ACKNOWLEDGMENTS

There are many people to whom I owe gratitude and appreciation for the support and inspiration they gave me during the writing of this study.

I am deeply grateful to my supervisor, Dr. Marlene Scardamalia. Without her guidance, constructive criticism and inspiring suggestions, this thesis could not have been completed in its present form. I am also grateful for her patience and support she had shown me, especially in the last stages of this project.

I am also deeply grateful to Dr. Peggy Ng. Without her inspiring comments, constructive criticism and support in my most difficult moments, this thesis would never have been possible. I am most appreciative of Dr. Ng's commitment to statistical accuracy and integrity. In the heat of fire, and in my lowest moments, Dr. Ng always persuaded me to keep my eye on the goal and to focus my energies in a positive way - for this I will also owe a debt.

I am also very grateful to Dr. Harriot Botta. Dr. Botta gave me the strength to complete this project to the end. Her kind words were always a motivating factor to my success with this project. Seeing Dr. Botta at my defense gave me the reassurance needed to present my findings with confidence. For this I will always be eternally grateful. I am also very grateful to Dr. Allan Trojan. He believed in the "code" behind this study. His appreciation for this effort gave me the conviction to see this project through to the end.

I am also very grateful to Dr. Ruth Hayhoe. Her administration and handling of my thesis was more than I could ever have hoped for. Dr. Hayhoe made me feel like there was always a light at the end of the tunnel. She restored my faith in the system. In the times when I needed reassurance, Dr. Hayhoe was there for me. My words cannot express the appreciation I have for Dr. Hayhoe, but I have thanked the Lord many times for placing her in charge of administering my project.

I am also grateful to Dr. Wahlstrom (internal appraiser) and Dr. Kirkwood (external appraiser) for their constructive criticisms and suggestions.

I would like to thank my family for everything that they have personally committed to this study - for believing in me, and insisting that I go forward. Even when things were proceeding slowly, they had faith and insisted that I look to the Lord for strength - for this I will always be grateful.

Two people have always been there for me and without their help, support and advice I would not have been able to complete this study. Let me first thank and acknowledge the years of support my friend Mr. Stephen Mills has given me. I cannot begin to thank you Stephen for your input and understanding. You stood beside me on the firing line and for this I will owe you a debt forever. To Ms. Liz Hoffman I owe an eternal debt. Her knowledge of how to make things happen, how to deal with the larger issues, gave this project a new life. How can I ever repay such a large debt? My eternal thanks Liz.

Over the years, many people have assisted me with this project - I would like to thank Professor Bob Kurys at Ryerson Polytechnic University, for his kind words and for a comment that I will always remember, i.e., "Go for the golden ring!" Thanks Bob. Over the years many of my teaching assistants have come to my rescue when I have been pulled in every direction possible. You know how much your help has meant to me. I only hope that one day I can return the favour.
ABSTRACT

The maxim of developers with respect to the design and use of expert systems (ES) is that "a system is only as good as it is useful." Practically speaking, the value of an expert system resides in its continued usage and this continued exposure to the knowledge base that is embedded in its underlying structure. User access is, in turn, directly related to interface design and related to user preferences. The present study examines an expert system developed by the author, called CAAD (Computer-Assisted Automotive Assistant) with respect to user interface preference.

CAAD was designed to address and diagnose automotive failures for General Motors, Ford, and Chrysler products. Performance and preference data were collected from 250 participants, each having operated CAAD under a Character-based User Interface (CUI) and a Graphically-based User Interface (GUI).

Central to this study are the following questions:

1. Do inexperienced computer users prefer using an ES equipped with a graphical user interface over the same system equipped with a character user interface?
2. As a user's level of computer experience increases, will there be a shift in preference for an ES equipped with a character user interface?
3. Is it possible for people to be equally effective solving problems related to the contents of the ES given one interface or the other?

The thesis provides a naturalistic context for this inquiry; namely, an automotive repair shop. Participants visited this shop to have their car repaired, and were asked to volunteer to use the expert system while waiting. Participants ranged from age 16-66, with approximately equal numbers of males and females.

In addition to examining interface preferences the thesis also explores gender and age differences in the data set. In the concluding section of the thesis, general issues regarding the design of expert systems are discussed.

KEY WORDS

Table of Contents

Abstract...................................................................................................................................................... ii
Preface........................................................................................................................................................... iii
1 Introduction............................................................................................................................................... 1
  1.1 Statement of the Problem...................................................................................................................... 3
  1.2 Goal of this Research.............................................................................................................................. 3
  1.3 Hypotheses........................................................................................................................................... 5
  1.4 Problem Environment............................................................................................................................. 6
  1.5 Types of User Interfaces.......................................................................................................................... 6
  1.6 Character User Interfaces......................................................................................................................... 6
    1.6.1 Menu-driven CUI.............................................................................................................................. 6
  1.7 Graphical User Interface (GUI)................................................................................................................ 7
  1.8 Current Research on GUI vs. CUI Usage................................................................................................. 13
    1.8.1 The Influence of User Skill on Interface Preference........................................................................ 14
    1.8.2 GUI Preference and Computer Experience...................................................................................... 18
    1.8.3 CUI Preference and Computer Experience.................................................................................... 21
    1.8.4 The Influence of GUI Experience on Productivity and Error Rates............................................. 24
    1.8.5 Gender Influence and Interface Preference.................................................................................... 27
    1.8.6 Younger people and GUI Preference.............................................................................................. 28
2 Study Description....................................................................................................................................... 30
  2.1 Purpose.................................................................................................................................................. 30
  2.2 Method.................................................................................................................................................. 30
    2.2.1 Measurement Criteria - Legend of Variables...................................................................................... 33
    2.2.2 Measurement Description.................................................................................................................. 34
      2.2.2.1 How was Computer Experience Determined?........................................................................... 34
    2.2.3 Measurement Description Continued................................................................................................ 37
      2.2.3.1 How was an Error Determined (CUI / GUI) Sessions?......................................................... 39
      2.2.3.3 How was Preference Determined?.............................................................................................. 41
  2.3 Apparatus: Hardware and Software Description..................................................................................... 45
  2.4 Additional Details.................................................................................................................................. 46
3 Problems, Solutions and Quizzes............................................................................................................... 47
  3.1 Problem 1.A........................................................................................................................................... 47
  3.2 Problem 1.B........................................................................................................................................... 48
  3.3 Problem 2.A........................................................................................................................................... 49
  3.4 Problem 2.B........................................................................................................................................... 50
  3.5 Quiz Problem 1.A................................................................................................................................. 51
  3.6 Quiz Problem 1.B................................................................................................................................. 52
  3.7 Quiz Problem 2.A................................................................................................................................. 53
  3.8 Quiz Problem 2.B................................................................................................................................. 55
  3.9 Screen Captures - CAAD CUI & CAAD GUI...................................................................................... 57
4 GOALS, OPERATIONS, METHODS, SELECTION (GOMS) Analysis ................................................................. 58
  4.1 GOMS Analysis - Character User Interface ......................................................................................... 60
  4.2 GOMS Analysis - Graphical User Interface ......................................................................................... 77
  4.3 Results of GOMS Analysis ................................................................................................................. 97

