(2.477, 4.293)
OPER. .SYSTEM	IDES	14	3.357	3.000	1.646	1.000	6 000	(2.407, 4.308)
COMP. HARDWARE	MANUAL	13	3.308	3.000	1.377	1.000	6.000	(2 475, 4.140)
COMP. HARDWARE	IDES	14	3.429	3.000	1.505	1.000	6.000	(2 560, 4.298)
INFO. REQUIRE.	MANUAL	13	2.923	3.000	1.256	1.000	5.000	(2.164, 3.682)
INFO. REQUIRE.	IDES	14	3.214	3.000	1.051	2.000	6.000	(2.607, 3 821)
OUTPUT/INPUT	MANUAL	13	2.692	3.000	1.032	1.000	4 000	(2 069, 3.316)
OUTPUT/INPUT	IDES	14	3.071	3.000	1.207	2.000	6.000	(2.375, 3 768)
FILES STRUCTURE	MANUAL	13	3.200	3.000	1.424	1.000	6.000	(2.411, 3.989)
FILES STRUCTURE	IDES	14	3.500	4.000	1.225	2.000	6.000	(2.793, 4.207)
OFFICE AUTOMA.	MANUAL	13	2.000	2.000	1.000	1.000	4.000	(1.396, 2.604)
OFFICE AUTOMA.	IDES	14	2.214	2.000	1.188	1.000	5.000	(1 528, 2.901)

APPENDIX F
SOLUTION OF CLASS QUIZ

Solution to class Quiz

SELECT "CREATE (Rectange)"	
reach for the puck	H
point to "CREATE (Rectange"	M+P
press any button on mouse	K
CREATE RECTANGLE 1	
point to location of RECTANGLE 1	M +P
press any button on mouse	K
SELECT "CREATE (Circle)"	
point to "CREATE (Circle)"	M+P
press any button on mouse	K
CREATE CIRCLE 1	
point to location of CIRCLE 1	M+P
press any button on mouse	K
SELECT "CONNECT"	
point to "CONNECT"	M+P
press any button on mouse	K
CONNECT Figures	
point to RECTANGIE	M+P
press any button on mouse	K
point to CIRCLE	P
press any button on mouse	K

SOLUTION

$$T_{execute} = H + 6M + 7P + 7K$$

$$= 16.76 \text{ SECONDS}$$

Ease of Learning with TAMPEL

```
GOAL: ACQUIRE-UNIT TASK
IF (MORE RECTANGLES TO BE CREATED) THEN                RULE_RECL
   UNIT-TASK = CREATE (RECTANGLE)                      PROC_REGZ
ELSE IF (MORE CIRCLES TO BE CREATED) THEN              RULE_RECL
   UNIT TASK = CREATE(CIRCLE)                          PROC_REGZ
ELSE IF (FIGURES TO BE CONNECTED) THEN                 RULE_RECL
   UNIT TASK = CONNECT                                 PROC_RECL

GOAL: EXECUTE CREATE (RECTANGLE)
IF ("CREATE (RECTANGLE)" IS NOT SELECTED) THEN         RULE_RECL
   UNIT-TASK = SELECT ("CREATE (RECTANGLE")            PROC_RECL
ELSE IF ("CREATE (RECTANGEL IS SELECTED) THEN          RULE_RECL
   UNIT TASK = SELECT (LOCATION)                       PROC_RECL

GOAL: EXECUTE-CREATE (CIRCLE)
IF ("CREATE (CIRCLE)" IS NOT SELECTED) THEN            RULE_RECL
   UNIT-TASK = SELECT ("CREATE (CIRCLE))               TRANSFERRED
ELSEIF ("CREATE (CIRCLE)" IS SELECTED) THEN            RULE_RECL
   UNIT TASK = SELECT (LOCATION)                       TRANSFERRED

GOAL: EXECUTE-CONNECT
IF ("CONNECT" IS NOT SELECTED) THEN                    RULE_RECL
   UNIT-TASK = SELECT ("CONNECT")                      TRANSFERRED
ELSE IF ("CONNECT" IS SELECTED) THEN                   RULE_RECL
   UNIT TASK = SELECT (FIGURE 1)                       PROC_RECL
ELSEIF (FIGURE 1 IS SELECTED) THEN                     RULE_RECL
   UNIT TASK=SELECT FIGURE 2                           TRANSFERRED

GOAL: EXECUTE-SELECT (OBJECT)
IF (CURSOR IS NOT POINTING TO OBJECT) THEN             RULE_RECL
   UNIT-TASK = POINT-TO (OBJECT)                       PROC_RECL
ELSE IF (CURSOR IS POINTING TO OBJECT) THEN            RULE_RECL
   UNIT TASK = PRESS (ANY BUTTON)                      PROC_RECL
```

SOLUTION

RULE_RECL = 12

PROC_RECL = 6

PROC_REGZ = 2

ADD _RULES = 12 - (6+2) = 4

TIME TO LEARN = 278.65 SECONDS

APPENDIX G
SOLUTION TO TEST

EASE OF USE
KEYSTROKE-LEVEL MODEL
TASK I

SELECT "DELETE-NODE"	
reach for the puck	H
point to "DELETE NODE"	M+P
press any button on mouse	K
DELETE NODE 1	
point to location of node 1	M +P
press any button on mouse	K
DELETE NODE 2	
point to location of node 2	M+P
press any button on mouse	K
SELECT "DELETE-EDGE"	
point to DELETE-EDGE	M+P
press any button on mouse	K
DELETE EDGE	
point to edge	M+P
press any button on mouse	K

SOLUTION

$T_{execute} = H + 5M + 5P + 5K$

$= 13.05$ SECONDS
(12.65 SECONDS WITHOUT THE H)

EASE OF LEARNING
TAMPEL
TASK I

```
GOAL: ACQUIRE-UNIT TASK
IF (NODES TO BE DELETED) THEN                         RULE_RECL
   UNIT-TASK = DELETE-NODE                            PROC_REGZ
ELSE IF (EDGE TO BE DELETED) THEN                     RULE_RECL
   UNIT TASK = DELETE-EDGE                            PROC_REGZ

GOAL: EXECUTE-DELETE NODE
IF ("DELETE-NODE" IS NOT SELECTED) THEN               RULE_RECL
   UNIT-TASK = SELECT ("DELETE -NODE")                PROC_RECL
ELSE IF ("DELETE-NODE" IS SELECTED) THEN              RULE_RECL
   UNIT TASK = SELECT (LOCATION)                      PROC_RECL

GOAL: EXECUTE-DELETE EDGE
IF ("DELETE-EDGE" IS NOT SELECTED) THEN               RULE_RECL
   UNIT-TASK = SELECT ("DELETE-EDGE")                  TRANSFERRED
ELSE IF ("DELETE-EDGE" IS SELECTED) THEN               RULE_RECL
   UNIT TASK = SELECT (EDGE)                          PROC_RECL

GOAL: EXECUTE-SELECT (OBJECT)
IF (CURSOR IS NOT POINTING TO OBJECT) THEN            RULE_RECL
   UNIT-TASK = POINT-TO (OBJECT)                      PROC_RECL
ELSE IF (CURSOR IS POINTING TO OBJECT) THEN           RULE_RECL
   UNIT TASK = PRESS (ANY BUTTON)                     PROC_RECL
```