5 Results .................................................................................................................................................. 100
  5.1 Observations ...................................................................................................................................... 104
  5.2 Summary of User Performance .......................................................................................................... 110
  5.3 Summary of User Preference ............................................................................................................. 113
  5.4 Summary of User Preference vs. Age and Gender ............................................................................. 117

6 General Discussion .............................................................................................................................. 120
  6.1 Summary of Findings ......................................................................................................................... 120
  6.2 Summary of Hypothesis Tested ......................................................................................................... 123
  6.3 Preference, Learning and Performance .............................................................................................. 130

7 A Reappraisal of the Findings from the CAAD Study ............................................................................ 135
  7.1 Two Interfaces for the Same Expert System ..................................................................................... 135
  7.2 Factors Affecting UI Preferences ....................................................................................................... 138

8 Conclusions .......................................................................................................................................... 142
  8.1 Recommendations for Developing Effective Interface Design ......................................................... 142
    8.1.1 Control ........................................................................................................................................ 145
    8.1.2 Directness ................................................................................................................................... 146
    8.1.3 Consistency .................................................................................................................................. 147
    8.1.4 Forgiveness ................................................................................................................................. 147
    8.1.5 Feedback ...................................................................................................................................... 148
    8.1.6 Aesthetics ..................................................................................................................................... 149
    8.1.7 Simplicity ...................................................................................................................................... 150
    8.1.8 Costs to the Developer: Dollars vs. Design Tradeoffs ................................................................. 151

9 Developing Expert Systems: Using the Results Presented in This Study .................................................. 154

References

Appendices

Appendix A - CAAD CUI Screen Captures
Appendix B - CAAD GUI Screen Captures
Appendix C - Experiment Data
Appendix D - Pearson Correlation Coefficients
An Exploration of the Facilitating Effects of Graphical User Interfaces (GUI) and Character User Interfaces (CUI) on the Design of Expert Systems

PREFACE

There are important principles and procedures that guide the development of the technologies of expert systems, and one of the most important considerations of commercial application developers is the consideration of utility. The maxim of developers with respect to expert systems (ES) is that “a system is only as good as it is useful.” Practically speaking, the primary value of any system resides in its continued usage. User acceptance of any expert system is directly related to its interface design and user preference. The present study examines an expert system with respect to user interface preference.

Many commercial expert systems are being developed in research centres throughout the world, but their development is typically conducted without reference to a sizeable user base. Little to no published data exists about the user acceptance studies of such systems.

What makes a system useful? If system usage is tied to the presentation layer of an operating system, as much of the literature holds, then end user acceptance of expert systems may be tied to the perceived “visual cues” underlying the system’s functionality. As such, user acceptance may depend on many variables, not just on the system’s decision-rendering abilities. The principles and procedures for development of expert systems is an important field of study, especially, the development of user interfaces -- that promote end user satisfaction and productivity. Significant questions include “What encourages a user to prefer one system over another?” “What are the factors underlying any individual’s productivity in a system?”
For this investigation, two expert systems were created by the author, CAAD (Computer-Assisted Automotive Assistant) CUI (Character User Interface) and CAAD GUI (Graphical User Interface). CAAD is specifically designed to address automotive problems, and the study explores the utility of this expert system in a real-world context. The study involves individuals who have brought their car to an automotive shop for servicing and who volunteered to use the CAAD system while waiting. CAAD allowed users to explore various automobile parts, including those of immediate diagnostic concern. Through exploration of CAAD, users were able to learn more about their presenting problem and the mechanic's diagnosis. These two systems are capable of solving the same problems in the area of automotive mechanics; the only real difference between them lies in the presentation layer of the operating system, i.e., the user interface. Given a choice between CAAD CUI and CAAD GUI, which interface would an automobile owner choose as the one to solve automotive problems and why? If these questions could be explained with some degree of statistical rigour and predictability, these findings could be of benefit to future developers of expert systems.

Many expert systems are used by non-homogeneous groups of individuals, i.e., people of differing ages, skills, educational experiences, and physical/visual abilities. CAAD CUI and CAAD GUI were set up in an automotive garage with the permission of the operators over one summer, eventually being tested on 250 people.

With these two systems in place, i.e., CAAD CUI and CAAD GUI, the intent was to study how an automobile owner would be able to recognize a solution to an automotive problem and assess whether the solution offered by a garage mechanic was reasonable. Because the systems employ the same underlying domain knowledge and heuristics, preference for one system over another, and rationales for those choices could prove useful to future systems developers. Included
in issues addressed are interface appearance, input device and facility, previous conditioning with either CUIs or GUIs, lack of graphical security, or preconceived notions (i.e., graphical user interfaces are more intuitive, friendly, forgiving or designed for the novice).

Upon examination of the literature, it was found that many designers hold the belief that individuals prefer a graphical user interface because of historical conditioning. The adage "A picture is worth a thousand words," suggests why people might prefer a GUI to CUI interface. In accordance with the idea that pictures best describe complex ideas, many people seem tacitly to accept GUIs over CUIs. This acceptance has been reinforced by the success of the Apple Macintosh (and other graphically driven personal computers). Still, a convincing answer to the question, "What makes a person more prone to accept one interface over another is an important field of study. The present study focuses on issues of preference related to experience, and also to gender and age. In the concluding section, recommendations and suggestions are offered to developers regarding the construction of front ends to expert systems.

At the time this study was conducted, given the introduction of Microsoft's Windows 3.0, the ability to understand the rationale for the abandonment of the command line interface was important and it remains important. Moreover, interface selection and development are important considerations to the larger "educational" community, because learning may be facilitated by giving users an interface that is intuitive, i.e., that facilitates movement, and that accomplishes end objectives without misdirection. Furthermore, expert systems are expensive to design, develop and implement. The development of expert systems requires an extensive technology "awareness" on the part of the developer, significant programming expertise, as well as in-depth domain knowledge. If users are to focus on content and the structure and application of knowledge warehoused within the framework of these expert systems, then understanding the interface which represents the access
route to this information is a critical first step.
1 Introduction

The graphical user interface (GUI) is one of the most revolutionary changes to occur in the evolution of modern computing systems. In the space of less than 10 years the expectation of what the interaction between human and computer would be like has changed from a terse, character-oriented exchange modeled on the teletypewriter to the now familiar Windows, Icons, Menus and Pointing device (WIMP) interface. This revolution has increased the accessibility and usability of computer systems to the general public and, in spite of some sparring between competing user interface (UI) camps, created a commonality of use across platforms that has gone a long way toward decreasing the learning curve when moving from system to system. The world of UI's is neither perfect, complete, nor static. There remains much to be done in terms of increasing the productivity of computer users, standardizing operations across disparate architectures and adapting the human-computer interface to nontraditional applications. (Mandelkern, 1993, p.36)

Research shows that a command line interface places a heavy cognitive load on all users and in particular on novice users. Recent innovations in human computer interaction and interface design suggest that systems based on windows, icons, menus and pointers are more user friendly than those based on characters (Minasi, 1995; Norman, 1995; Mandel, 1994; Bowerman, 1992). Generating a theory about the user-friendliness of such interfaces is one thing, but proving user friendliness is quite another (Norman, 1995).

There is a growing body of literature that stresses the many benefits of the Graphical User Interface (GUI). Recent sales of such GUI products as Windows, Win 95, OS2, OpenLook, Motif, and Systems 7 similarly reflect this preference. User interface is an area of interaction in which there is a great potential for improvement (Norman, 1995). GUI's are a large investment for individuals, businesses, and corporations, considering the investment in hardware, software, training and support. Investors expect that the public has the time and inclination to learn and use them (Mandel, 1994).
The focus of this thesis is an investigation of responses to graphical user interfaces (GUI) and character user interfaces (CUI). To facilitate this investigation, a rule-based expert system (ES) was developed to diagnose automobile failures. This system may be operated under a character user interface or under a graphical user interface. There are several reasons why this application area was selected. First, rule-based expert systems are an increasingly important area of interest to researchers (Brown & O'Leary, 1995; Watkins & Elliot, 1993; Vasarhelyi & Srinidhi, 1993; Davies, 1984). This is largely due to the proliferation of various Personal Computer (PC) based expert system shells (Tway & Riedel, 1996; Stelzer, 1995).

Second, although there is much interest in expert systems (Goodner & Markowitz, 1996; Dooley, 1996; Harriger, 1995; Crespoan, Carter, & Gibson, 1994; 1994; Huang & Brandon, 1993; Harmon, Maus & Morrissey, 1988; Kearsley, 1987; Whitney, 1987), for the most part, little attention has been given to the environments / interfaces under which these applications are implemented (Norman, 1995; Clancey & Soloway, 1990). The evolution of expert systems is making interface design a more critical factor in the overall development of the system. This is due in part to the increasing size and complexity of the systems. As well, since expert systems are continuing to evolve, the requirements for user interfaces have not yet stabilized (Stelzer & Williams, 1988).

Third, much of the early expert systems research was directed to the development of different knowledge representations and inferencing procedures (Goodner & Markowitz, 1996; Dooley, 1996; Harriger, 1995; Crespoan, Carter, & Gibson, 1994; Gevarter, 1984; Bowerman & Glover, 1988; Liebowitz, 1988; Rolston, 1988; Holtzman, 1989; Pedersen, 1989), "[Yet], considerably less attention [has been] paid to the user interface" (Berry & Hart, 1990, p. 153).
"Interfaces of the future will concentrate much more on the human aspect of user interfaces" (Mandel, 1994, p. 296).

Clearly, the human/computer interface is a rapidly changing surface (Douglas, Doerry & Novick, 1992; Nielsen, 1993; Cowan, Durance, Giguere & Pianosi, 1993). Experts agree to disagree about the future of this "surface". It is not clear what this surface will ultimately be; nevertheless, it is clear that it is important. Indeed, the balance of power of the computer rests at the human/computer interface. In the years to come, research and development of the interface will be a major thrust in bringing about the next wave in computer revolution of society (Norman, 1991, p. 322).

Well-designed expert systems can speed up human work by a factor of ten, and sometimes by a factor of a hundred or more (Engelmore & Feigenbaum, 1995); yet such productivity gains will only be realized through the implementation of an appropriate user interface.

1.1 Statement of the Problem

The purpose of this thesis is to study responses to a Graphical User Interface (GUI) and a Character User Interface (CUI) in the area of diagnostic expert systems-based instruction specific to the analysis of automotive failures and repairs.

1.2 Goal of this Research

Through the study conducted in this thesis, the author will demonstrate the following:
(1) Inexperienced computer users prefer using systems equipped with a graphical user interface (GUI) than systems with a character user interface.

(2) As the user's level of computer experience increases, there is a shift in preference toward using a system equipped with a character user interface (CUI).

   If a developer expects his/her application to be used by individuals who have little computer experience, then consideration for the implementation of a GUI might prove worthwhile. This is especially true if intended use of this system is infrequent.

   “GUI's make applications easier to use and thus decrease the need for training in the long run” (Fitzgerald & Hildebrand, 1993, p. 66). Additionally, if a developer knows that the intended audience has a significant amount of computer experience, then providing them with a CUI may be the better choice.

   In the larger framework of systems design, regardless of purpose, if the developer wants user comfort, then it is worth the time, money and effort to develop a GUI front-end largely because inexperienced users will benefit from it, whether the developer's application is diagnostic in nature or an executive information system or an information system provided to shoppers in a shopping mall.

(3) Given lots of computer experience, people are equally productive on either interface.

   A GUI may only be necessary for novice users.
1.3 Hypotheses

To explain the goals previously cited, this thesis will formally examine the following four hypotheses:

(1) Users with no computer experience will prefer using a GUI equipped expert system over the same CUI equipped expert system.

(2) Users with lots of computer experience will prefer using a CUI equipped expert system to the same GUI equipped expert system.

(3) Users with GUI experience will be more productive using a GUI equipped expert system than with the same CUI equipped expert system.

(4) Users with GUI experience will make fewer errors using a GUI equipped expert system than with the same CUI equipped expert system.