SOLUTION

RULE_RECL = 8

PROC_RECL = 5

PROC_REGZ = 2

ADD _RULES = 8 -(5 + 2) = 1

TIME TO LEARN = 217.05 SECONDS

EASE OF USE
KEYSTROKE-LEVEL MODEL
TASK II

SELECT "ADD-NODE"	
reach for the puck	H
point to "ADD-NODE"	M+P
press any button on mouse	K
CREATE NODE 1	
point to location of node 1	M +P
press any button on mouse	K
CREATE NODE 2	
point to location of node 2	M+P
press any button on mouse	
CREATE NODE 3	
point to location of node 3	M+P
press any button on mouse	K
SELECT "ADD-EDGE"	
point to ADD-EDGE	M+P
press any button on mouse	K
CREATE EDGE 1-2	
point to node 1	M+P
press any button on mouse	K
point to node 2	P
press any button on mouse	K
CREATE EDGE 2-3	
point to node 2	M+P
press any button on mouse	K
point to node 3	P
press any button on mouse	K
CREATE EDGE 3-1	
point to node 3	M+P
press any button on mouse	K
point to node 1	P
press any button on mouse	K

SOLUTION

$T_{execute} = H + 8M + 11P + 11K$

$= 24.18$ SECONDS
(23.78 SECONDS WITHOUT THE H)

EASE OF LEARNING
TAMPEL
TASK II

```
GOAL: ACQUIRE-UNIT TASK
IF (MORE NODES TO BE ADDED) THEN                        RULE_RECL
   UNIT-TASK = ADD-NODE (LOCATION)                      PROC_REGZ
ELSE IF (MORE EDGES TO BE ADDED) THEN                   RULE_RECL
   UNIT TASK = ADD-EDGE (NODE 1, NODE 2)                PROC_REGZ
ELSE IF (MORE NODES TO BE DELETED) THEN                 RULE_RECL
   UNIT TASK = DELETE NODE (NODE)                       PROC_REGZ
ELSE IF (MORE EDGES TO BE DELETED) THEN                 RULE_RECL
   UNIT TASK = DELETE EDGE (EDGE)                       PROC_REGZ

GOAL: EXECUTE ADD-NODE (LOCATION)
IF ("ADD NODE" IS NOT SELECTED) THEN                    RULE_RECL
   UNIT-TASK = SELECT ("ADD-NODE")                      PROC_RECL
ELSE IF ("ADD-NODE" IS SELECTED) THEN                   RULE_RECL
   UNIT TASK = SELECT (LOCATION)                        PROC_RECL

GOAL: EXECUTE-ADD-EDGE (NODE 1 NODE 2)
IF ("ADD-EDGE" IS NOT SELECTED) THEN                    RULE_RECL
   UNIT-TASK = SELECT ("ADD-EDGE")                       TRANSFERRED
IF ("ADD-EDGE" IS SELECTED) THEN                         RULE_RECL
   UNIT TASK = SELECT (NODE 1)                          PROC_RECL
IF (NODE 1 IS SELECTED) THEN                            RULE_RECL
   UNIT TASK = SELECT (NODE 2)                          TRANSFERRED

GOAL: EXECUTE-DELETE NODE (NODE)
IF ("DELETE-NODE" IS NOT SELECTED) THEN                 RULE_RECL
   UNIT-TASK = SELECT ("DELETE-NODE")                   TRANSFERRED
ELSE IF ("DELETE-NODE" IS SELECTED) THEN                RULE_RECL
   UNIT TASK = SELECT (NODE)                            TRANSFERRED

GOAL: EXECUTE-DELETE EDGE (EDGE)
IF ("DELETE-EDGE" IS NOT SELECTED) THEN                 RULE_RECL
   UNIT-TASK = SELECT ("DELETE-EDGE")                   TRANSFERRED
ELSE IF ("DELETE-EDGE" IS SELECTED) THEN                RULE_RECL
   UNIT TASK = SELECT (EDGE)                            PROC_RECL

GOAL: EXECUTE-SELECT (OBJECT)
IF (CURSOR IS NOT POINTING TO OBJECT) THEN              RULE_RECL
   UNIT-TASK = POINT-TO (OBJECT)                        PROC_RECL
ELSE IF (CURSOR IS POINTING TO OBJECT) THEN             RULE_RECL
   UNIT TASK = PRESS (ANY BUTTON)                       PROC_RECL
```