Additionally, this thesis will explore relationships that might exist in the current data set between gender and interface preference, as well as between age and interface preference. These hypotheses will further be considered in relation to the individual’s learning experiences under graphical and character user interfaces.
1.4 Problem Environment

Consider the case of an individual who, while driving to work, encounters automotive problems. Taking the vehicle to a garage, the individual is confronted by a mechanic who diagnoses the problem in terms too difficult for a lay person to comprehend. Having neither the time nor the knowledge to pursue the matter at this point, the individual elects to leave the car sitting idle in the garage parking lot. Eventually, the individual must make a decision regarding the mechanic's suggested course of action.

To deal with such a dilemma, a highly detailed expert system called Computer-Assisted Automotive Assistant (CAAD) has been developed. CAAD can be operated under both a character-based user interface (CUI) and a graphics-based user interface (GUI).

Since many types of automotive failures are possible, CAAD's domain knowledge has been confined to General Motors, Ford, and Chrysler products.

1.5 Types of User Interfaces

A user interface refers to those aspects of a system that the user refers to, perceives, knows and understands (Bass & Coutaz, 1989). UI's may be either character-based or graphic-based (Norman, 1991; Vassiliou, 1982).

1.6 Character User Interfaces (CUI)

CUI's fall into two groups. The first group is command line (considered the traditional style of computer-human interaction where prompting and instructions provided by the computer are in terms of a simple 'ready' prompt, asterisk, or other special symbol (Mayhew, 1992).
The second group is menu-driven (Smith & Mosier, 1986). “The command line interface was the first interactive dialogue style to be commonly used and, in spite of the availability of menu-driven interfaces, it is still widely used” (Dix, Finlay, Abowd & Beale, 1993, p. 102).

1.6.1 Menu-driven CUI

In contrast to the “command line” style of interaction, menu-driven CUI may use elaborate prompts in the form of complex screen structures. Five basic structures provide the foundation for menu-driven CUI’s (Smith & Mosier, 1986; Hix & Hartson, 1993). Specifically, these five are as follows: (i) static menuing; (ii) bounce-bar menuing; (iii) layered menuing; (iv) point-and-shoot menuing; and (v) windowing environments, not to be confused with ‘multitasking’ (Zelkowitz, 1988).
Static menus present a fixed set of options in the form of a list from which to select an action (see Figure 1.6.1-1). These actions are static in the sense that they depend on a predetermined screen design. Selections are generally made by typing an option's name, number, or letter.

![Static Menuing Diagram](image)

*Figure 1.6.1-1 Static Menu*
Bounce-bar menus, like static menus, present a fixed set of selection options to the user in the form of a menu (see Figure 1.6.1-2). The user may make a selection by using the cursor keys to highlight a specific action from the list.

```
Static Menuing
Option
1. Selection one
2. Selection two
3. Selection three
N. Selection N
```

Figure 1.6.1-2 Bounce-bar Menu
Layered menus (see Figure 1.6.1-3) are two or more menus. They may be either a series of static menus or a series of bounce-bar menus in any combination. As the user makes a choice from one menu, an action takes place, eventually resulting in another menu presentation overlaying the previous menu. This cycle occurs throughout the interaction.
Point-and-shoot menus (see Figure 1.6.1-4) are composites of the previous menu structures. However, selections are made possible by the use of some form of pointing device. Once a desired menu has been pointed to (activated), a secondary pull-down menu provides the selection options. The secondary options on the pull-down menu may be selected in the same manner.

![Point-and-Shoot Menus](image)

Figure 1.6.1-4 Point-and-Shoot Menus
CUI window environments that are menu-driven direct interaction around a series of character-based menus (any format) to provide access to multiple windows. Interaction is achieved through cursor key-controlled movements (see Figure 1.6.1-5).
1.7 Graphical User Interfaces (GUI)

The graphical user interface has been variously defined. Here are two definitions:

"Some users think that if you want windows, mice, and fancy pop-up menus, you must have a graphical user interface" (Kelly-Boote, 1989, p.125; Dix, Finlay, Abowd & Beale, 1993, pp.107-9)

According to Microsoft, a GUI must meet each of the following requirements (Townsend, 1990, p.7):

1. The screen is a true representation of what will be printed.
2. The user interface is graphically oriented, making extensive use of icons.
3. Standard elements (resources) are used across all applications. Examples include menus, dialogue boxes, and windows.

For the purposes of this study, a GUI is an icon based, mouse-driven, graphical windowing system that makes use of dialogue boxes and pop-up menus to convey information to the end-user.

Similarly, a CUI is a keyboard-driven, non-graphical, non-windowing, menu-based system that makes use of command line entries as a means of system interaction.

---

This interaction style is in use with the vast majority of computer systems around the world today and is occasionally referenced in both popular and academic literature as the "WIMP" (windows, icons, menus and pointer) interface. Windows (proper) are separate areas of the screen which behave as independent terminals, which can be re-sized, scrolled and opened (one or more) at any one time within an application. Icons are used as schematic representations of system components, such as system applications, utilities, or events. Menus provide a list of options representing operations or services which can be performed by the system. Menus generally contain a short number of items to avoid confusion. Cascading (or layered) menus are employed to allow successive refinement of the selection process. A pointer is an input device which supports a selection style of interaction, such as a mouse, track-ball, glide-pad, touch-stick, or pen. Features commonly associated with WIMP environments include palettes, buttons, dialogue boxes, and scroll bars, and are commonly referred to as controls.
1.5 Current Research on GUI vs. CUI Usage

From the designer’s perspective, the primary function of the human/computer interface is to provide users with an efficient means of controlling a complex process.

It is too often the case that the encounter with the end user of the product is the last stage in the development of new technologies. The challenge of today’s software is to develop a product that can be easily and efficiently used to accomplish its intended purpose. The key element is to design an interface between the user and the machine that allows for the proper flow of control between human procedures and machine processes. This is particularly true of ES-based technology.

Ledgard, Singer, and Whiteside (1981) point out the high costs of poor human engineering in the design of interactive systems. They identify four areas of cost:

1) The direct costs of poor design are observed in wasted time and excessive errors. The system itself may have been designed for maximum efficiency in terms of memory management, input/output, and computation, but it may sit idly by as the user ponders an error message or thumbs through documentation looking for the correct command. Menu selection systems may alleviate such wasted time by self-documenting the system, and they may reduce errors by limiting user input to only the set of legal options.