SOLUTION

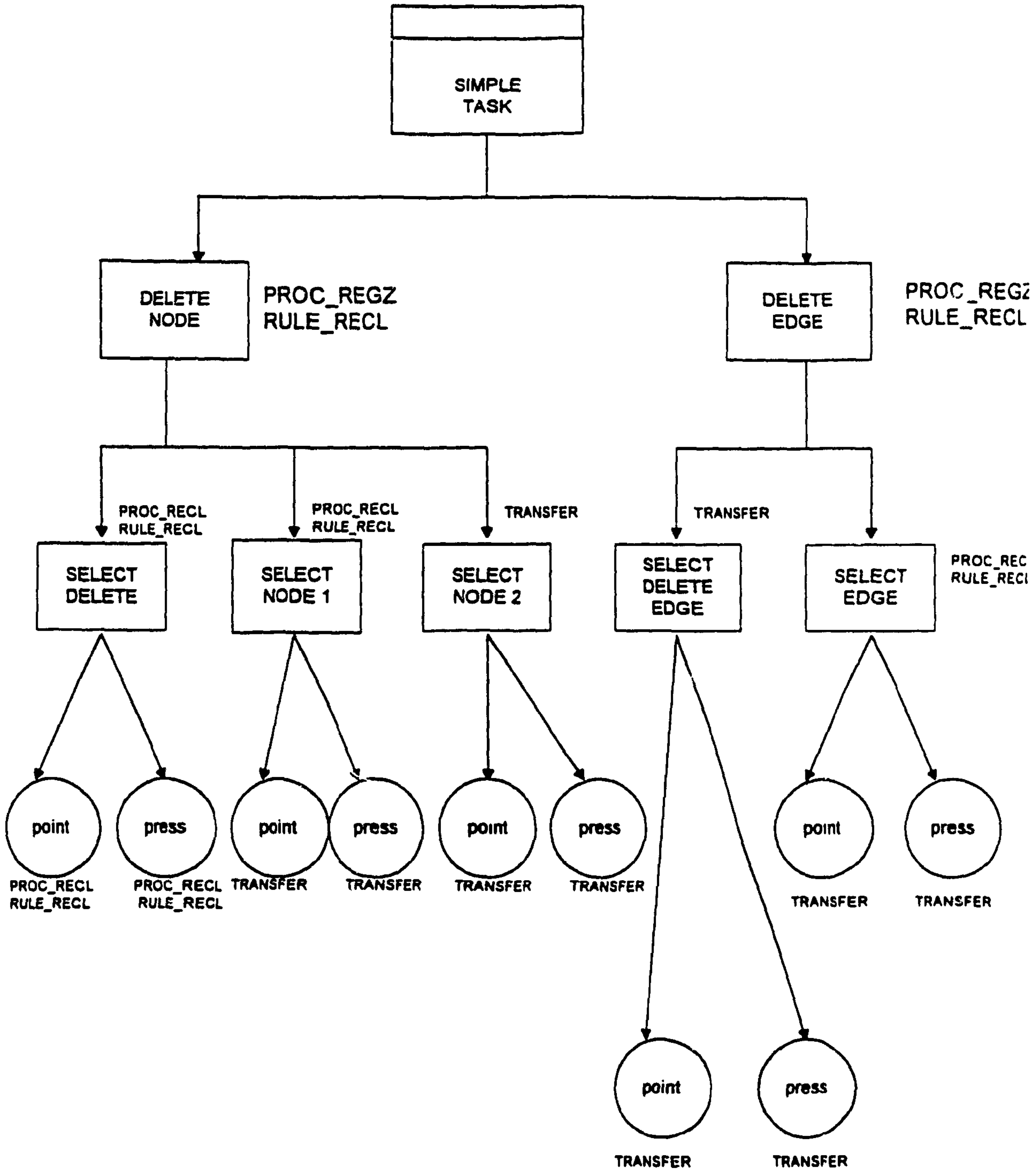
RULE_RECL = 15

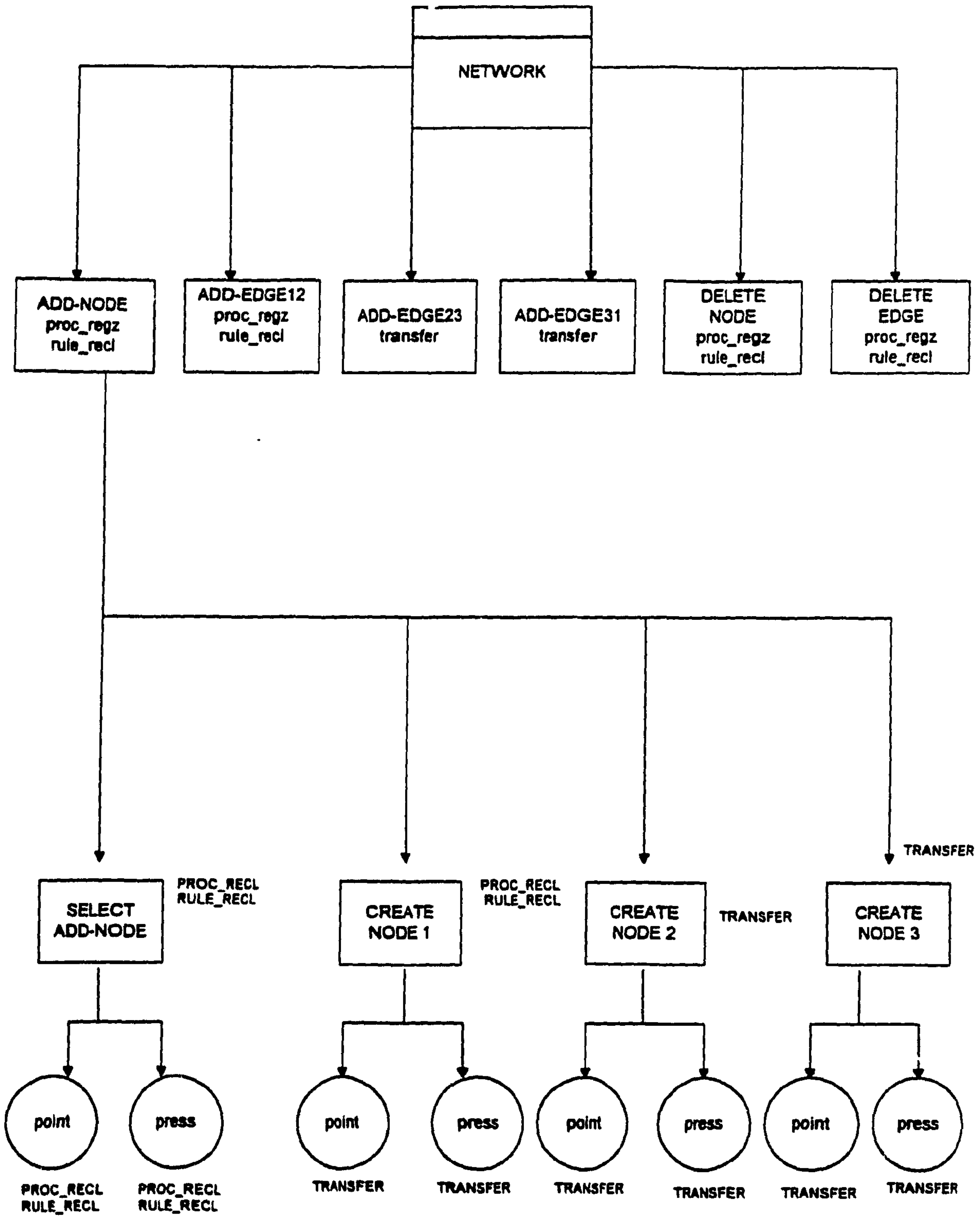
PROC_RECL = 6

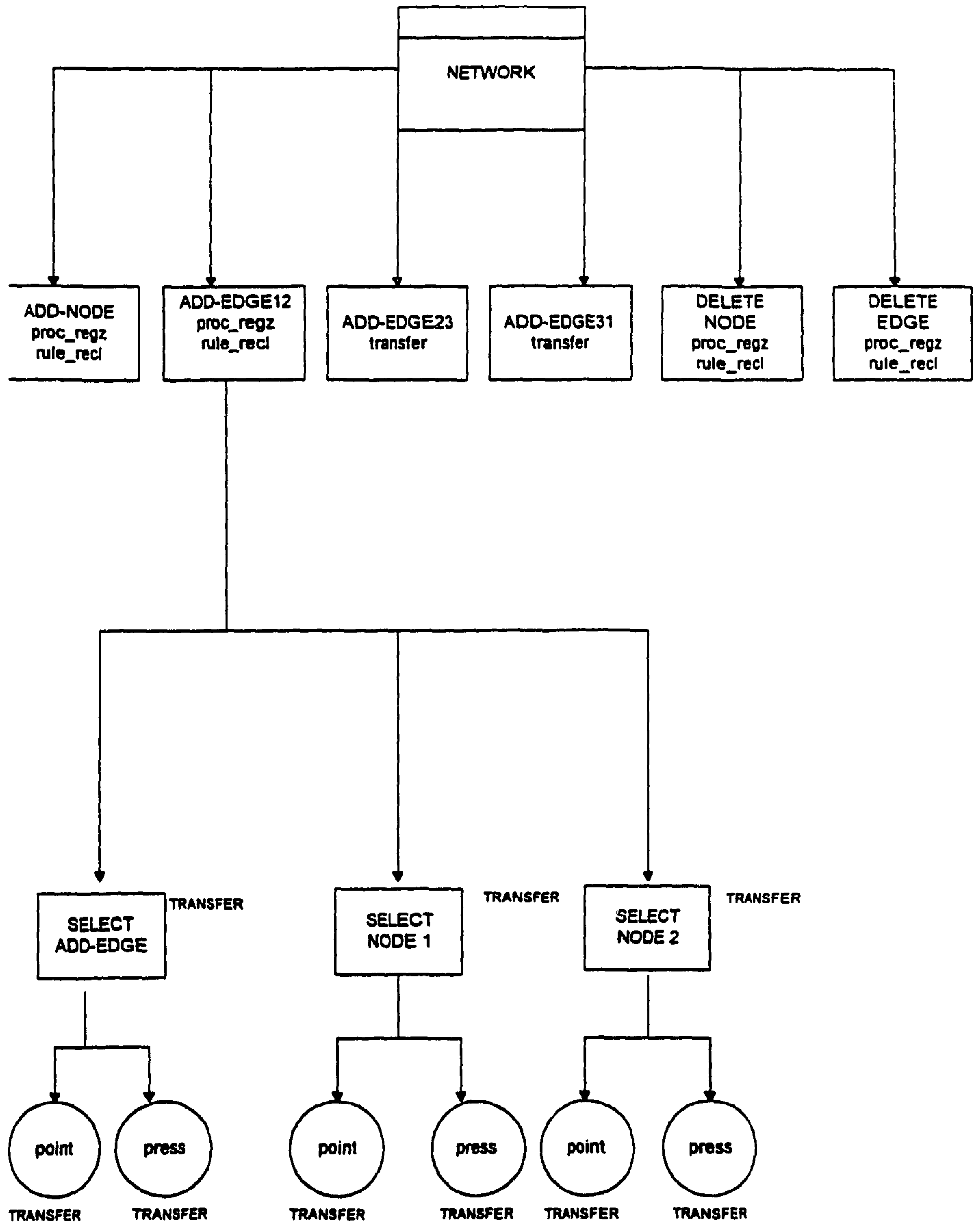
PROC_REGZ = 4

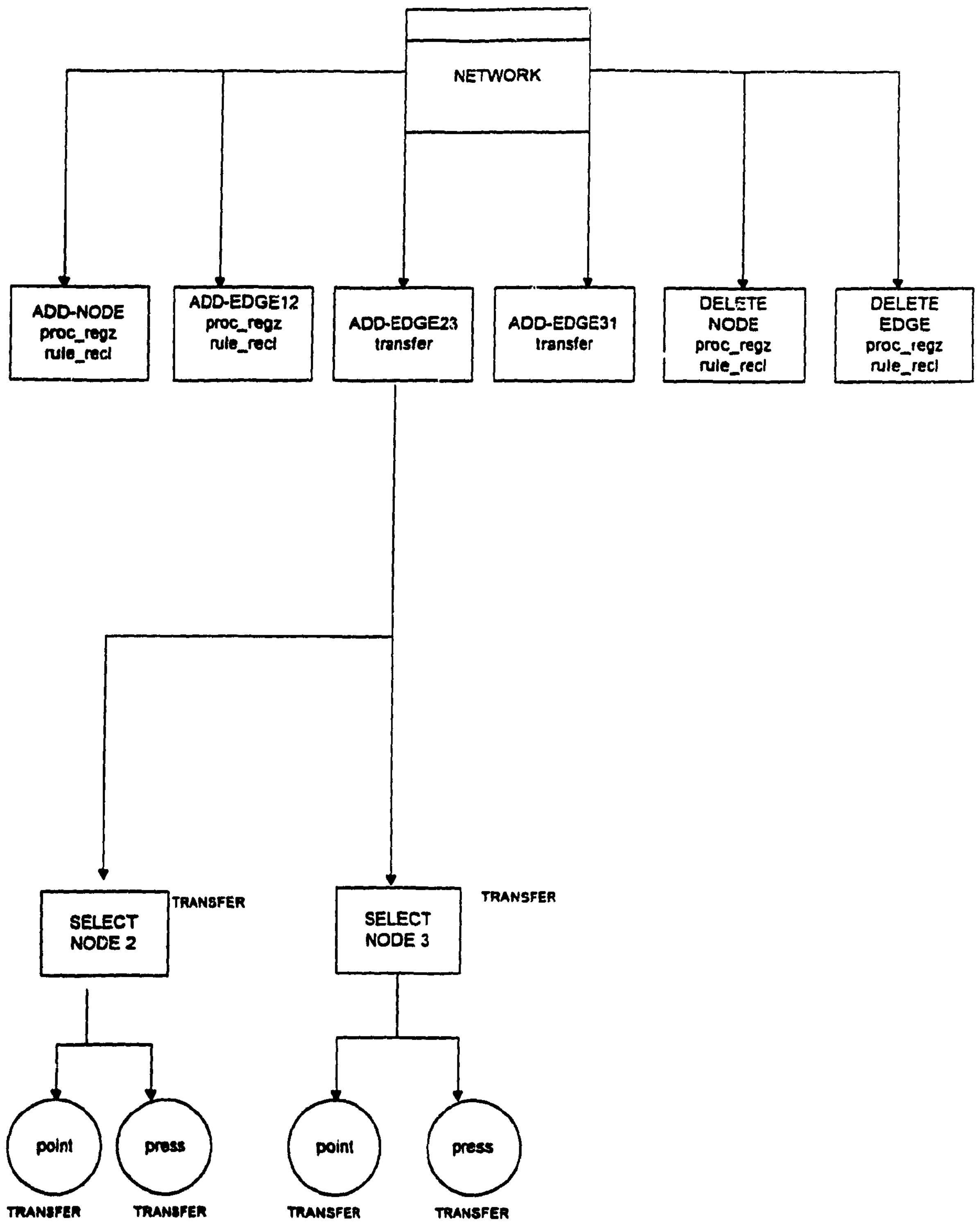
ADD _RULES = 15 - (6+4) = 5

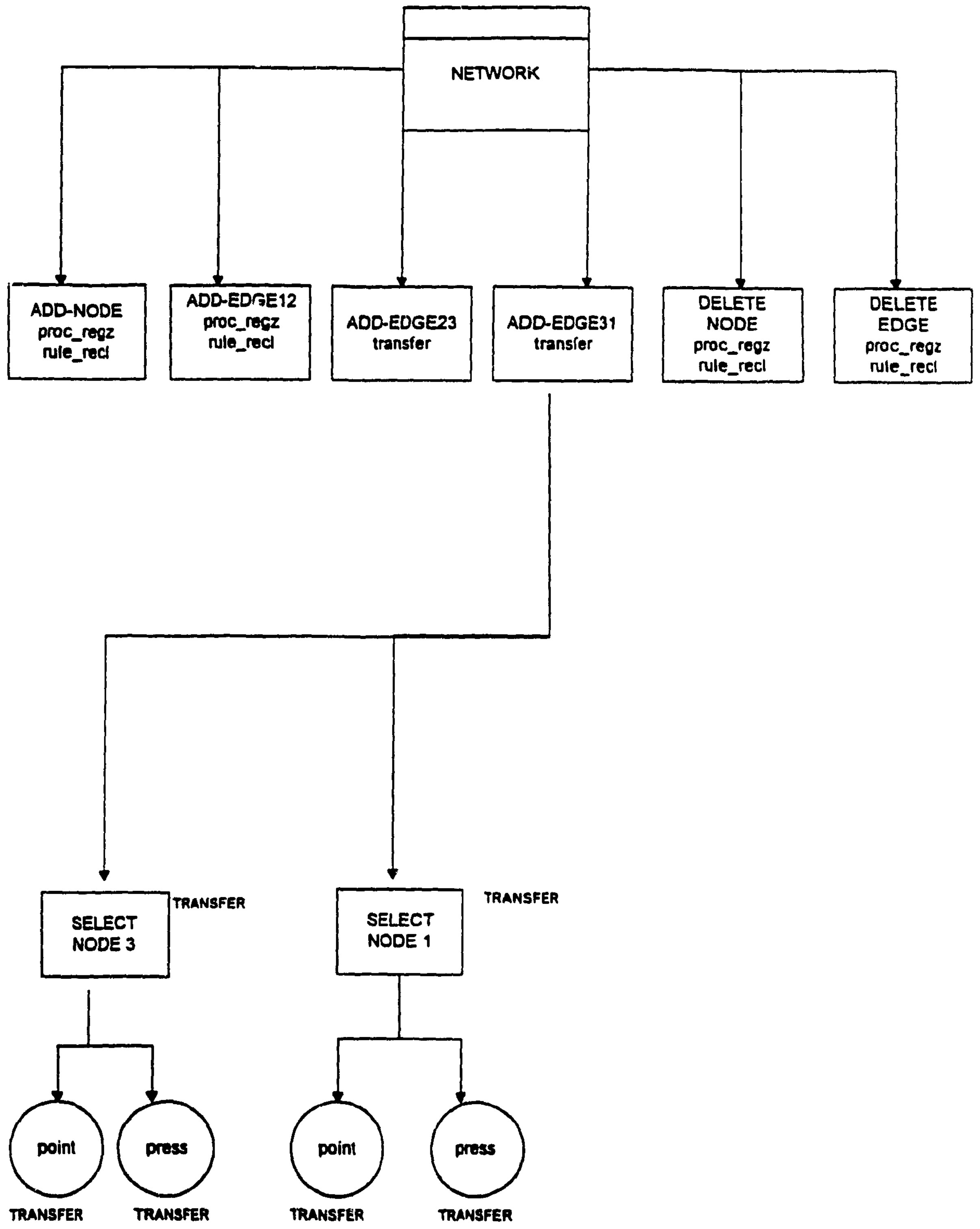
TIME TO LEARN = 335.70 SECONDS

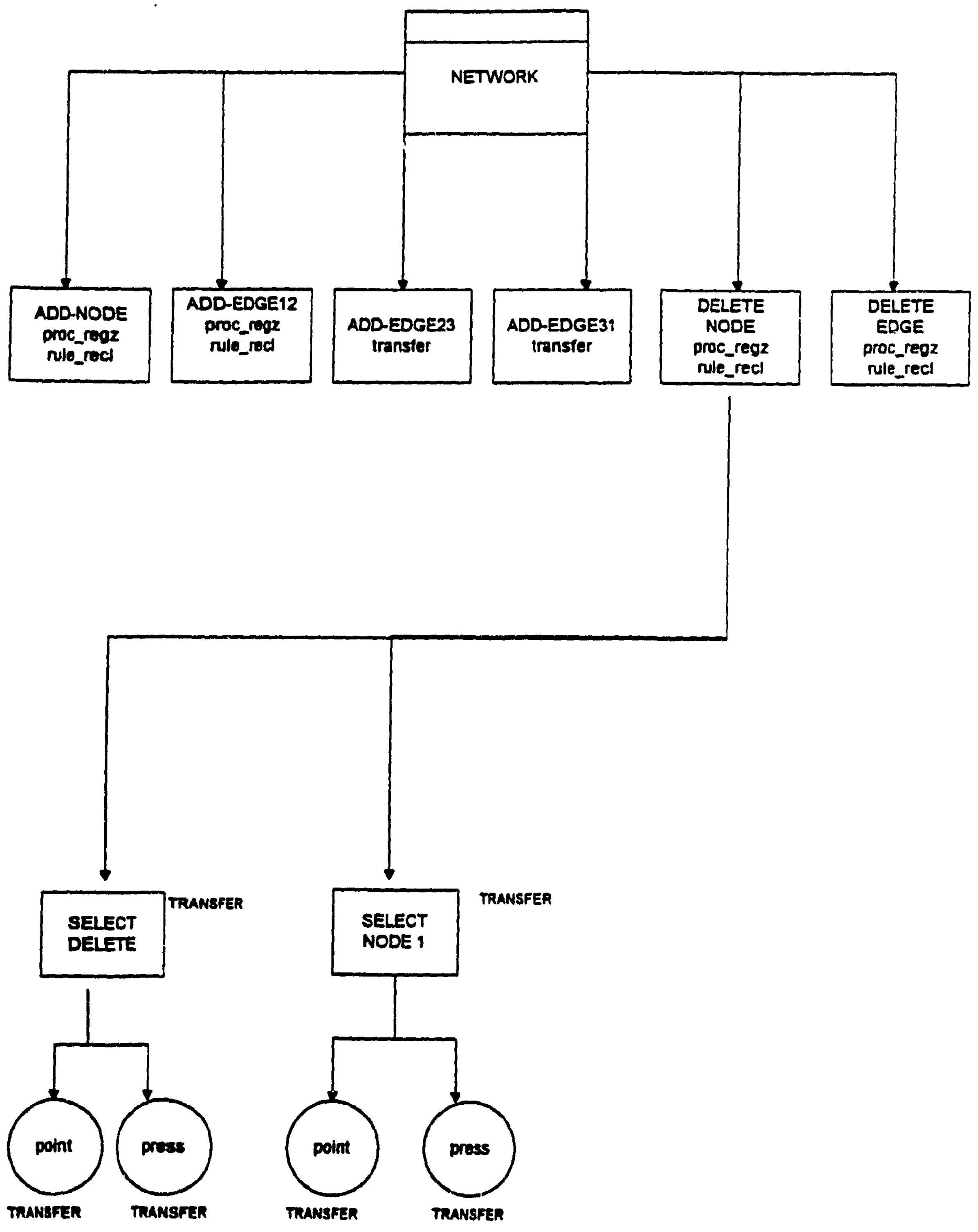

SIMPLE TASK
DELETE NODE
PROC_REGZ
RULE_RECL
DELETE EDGE
PROC_REGZ
RULE_RECL
PROC_RECL
RULE_RECL
SELECT DELETE
PROC_RECL
RULE_RECL
SELECT NODE 1
TRANSFER
SELECT NODE 2
TRANSFER
SELECT DELETE EDGE
SELECT EDGE
PROC_REC
RULE_RECI
point
press
PROC_RECL
RULE_RECL
PROC_RECL
RULE_RECL
TRANSFER
TRANSFER
TRANSFER
TRANSFER
TRANSFER
TRANSFER
TRANSFER
TRANSFER