2) Indirect costs are incurred through the allocation of time and training necessary to learn a system. A novice user often has a great deal to learn about a system. This typically involves reading manuals, working through tutorials, memorizing commands, and developing performance skills. The time to learn such systems can involve a great cost that must be born by the employee, the employer, or both. For the employer, the cost of training employees is particularly important where employee turnover is high, often itself
an indirect result of poor human engineering due to a third area of cost.

(3) This third cost is paid when users are frustrated and irritated by a poorly designed system. The psychological cost can result in a number of spin-off problems such as low morale, decreased productivity, and high employee turnover rate. It has been said that systems should be fun to use rather than frustrating. They should encourage exploration rather than intimidate users and inhibit use.

(4) Limited use or lack of use is costly. Poorly designed systems, no matter how powerful, will simply not be used. Users will tend to employ only those components that are of use to them. It is typically the case that for systems with 40 plus commands, only about 7 commands show any frequency of use. Limited use itself is a cost, as it negates potential benefits.

Since the turn of the twentieth century, psychologists and engineers have been interested in studying the generic man/machine problem in order to reduce costs in all areas. From the designer’s perspective, systems will not be purchased if the cost of their use negates the perceived benefits to their audience. This fact alone has driven some of the research into UI preference.

Interface evaluation and comparison have, in the past, mainly concentrated on investigations into aspects of specific interaction methods. “The field has expended very little effort in direct experimental comparison between systems that are functionally identical but which have different user interfaces” (Morgan, Morris & Gibbs, 1991, p. 265). Morgan et al., studied computer users’ performance and preferences using both a CUI and GUI version of a functionally equivalent data query language as the target vehicle for the study (Morgan, Morris & Gibbs, 1991, p. 265).

The research of Morgan et al., has led to the postulation of the following hypotheses:
Hypothesis 1. Session throughput will be slower using a GUI interface.

Hypothesis 2. Session time will be longer using a GUI interface.

Hypothesis 3. Naive users will prefer the GUI interface.

Hypothesis 4. Once competence is reached using the CUI interface, it will be preferred by both experienced and naive users.

Hypothesis 5. Error rates will be greater in the CUI interface.

After the completion of their study, Morgan et al., came to the following conclusions:

Result 1. Session throughput was not slower using a GUI interface.

Result 2. Session times were shorter using a GUI interface.

Result 3. Naive users did prefer the GUI interface. Subjects felt that the GUI was more powerful and were attracted to it.

Result 4. There is strong evidence against the belief that once competence is reached in a CUI interface, it will be preferred by both experienced and naive users.

Result 5. Error rates were greater using the CUI interface.

They recommended that studies continue with different populations in different areas.

Moreover, Morgan et al., advocate the need for further research to see if the GUI interface’s superiority will be maintained with use over a period of time. Scowen (1992) endorsed the view that additional research should be conducted in non-traditional application areas [e.g., expert systems].
The results of the study conducted by Morgan et al. must be viewed in light of the following limitations:

1. The study employed "naive" computer users only. The question that naturally follows is, "Would their results have been the same if a wider audience had been studied?"

2. The study employed only first-year, undergraduate business students. This is a fairly homogeneous sample. First-year business students may have "shared, common experiences" that drive the results in a given direction. The question that naturally follows is, "Would their results have been the same if a wider audience had been sampled, i.e., some business students, some arts students, some graduate students, some faculty, etc.?"

Mandel (1994) discusses some of the many studies that have investigated the benefits of GUIs compared with those of their predecessors, CUIs. A leading study that shows the benefits of the graphical user interface was conducted by Temple, Barker & Sloane, 1990. It was commissioned by Microsoft and Zenith. This study addressed the question of the effectiveness of graphical-user interfaces compared with character-based user interfaces. The study involved both experienced and novice users of word processing and spreadsheet products. Users results were analyzed for speed and accuracy, and users themselves rated their subjective experiences both during and after the study. The research results supported seven benefits of GUIs over CUIs. The benefits were:

1. GUI users worked faster.
2. GUI users completed more tasks.
3. GUI users have higher productivity.
4. GUI users expressed lower frustration.
5. GUI users perceived lower fatigue.
(6) GUI users were better able to self-teach and explore applications.

(7) GUI users were better able to learn more capabilities of applications.

The results of this focused study may be seen to be somewhat self-serving. As Mandel points out, "Throughout the 1990s, many businesses have found it very difficult to find any increase in actual user productivity that could be even remotely attributed to GUIs" (p. 218).

1.8.1 The Influence of User Skill on Interface Preference

Researchers are divided on the issue of the influence of user skill on interface preference. On one hand, a group of researchers support the view that user skill level does have a profound influence on the selection of interface style. Novices prefer a GUI. As proficiency increases, users will prefer a more rapid and productive interface with less to look at on the screen (Ben Shneiderman, 1991, p. 340). On the other hand, researchers such as Prummer, Zapf, Brodbeck & Frese, 1992 support the opposite view and argue that user skill has little influence on the selection of interface style.

1.8.2 GUI Preference and Computer Experience

Why do individuals prefer GUI's?

Ferraro (1989) conducted a reading comprehension experiment on 48 undergraduates. According to the results of his study, the computer user's question-answer ability was affected by both rate of text presentation and text difficulty. Text which was presented one word at a time was comprehended better than text presented in multiword blocks or chunks. The appeal of a GUI may be related to reading skill and to a more general preference for presentation in small chunks.
Moreover, for new users, the "visual interest" of a GUI may offer an added dimension of appeal. When users are exposed to a new GUI for the first time, they may be interested in the fact that there is a "difference" in appearance (and consequently in "feel") to this interface. The layout, colours, and graphics may be considerably different from what they have seen in other applications. Therefore, they may prefer a GUI interface solely because of its difference and aesthetic appeal. The user may not even consider "functionality" when determining UI preference.

According to Purvy, et al. (1983), new users see a system that promotes a see/point/push-a-button style of interaction with immediate feedback as more effective than a command language interface. The benefits of such systems include an economy of concepts and effort for the user, along with the synergy of a unified environment.

Many users may already have the preconceived notion that GUIs are easier to use than CUIs. The fact that only one hand is used for selection, and only two fingers employed for clicking, may be attractive to some people. Keyboarding skills are incidental in most GUIs.

Blankenberger & Hahn (1991) studied the effect of icon design on human computer interaction. Their study examined the effects of articulatory differences (i.e., the difference between a picture and its meaning) on user performance in selection tasks. They concluded that icon designs had little influence on the performance of more experienced computer users and that the GUIs are aimed at serving the needs of the computer novice.

Extending Blankenberger & Hahn's contention that the design of a graphical user interface has a considerable impact on the preferences of inexperienced users, GUIs by their very nature often play host to a complex-looking set of graphics. This graphical array (and other screen "clutter") might suggest to the new user that the underlying application is "powerful". It may create the
...illusion that the user has many more options and functions available than there actually are. This "illusion" is particularly attractive to the new user in the short term, as with increased system exposure, the true extent of the application's functionality will show through despite the interface.

Easterby (1970), Lodding (1983) and Gittens (1986) argue that people find images natural because icons can be easily learned and recognized, and because images can possess more universality than text.

Scott (1991) says that iconic interfaces can reduce the learning curve in both time and effort, and facilitate user performance while reducing errors. This latter statement of Scott's is at issue in this thesis. Citing dualistic psychological theories, Scott argues that:

1. the visual and verbal functions of the brain are clearly differentiated between the hemispheres; and
2. individuals do not exploit the visual functions because people are predominantly verbal creatures.

Lastly, a large part of one's own educational experiences are fashioned around the idea of "learning by doing." The doing, whether it is "building a puzzle" as an elementary school student or "performing a chemistry experiment" as a secondary school student, may involve models; these models are often expressed pictorially. Hence a great deal of our prior conditioning may lay the foundation for preference or bias toward GUIs.
1.8.3 CUI Preference and Computer Experience

Why do individuals prefer CUI’s?

Dvorak & Seymour (1990) contend that GUIs are an old idea recently made fashionable again. In the past graphical interfaces were unfriendly and hard to learn. Pulling at a window, clicking on an icon, pointing at a dialogue box, and the other procedures result in delays in operational efficiency. Moreover, these actions demand understanding of the action of the application. It also takes a great deal of time to read the various menus and dialogue boxes to determine the next step (Morse & Reynolds, 1993).

Morse & Reynolds also develop the position that "market forces" lead interface developers to offer GUI front-ends to their applications. Moreover, these front-ends generally offer as many options to their customers as possible, thereby creating two classes of users: those who know how to use the tools/options and those who do not. Consequently, with GUI usage, even locating basic applications becomes a major task. Screens are often so filled with unrelated “bells and whistles” that the user’s attention becomes misdirected and unfocused, consequently, application prompts often go unnoticed. The entire process of pointing and clicking a mouse to select operations becomes tedious.

Proponents of the CUIs criticize GUIs for their:

1. Difficulty in basic pointer manipulation.
2. Difficulty in initiating the object-action paradigm, implemented in many GUIs.
3. Difficulty in understanding how to access an application.
4. Difficulty associating an icon with an application.
Difficulty due to the ambiguous nature of icon definitions. Icons are often times uninformative to the user; moreover, some concepts do not lend themselves to iconic representations.

GUIs also require "fine" motor skills that take time to learn and are more difficult to acquire as one gets older. This problem is exacerbated by the fact that GUIs, by definition, may use icons, dialogue boxes, windows, pull-downs, title bars, scroll bars, and a host of other features. These interface elements take up space in the user's viewing area. All this "screen real estate" (overhead) often reduces the size of the object in the interface. This smaller size makes access even more difficult, so individuals may opt for the CUI as the alternative. In addition, new computer users may have difficulties due to hand/eye coordination demands. This problem is amplified while using a GUI. Input movement is in the horizontal plane, while cursor movement is in the vertical plane. The transition between the two modes of reference is confusing for some users. This initial to short-term confusion may be enough to turn the individual's preference toward the CUI.

Moreover, because GUIs take up so much screen real estate, developers are forced to build their applications in multiple step tasks. This gives the illusion that GUIs are less powerful than CUIs because so many incidental steps are required along the way to get to one's end goal in the application. This is further complicated by the fact that inherent in the design premise of a GUI is the fact that GUIs provide "recognition" memory to the user in multi-levelled menus. These multi-levelled menus further contribute to the user's perception that GUIs are less powerful than CUIs.

Through research and practical experience, IBM usability professionals have compiled a list of the most common usability problems found across most graphical interfaces and object
oriented interfaces. The top ten usability problems in the list are as follows:²

(1) Ambiguous menus and icons
(2) Single direction languages
(3) Input and direct manipulation limits
(4) Highlighting and selection limitations
(5) Unclear step sequences
(6) More steps to manage the interface than to do tasks
(7) Complex linkage between and within applications
(8) Inadequate feedback and confirmation
(9) Lack of system anticipation and intelligence
(10) Inadequate error messages, help, tutorials, and documentation.

A study conducted by Lawrance & Matthews (1984) devoted to public online catalogue users and non-users came to four conclusions:

(1) The most important user characteristic linked with success in using online catalogues is frequent experience with the online catalogue.
(2) The kind of search being performed is important as a determinant of success and satisfaction.
(3) Users adapt their search approaches to the capabilities and limitations of the online catalogues used.
(4) The form and nature of training and user assistance are important, with users who received at least some initial training and assistance showing more satisfaction than those who did

² Mindel, 1994; page 313 provides additional elaboration.
They conclude that the level of user experience is the strongest determining factor on preference. Clearly, CUI exposure and familiarity is going to be greater with experienced computer users. If Lawrence & Matthews are correct, it follows that experienced computer users are likely to prefer CUIs due to their past exposure, training and familiarity.

Parenthetically, according to Virzi (1992), empirical research studies on user interface design need not involve many test subjects, because 80% of the usability problems are detected with four or five subjects; additional subjects are less and less likely to reveal new information, and the most severe usability problems are likely to have been encountered by the first few subjects.

1.8.4 The Influence of GUI Experience on Productivity and Error Rates

According to a study conducted by Rauterberg (1992), people with GUI experience were 51% more productive in the completion of their data entry tasks when using the GUI as compared with CUI. The study involved the use of 24 people, 6 experts and 6 novices on each of the GUIs and CUIs.

There are several reasons one may question Rauterberg’s findings, i.e., (1) his small sample size is not presented within the scope of any confidence limits, and (2) “heavy” data-driven tasks may already have a bias toward GUI front-ends based on the use of such tools as “hyperforms” and “smartforms”, etc.