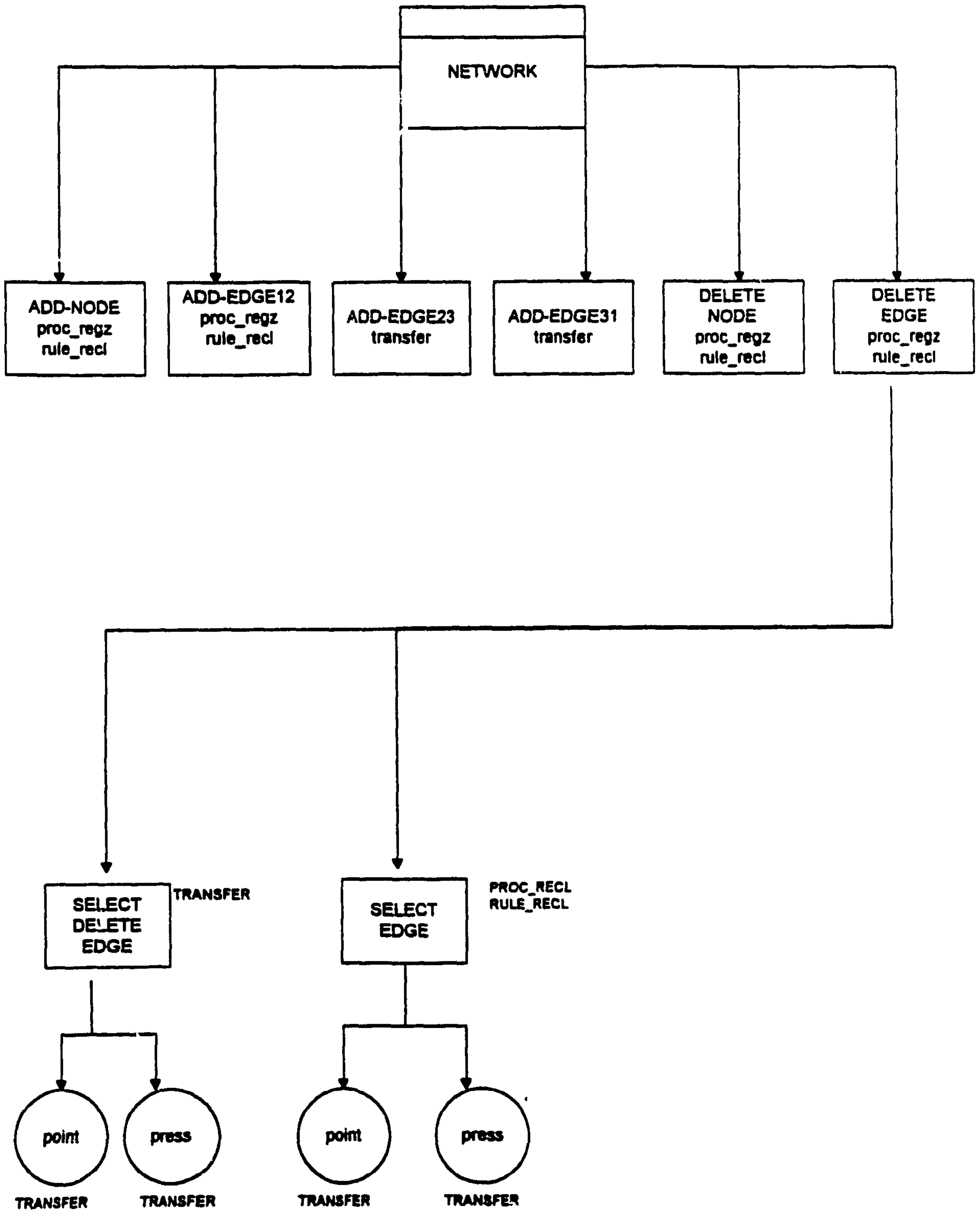

NETWORK
ADD-NODE proc_regz rule_recl
ADD-EDGE12 proc_regz rule_recl
ADD-EDGE23 transfer
ADD-EDGE31 transfer
DELETE NODE proc_regz rule_recl
DELETE EDGE proc_regz rule_recl
SELECT ADD-NODE
PROC_RECL RULE_RECL
CREATE NODE 1
PROC_RECL RULE_RECL
CREATE NODE 2
TRANSFER
CREATE NODE 3
TRANSFER
point
PROC_RECL RULE_RECL
press
PROC_RECL RULE_RECL
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER


NETWORK
ADD-NODE proc_regz rule_recl
ADD-EDGE12 proc_regz rule_recl
ADD-EDGE23 transfer
ADD-EDGE31 transfer
DELETE NODE proc_regz rule_recl
DELETE EDGE proc_regz rule_recl
SELECT ADD-EDGE
TRANSFER
SELECT NODE 1
TRANSFER
SELECT NODE 2
TRANSFER
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER


NETWORK
ADD-NODE proc_regz rule_recl
ADD-EDGE12 proc_regz rule_recl
ADD-EDGE23 transfer
ADD-EDGE31 transfer
DELETE NODE proc_regz rule_recl
DELETE EDGE proc_regz rule_recl
SELECT NODE 2
TRANSFER
SELECT NODE 3
TRANSFER
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER


NETWORK
ADD-NODE
proc_regz
rule_recl
ADD-EDGE12
proc_regz
rule_recl
ADD-EDGE23
transfer
ADD-EDGE31
transfer
DELETE
NODE
proc_regz
rule_recl
DELETE
EDGE
proc_regz
rule_recl
SELECT
NODE 3
TRANSFER
SELECT
NODE 1
TRANSFER
point
press
point
press
TRANSFER
TRANSFER
TRANSFER
TRANSFER


NETWORK
ADD-NODE proc_regz rule_recl
ADD-EDGE12 proc_regz rule_recl
ADD-EDGE23 transfer
ADD-EDGE31 transfer
DELETE NODE proc_regz rule_recl
DELETE EDGE proc_regz rule_recl
SELECT DELETE
TRANSFER
SELECT NODE 1
TRANSFER
point
TRANSFER
press
TRANSFER
point
TRANSFER
press
TRANSFER


NETWORK
ADD-NODE proc_regz rule_recl
ADD-EDGE12 proc_regz rule_recl
ADD-EDGE23 transfer
ADD-EDGE31 transfer
DELETE NODE proc_regz rule_recl
DELETE EDGE proc_regz rule_recl
SELECT DELETE EDGE
TRANSFER
SELECT EDGE
PROC_RECL
RULE_RECL
point
press
point
press
TRANSFER
TRANSFER
TRANSFER
TRANSFER

APPENDIX H
ATTITUDE QUESTIONS ON MODELS

EVALUATION OF THE KEYSTROKE-LEVEL MODEL AND TAMPEL

Please answer the following questions regarding the use of the Keystroke-level Model and TAMPEL to evaluate the ease of use and the ease of learning of an interface Please circle your answer in the appropriate box. In answering these questions think of the different methods to evaluate interfaces that were discussed in class.

1) By using the Keystroke-level Model and TAMPEL, how much faster or slower is it to evaluate the ease of use and the ease of learning of an interfaces?

-2	-1	0	1	2
Much Slower	Slower	Neutral	Faster	Much Faster

2) By using the Keystroke-level Model and TAMPEL, how much better of worse is your performance in evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much Worse	Worse	Neutral	Better	Much Better

3) By using the Keystroke-level Model and TAMPEL, how much more or less productive are you when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much less Productive	Less Productive	Neutral	More Productive	Much More Productive

4) By using the Keystroke-level Model and TAMPEL, how much more or less effective are you when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much less Effective	Less Effective	Neutral	More Effective	Much More Effective

5) By using the Keystroke-level Model and TAMPEL, how much easier of harder is it to evaluate the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much Harder	Harder	Neutral	Easier	Much Easier

6) How useful is the Keystroke-level Model and TAMPEL when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Useless	Somewhat Useless	Neutral	Useful	Very Useful

7) How easy or difficult is it to learn how to apply the Keystroke-level Model and TAMPEL?

-2	-1	0	1	2
Very Difficult	Difficult	Neutral	Easy	Very Easy

8) How easy or difficult is it to become skillful at using the Keystroke-level Model and TAMPEL?

-2	-1	0	1	2
Very Difficult	Difficult	Neutral	Easy	Very Easy

9) How easy or difficult is it to apply the Keystroke-level Model and TAMPEL?

|_____-2_____|_____-1_____|_____0_____|_____1_____|_____2_____|

-2	-1	0	1	2
Very Difficult	Difficult	Neutral	Easy	Very Easy

10) Overall, how satisfied or unsatisfied are you with the Keystroke-level Model and TAMPEL as a tool to evaluate the ease of use and the ease of learning of an interface?

|_____-2_____|_____-1_____|_____0_____|_____1_____|_____2_____|

-2	-1	0	1	2
Very Unsatisfied	Unsatisfied	Neutral	Satisfied	Very Satisfied

APPENDIX I
ATTITUDE QUESTIONS ON IDES

EVALUATION OF IDES

In the class lecture you were taught how to evaluate the ease of use and the ease of learning of an interface by applying the Formal Models manually The following questions will ask you to compare the methods taught in class with that of using IDES to evaluate the ease of use and the ease of learning of an initerface. Please circle the answer in the appropriate box.

1) By using IDES, how much faster or slower is it to evaluate the ease of use and the ease of learning of an interfaces?

-2	-1	0	1	2
Much Slower	Slower	Neutral	Faster	Much Faster

2) By using IDES, how much better of worse is your performance in evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much Worse	Worse	Neutral	Better	Much Better

3) By using IDES, how much more or less productive are you when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much less Productive	Less Productive	Neutral	More Productive	Much More Productive

4) By using IDES, how much more or less effective are you when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much less Effective	Less Effective	Neutral	More Effective	Much More Effective

5) By using IDES, how much easier of harder is it to evaluate the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Much Harder	Harder	Neutral	Easier	Much Easier

6) How useful is IDES when evaluating the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Useless	Somewhat Useless	Neutral	Useful	Very Useful

The following questions relate to the use of IDES as a tool to evaluate the ease of use and the ease of learning of an interface. Please circle you answer in the appropriate box.

7) How easy or difficult is it to learn to operate IDES?

-2	-1	0	1	2
Very Difficult	Difficult	Neutral	Easy	Very Easy

8) How easy or difficult is it to get IDES to do what you want it to do?

|______-2______|______-1______|______0______|______1______|______2______|

Very Difficult	Difficult	Neutral	Easy	Very Easy

9) My interaction with IDES is clear and understandable?

|______-2______|______-1______|______0______|______1______|______2______|

Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

10) How flexible is IDES to interact with?

|______-2______|______-1______|______0______|______1______|______2______|

Very Inflexible	Inflexible	Neutral	Flexible	Very Flexible

11) How easy or difficult is it to become skilfull at using IDES?

|______-2______|______-1______|______0______|______1______|______2______|

Very Difficult	Difficult	Neutral	Easy	Very Easy

12) How easy or difficult is it to use IDES?

|______-2______|______-1______|______0______|______1______|______2______|

Very Difficult	Difficult	Neutral	Easy	Very Easy

13) How useful is the presentation format of the output of IDES?

-2	-1	0	1	2
Useless	Somewhat Useless	Neutral	Useful	Very Useful

14) How clear is the information provided by IDES?

-2	-1	0	1	2
Very Unclear	Unclear	Neutral	Clear	Very Clear

15) Overall, how satisfied or unsatisfied are you with IDES as a tool to evaluate the ease of use and the ease of learning of an interface?

-2	-1	0	1	2
Very Unsatisfied	Unsatisfied	Neutral	Satisfied	Very Satisfied

APPENDIX K
DESCRIPTIVE STATISTICS

Descriptive Statistics

1. SIMPLE TASK (TASK I)

A. KEYSTROKE-LEVEL MODEL

1) K1MS = Evaluation of ease of use using the manual method

K1MS = (12.65, 12.65, 12.65, 7.99, 12.65, 5.06, 12.65, 12.65, 7.59, 12.65, 13.05, 14.17)

N = 13

MEAN = 11.077

MEDIAN = 12.65

STDEV = 2.899

SEMEAN = 0.804

MIN = 5.06

MAX = 14.170

2) K1MSD = Deviation from the true ease of use score using the manual method

K1MSD = (0.00, 0.00, 0.00, -4.66, -5.06, 0.00, -7.59, 0 00, 0.00, -5.06, 0.00, 0 40, 1.52)

N = 13

MEAN = -1.73

MEDIAN = 0.000

STDEV = 2.899

SEMEAN = 0.804

MIN = -7.590

MAX = 1.520

3) KIMT = Time to complete the task using the manual method

KIMT = (5, 2, 4, 6, 4, 5, 4, 18, 6, 7, 4, 9, 7)

N = 13

MEAN = 6.23

MEDIAN = 5.00

STDEV = 3.96

SEMEAN = 1.10

MIN = 2.00

MAX = 18.00

4) K1IS = Ease of use evaluation using IDES

K1IS = (12.65, 15.18, 10.12, 12.65, 12.65, 6.59, 14.18, 6.59, 12.65, 12.65, 13 05, 15 18, 12.65, 7.59)

N = 14

MEAN = 12.956

MEDIAN = 13.65

STDEV = 2.715

SEMEAN = 0.726

MIN = 8.590

MAX = 16.180

5) K1ISD = Deviation from the true ease of use using IDES

K1ISD = (0.00, 2.53, -2.53, 0.00, 0.00. -5.06. 2.53. -5.06. 0.00. 0.00. 0 40. 2.53. 0 00. -5.06)

N = 14

MEAN = -0.694

MEDIAN = 0.00

STDEV = 2.715

SEMEAN = 0.726

MIN = -5.060

MAX = 2.530

6) K1IT = Time to measure ease of use using IDES

K1IT = (5, 10, 25, 20, 12, 11, 20, 16, 14, 26, 20, 25, 18, 20)

N = 14

MEAN = 17.29

MEDIAN = 19.00

STDEV = 6.26

SEMEAN = 1.67

MIN = 5.00

MAX = 11.75

B. TAMPEL

1) T1MS = Evaluation of ease of learning using the manual method

T1MS = (265, 253, 340, 276, 290, 265, 382, 153, 290, 324, 265, 314, 267)

N = 13

MEAN = 283.4

MEDIAN = 276

STDEV = 53.9

SEMEAN = 14.9

MIN = 153

MAX = 382

2) T1MSD = Deviation from true score for ease of learning using the manual method

T1MSD = (47.95, 35.95, 122.95, 58.95, 72.95, 47.95, 164.95, -64.05, 72.95, 106.95, 47.95, 96.95, 49.95)

N = 13

MEAN = 66.3

MEDIAN = 58.90

STDEV = 53.9

SEMEAN = 14.9

MIN = -64.01

MAX = 164.9

4) TIIS = Evaluation of ease of learning using IDES

TIIS = (254, 341, 179, 155, 217, 211, 280, 185, 211, 217, 153, 305, 298, 272)

N = 14

MEAN = 234.1

MEDIAN = 217.0

STDEV = 58.4

SEMEAN = 15.6

MIN = 153

MAX = 341

3) TIMT = Time to measure ease of learning using the manual method

TIMT = (20, 29, 20, 26, 15, 25, 23, 23, 26, 22, 14, 20, 26)

N= 13

MEAN = 22.23

MEDIAN = 23.00

STDEV = 4.40

SEMEAN = 1.22

MIN = 14.00

MAX = 29.00

5) T1ISD = deviation from true ease of learning using IDES

T1ISD = (36.95, 123.95, -38 05, -62.05, -0 05, -6.05, 62.95, -32.05, -6 05, -0.05, -64 05, 87.95, 80.95, 54 95)

N = 14

MEAN = 17.1

STDEV = 58.4

SEMEAN = 15.6

MIN = -64.1

MAX = 123.9

6) T1IT = Time to measure ease of learning using IDES

T1IT = (6, 2, 8, 2, 3, 2, 6, 1, 11, 4, 6, 2, 7, 8)

N = 14

MEAN = 4.857

MEDIAN = 5.000

STDEV = 3.009

SEMEAN = 0.804

MIN = 1.00

MAX = 11.00

2. COMPLEX TASK (TASK II)

A. KEYSTROKE-LEVEL MODEL (EASE OF USE)

1) K2MS = Evaluation of ease of use using the manual method

K2MS = (25.53, 17.94, 25.53, 13.68, 7.65, 20.63, 7.82, 15 41 17.94, 29.17, 23.33, 19.93, 17.94)

N = 13

MEAN = 18.65

MEDIAN = 17.94

STDEV = 6.51

SEMEAN = 1.80

MIN = 7.65

MAX = 29.17

5) K2ISD = Deviation from true score for ease of use using IDES

K2ISD = (4.05, -1.01, -6.07, 4.05, -3.54, -8.60, 1.52, 4.05, -6 07, 6 58, -11 13, 4.05, 9.11, 4.05)

N = 14

MEAN = -0.07

MEDIAN = 2.78

STDEV = 6 17

SEMEAN = 1.65

MIN = -11.13

MAX = 9.71

C. TOTAL TIME TO MEASURE EASE OF USE AND EASE OF LEARNING

1) TT1M = Total time to measure botht he ease of use and the ease of learning using the manual method

TT1M = (25, 31, 24, 32, 19, 30, 27, 41, 32, 29, 18, 29, 33)

N = 13

MEAN = 28.46

MEDIAN = 29.00

STDEV = 6.09

SEMEAN = 1.66

MIN = 18.00

MAX = 41.00

2) K2MSD = Deviation from true score for ease of use using the manual method

K2MSD = (1.75, -5.84, 1.75, -10.10, -16.13, -3.15, -15.96, -8.37, -5 84, 5 39, -0.45, -3.85, -5.84)