There are many problems associated with character-based interfaces. Keyboard entry can be error-prone and tiresome, requiring the user to remember specific inputs to the system as well as to remember the state of the system being controlled (Morse & Reynolds, 1993).
GUIs may take more time to use than CUIs for the following five reasons: (1) the complex array of graphics and menu items requires additional time for the individual to find the option needed to perform one's tasks. The user may also be tempted to stray and explore the functions of other menu/option items. This leads to wasted time and hence longer periods required for performing tasks. (2) Menu/option items are not often self-explanatory, and therefore the user may choose an option/item that is not required. Once the user is aware of this, the user needs time to go back to the original position before progressing with the tasks. (3) The movement of the mouse along a surface takes time away from performing required functions. Often the mouse must be lifted and then placed back down again, particularly when the pointer is near the edge of the screen. (4) Many programs require double clicking of menu/option items before the task will be performed. The user may err by over- or under-clicking, which leads to errors and therefore more wasted time. Lastly, (5) as was mentioned previously, often the pointer is not placed in the correct position over the menu item/icon. Therefore, the user often pauses, thinking the program is processing the command, when in fact nothing has been initiated.

Graphical user interfaces are typical of the movement in the personal and professional computer worlds to remove the user from procedural responsibilities and to develop natural ways to understand and to abstract the data and processes which software systems manipulate (Zhang & Mendelson, 1983). Consequently, developers may believe that GUIs are a more effective interactive medium to help users with their cognitive processes.

However, some researchers believe that interface design cannot be effectively measured through scientific means. James (1983), in a review of Shneiderman's book on human-computer interaction states that
the trouble is that when we investigate errors on the user interface we are not in the
world of hard science, but in the area of emotional response, concerned with human
reactions to uncertainty and failure. In these stressful situations "scientific" tests
seem hardly relevant. The Hawthorne effect (that you cannot isolate behavioural
factors and measure the effects of changing them like parameters in a mathematical
equation, and also that people behave differently when they are watched, in any
case) is probably enough to invalidate any statistically significant measures which
can be calculated. (James, 1983, p. 220)

Douglas, et al. (1992), assert that the last decade has, without a doubt, placed the graphical
user interface in a superior position over the traditional text-based approach. The authors cite the
fact that, in most applications, users have found graphical interfaces easier to learn, faster to use,
and less error-prone. Yet, as was previously mentioned, if users have a preconceived notion that a
GUI is easier to operate, they may not pay as close attention to their input commands as they
normally would with a CUI. They may believe that a GUI is more powerful than it actually is, and
are therefore lax in checking their input accuracy. If this is true, GUI usage may be potentially more
prone to errors.

Yet as was cited previously, a GUI may assist users to visualize their decision options and
hence improve their productivity. Savage (1991) contends that GUIs may reveal the hidden
structure of data and information being used in an application, and may make abstract concepts
appear concrete and understandable.

Adding to this complication, some research has been conducted recently that disputes any
correlation between interface design and productivity. Mitta, et al. (1993), studied the impacts of
how combining various command structures (i.e., menu, button, keypad) and input devices (i.e.,
mouse, touch screen) impact on user performance. The application was an automated part
recognition system. The results of the study showed that the successful performance of required
user-system interaction tasks remains independent of interface design, and that none of the interface
designs had negative effects on human task performance. The users in this study also indicated a
preference for a button-style command structure.

1.8.5 Gender Influence and Interface Preference

There has been little research done in the area of gender influence and interface preference.

This study was conducted in an automotive repair environment. This profession and
domain are traditionally the preserve of the male, not the female. The majority of automotive repair
personnel are male; clients may be of either gender. Poculs (1991) investigated personality traits
that may affect human-computer interaction. Two of these include: (a) introversion-extroversion
and traits characterizing introversion-extroversion are related to many aspects of human-computer
interaction and (b) anxiety characterized by a threat to self-esteem facilitates or debilitates
performance contingent on specific aspects of the learning task.

Inadequate knowledge of a particular domain may cause anyone to become anxious or
stressed or to exhibit introversion-like responses. In contrast familiarly with the domain may result
in greater ease in execution of tasks. In the automotive repair environment, women may be at a
disadvantage vis-a-vis men, especially if they are new to the technology that is being used to present
a subject domain to which they are largely unaccustomed.

If a graphical user interface is more comfortable to work with than a character user interface
(Mitta, Avitsa & Foster, 1993), and if a direct manipulation interaction style (such as a GUI)
provides the most empowering environment for a computer user (Hartson & Weller, 1992), then
it is reasonable to assume that in this particular study of automotive repair, women will demonstrate
a greater preference than men for the graphics interface.

1.8.6 Age and Interface Preference

GUI-based programs are more frequent than CUI-based programs in current-day applications. Thus younger users have been exposed to graphics interfaces from a variety of new technologies, e.g., video games, virtual reality simulators etc. Often these products are accompanied by extensive marketing campaigns geared toward young people. It is possible that the younger people in this study entered with a bias toward graphics-based interfaces. Buchanan (1982) described 'stylized' expert systems of the video-game variety as narrow in their domain of expertise and in their assumptions about approach to problem solving, with style becoming more important than substance.

According to Briggs (1988), novices tend to create their own patterns of interaction with the machine. This conclusion came from a study of novice users on a word processing application. Users were assessed as to how well they were able to identify the knowledge they needed for learning a new system, as well as how well they could identify the specific types of information they required. Briggs concluded that in the absence of a suitable, generalized model of the application system, the users were unable to effectively structure their own learning experience, and made poor use of the little experience they did have.

It is possible to argue that familiarity with previous graphics products provides individuals with a "framework" in which to operate, thereby providing a model to structure their learning.

One might additionally argue that older individuals may be more apprehensive about new technologies and interfaces with which they are not familiar (e.g., expert systems). Their first
exposure to a computer system was more likely through the use of a character-based user interface, and therefore they may feel more comfortable conducting their computer-related tasks through this form of interaction.

Nielsen (1993) reiterated the latter point through another study. He noted that increasing use of GUI systems created problems for people already proficient in computer use. He stated that problems were

seen in the transfer from the previous generation of character-based interfaces to the current graphical interfaces. When graphical interfaces were first starting to see widespread use in the mid-1980s, I observed several experienced computer users being unable to rename files. They were used to having a separate operating system command for this operation and searched through all the system menus for a rename command. These users did not realize the new principle of generic commands made the change of a file name just one more instance of text editing to be accomplished by selecting the old name with the mouse and typing in the change. (Nielsen, 1993, p. 83)

Wenger (1991) conducted a study that explored the expectation for appropriate interaction, as it developed through human-computer interaction. The participants used either a direct manipulation or a command interface that, at an unexpected point in the interaction, displayed an error message that was either consistent or inconsistent with the interface’s previous pattern of interaction. The results of the study show that the direct manipulation interfaces were more likely to establish expectations in the users, and when they experienced inconsistent error messages, they expressed negative reactions to the computer’s output - hence showing a future CUI preference.
2 Study Description

2.1 Purpose

As detailed in Section 1.2, a diagnostic expert system (CAAD) was developed to diagnose automotive failures. CAAD, operating in either its CUI or its GUI implementation, provides two levels of diagnostics, i.e., quick trouble-shooting or detailed diagnostics.