N = 13

MEAN = -5.13

MEDIAN = -5.84

STDEV = 6.51

SEMEAN = 1.80

MIN = -16.13

MAX = 5.39

3) K2MT = Time to measure ease of use using the manual method

K2MT = (9, 6, 7, 7, 5, 7, 10, 14, 4, 11, 7, 12, 10)

N = 13

MEAN = 8.385

MEDIAN = 7.000

STDEV = 2.902

SEMEAN = 0.805

MIN = 4.000

MAX = 14.00

4) K2IS = Evaluation of ease of use using IDES

K2IS = (27.83, 22.77, 17.71, 27 83, 20.24, 15 18, 25.30, 27 83, 17 71, 30 36, 12.65, 27.83, 32.89, 27.83)

N = 14

MEAN = 23.85

MEDIAN = 26.56

STDEV = 6.17

SEMEAN = 1.65

MIN = 12.65

MAX = 32.89

6) K2IT = Time to measure ease of use using IDES

K2IT = (7. 8. 18. 14. 7. 13. 18. 11. 9. 23. 17. 24. 13. 19)

N = 14

MEAN = 14.36

MEDIAN = 13.50

STDEV = 5.65

SEMEAN = 1.51

MIN = 7.00

MAX = 24.00

B. TAMPEL (EASE OF LEARNING)

1) T2MS = evaluation of ease of learning using the manual method

T2MS = (500, 473, 372, 402, 570, 430, 488, 338, 521, 1151, 438, 627, 612)

N = 13

MEAN = 532.5

MEDIAN = 489

STDEV = 205.3

SEMEAN = 56.9

MIN = 338

MAX = 1151

2) T2MSD = Deviation from true ease of learning using the manual method

T2MSD = (164.3, 137.3, 36.3, 66.3, 234.3, 94.3, 152.3, 2.3, 185.3, 815.3, 102.3, 291.3, 276.3)

N = 13

MEAN = 196.8

MEDIAN = 152.3

STDEV = 205.3

SEMEAN = 57.0

MIN = 2.3

MAX = 815.3

3) T2MT = Time to measure ease of learning using the manual method

T2MT = (39, 22, 29, 26, 20, 30, 28, 45, 27, 28, 19, 23, 36)

N = 13

MEAN = 28.62

MEDIAN = 28.00

STDEV = 7.56

SEMEAN = 2.10

MIN = 19

MAX = 45

4) T2IS = Time to measure ease of learning using IDES

T2IS = (282, 434, 358, 329, 470, 471, 327, 376, 507, 432, 297, 434, 507, 394)

N = 14

MEAN = 401.3

MEDIAN = 413.0

STDEV = 75.2

SEMEAN = 20.1

MIN = 282

MAX = 507

5) T2ISD = Deviation from true ease of learning using IDES

T2ISD = (-53.7, 98.3, 22.3, -6.7, 134.3, 135 3, -8 7, 40.3, 171 3, 96 3, -38 7, 98.3, 171.3, 58.3)

N = 14

MEAN = 65.3

MEDIAN = 98.0

STDEV = 75.2

SEMEAN = 20.0

MIN = -54

MAX = 171

2) TT1I = Total tume to measure ease of use and ease of learnung usıng IDES

TT1I = (11, 12, 33, 22, 15, 13, 26, 17, 25, 30, 26, 27, 25, 28)

N = 14

MEAN = 22.14

MEDIAN = 25.00

STDEV = 7.19

SEMEAN = 1.92

MIN = 11.00

MAX = 33.00

6) T2IT = Time to measure ease of learning using IDES

T2IT = (5, 7, 9, 5, 11, 4, 7, 4, 8, 14, 10, 10, 8, 10)

N = 14

MEAN = 8.0

MEDIAN = 8.00

STDEV = 2.909

SEMEAN = 0.777

MIN = 4.0

MAX = 14.00

C. TOTAL TIME FOR EASE OF USE AND EASE OF LEARNING

1) TT2M = Total time for ease of use and ease of learning using the manual method

TT2M = (48, 28, 36, 33, 25, 37, 38, 59, 31, 39, 26, 35)

N = 13

MEAN = 37

MEDIAN = 36.00

STDEV = 9.55

SEMEAN = 2.65

MIN = 25.00

MAX = 59.00

2) TT2I = Total time to measure ease of use and ease of learning using IDES

TT2I = (12, 15, 27, 19, 18, 17, 25, 15, 17, 37, 27, 34, 21, 29)

N = 14

MEAN = 22.36

MEDIAN = 20.00

STDEV = 7.59

SEMEAN = 2.03

MIN = 12.00

MAX = 37.00

BACKGROUND INFORMATION

YEARS OF UNIVERSITY EDUCATION

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
YEARS	MANUAL	13	3 154	3 000	0 376	3 000	4 000	(2 927 3 381)
YEARS	IDES	14	3.357	3 000	1 008	2 000	5 000	(2 77[illegible] 3 939)

UNIVERSITY M.I.S. COURSES

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
# OF COURSES	MANUAL	13	8 000	9.000	1 780	5 000	10 000	(6 94 9 076)
# OF COURSES	IDES	14	6 929	7 000	2 303	2 000	11 000	(5 599 8 258)

FORMAL M.I.S. WORK EXPERIENCE

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
YEARS	MANUAL	13	0.0218	0 000	0.381	0 000	1 000	(-0 013, 0 448)
YEARS	IDES	14	0 0950	0	0.2749	0 000	1 000	(-0 0638, 0 2538)

KNOWLEDGEABLE WITH THE FOLLOWING

QUESTION	GROUP	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C.I.
OPER. SYSTEM	MANUAL	13	3 385	4 000	1.502	1.000	6 000	(2 477, 4 29[illegible]
OPER. .SYSTEM	IDES	14	3.357	3 000	1.646	1 000	6 000	(2.407, 4 308)
COMP. HARDWARE	MANUAL	13	3.308	3 000	1.377	1.000	6 000	(2 475, 4 140)
CCMP. HARDWARE	IDES	14	3.429	3 000	1 505	1 000	6.000	(2.560, 4 298)
INFO. REQUIRE.	MANUAL	13	2.923	3.000	1.256	1.000	5 000	(2 164, 3 682)
INFO. REQUIRE.	IDES	14	3.214	3.000	1.051	2.000	6 000	(2 607, 3 821)
OUTPUT/INPUT	MANUAL	13	2.692	3 000	1 032	1 000	4 000	(2.069, 3 316)
OUTPUT/INPUT	IDES	14	3 071	3 000	1.207	2.000	6 000	(2 375, 3 768)
FILES STRUCTURE	MANUAL	13	3 200	3.000	1.424	1 000	6.000	(2.411, 3 989)
FILES STRUCTURE	IDES	14	3.500	4 000	1.225	2 000	6 000	(2.793, 4 207)
OFFICE AUTOMA.	MANUAL	13	2.000	2.000	1.000	1 000	4 000	(1 396, 2 604)
OFFICE AUTOMA.	IDES	14	2.214	2.000	1 188	1 000	5 000	(1.528, 2.901)

SUMMARY STATISTICS FOR CLASS QUIZ

EASE OF USE (KEYSTROKE-LEVEL MODEL)

ESTIMATE	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
SCORE (deviation)	28	-5 53	1.15	5 67	-11 30	16 37	(-3 34, 1 05)
TIME	28	9 429	9 5	3 120	5 000	16 000	(8 218, 10 639)

EASE OF LEARNING (TAMPEL)

ESTIMATE	N	MEAN	MEDIAN	STDEV	MIN	MAX	95 % C. I.
SCORE (deviation)	28	99.7	100 0	72 7	0 1	255	(71.5, 127 9)
TIME	28	27 93	27 50	6 64	18.00	50 00	(25 35, 30 51)