CAAD provides trouble-shooting diagnostics in the following areas: (i) engine - mechanical; (ii) engine - performance; (iii) lighting - both interior and exterior; (iv) rack and pinion steering gear, and (v) steering with the exception of rack and pinion. In addition, a further level of highly detailed diagnostics is available for specific braking systems, front-end systems, and air-conditioning systems.

2.2 Method

As presented in section 1.2, consider the case of an individual who is in a hurry to drive to work and who experiences some degree of automotive failure. When challenged by a mechanic's trade-like explanation, the individual becomes perplexed. Should the car be repaired, not be repaired, be partially repaired, or receive alternative servicing? In other words, the motivation for this study is indeed a scenario familiar to most automobile owners.

To model this situation, two sample problems have been developed in the aforementioned areas - quick trouble-shooting and detailed diagnostics. These problems (listed with their anticipated solutions in Section 3) are intended to be representative of the diagnosis that the garage mechanic would offer the troubled motorist.
An actual automotive repair shop was used as the stage for data collection. Customers who came in off the street in need of an automotive repair were asked if they would like to participate in a study involving automotive repairs. For the most part, they were a "sample of convenience" while waiting for their own automobile's repair.

Step 1. Prior to participation, each participant was interviewed regarding his/her previous computer experience. Demographic variables such as (1) age, (2) gender, (3) amount of computer experience, and (4) degree of familiarity with GUIs are then collected.

Step 2. Each individual was asked to listen carefully to an audio tape of an automotive problem similar to the sort of problem that the individual was presently experiencing.

Step 3. The individual was asked to consult either the CUI version of CAAD or the GUI version of CAAD. To minimize bias, the CUI version and the GUI version were randomly4 presented from one individual to the next.

The goals of the consultation were to (1) furnish the user with an understanding of the mechanic’s diagnosis, with the causes of the failure, and with an insight into other related problems; and to (2) provide the user with the knowledge required to make an informed decision regarding the course of action.

4Computer experience may range from (a) lots of prior experience using computers, (b) moderate use of computers (i.e., word processing, spreadsheets, etc.), to (c) little or no previous use of computers.

4Familiarity may range from (a) lots of experience (i.e., Macintosh, Windows, Xwindows, etc.), (b) some exposure, to (c) little or no previous exposure.

4This was determined by the toss of a coin, i.e., on heads occurring, the individual was presented with the CUI version of CAAD first. On tails occurring, the individual was presented with the GUI version of CAAD first.
that the mechanic should pursue.

These goals were stressed so that each individual could comply with the purpose and intent of the study.

Step 4. The individual was then asked to take a short quiz regarding his/her understanding of the mechanic's solution.

Step 5. Measurements were conducted according to Section 2.2.1.

Step 6. The remaining audio tape was played to the participant.

Step 7. The individual was asked once again to consult the other implementation of the CAAD program, keeping in mind the same goals of the consultation for this new problem.

Step 8. The individual was then asked to take a short quiz regarding his/her understanding of the mechanic's solution.

Step 9. Measurements and evaluation was conducted according to Section 2.2.1.

Step 10. Lastly, the individual was interviewed regarding his/her convictions about the respective user interfaces, specifically:

(i) Which system he/she preferred to use, and why;

(ii) Which system he/she would be more likely to use, given the opportunity; and

(iii) Whether he/she found the system useful.
2.2.1 Measurement Criteria - Legend of Variables

The following measurements were conducted during this study.

Table 2.2.1-1 Description of Measurements by Variable Name

Note: Columns 1, 2, 3, 4, 5 and 19 common for both GUI and CUI sessions.

<table>
<thead>
<tr>
<th>C1</th>
<th>Range: Each individual's record number</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>Age: The age in years of the individual</td>
</tr>
<tr>
<td>C3</td>
<td>Gender: 0 = female, 1 = male</td>
</tr>
<tr>
<td>C4</td>
<td>Computer Experience: 0 = little / none, 1 = some, 2 = lots</td>
</tr>
<tr>
<td>C5</td>
<td>GUI Experience: 0 = little / none, 1 = some, 2 = lots</td>
</tr>
<tr>
<td>C6</td>
<td>Time: Time in minutes to complete task</td>
</tr>
<tr>
<td>C7</td>
<td>%-task: Percentage of Task Completed: 0 = little / none, 1 = some, 2 = lots</td>
</tr>
<tr>
<td>C8</td>
<td>Task/time: Percentage of Task Completed per unit of time</td>
</tr>
<tr>
<td>C9</td>
<td>Task-err: Time in minutes spent in errors</td>
</tr>
<tr>
<td>C10</td>
<td>%task-err: Percentage of task time spent in errors</td>
</tr>
<tr>
<td>C11</td>
<td>Commands: Number of commands used</td>
</tr>
<tr>
<td>C12</td>
<td>Help: The number of times help function was used</td>
</tr>
<tr>
<td>C13</td>
<td>Time-help: Time in minutes spent in help</td>
</tr>
<tr>
<td>C14</td>
<td>%fav/unf: The % of favourable commands to unfavourable commands</td>
</tr>
<tr>
<td>C15</td>
<td>Failed: The total number of repetitions of failed commands</td>
</tr>
<tr>
<td>C16</td>
<td>Mislead: The number of times the interface misled the user</td>
</tr>
<tr>
<td>C17</td>
<td>Regress: The number of regressive commands issued</td>
</tr>
<tr>
<td>C18</td>
<td>Control: The number of times the user lost control of the system</td>
</tr>
<tr>
<td>C19</td>
<td>Pref: Expressed system preference, 0 = GUI, 1 = CUI, 2 = None</td>
</tr>
</tbody>
</table>

GUI and CUI data are presented numerically according to these column descriptions in the appendix supplied with this thesis.
2.2.2 Measurement Descriptions

The following is a discussion of how each measurement was determined.

<table>
<thead>
<tr>
<th>C1</th>
<th>[Range] Assigned successive number starting at 1, incrementing by 1, going to 250.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>[Age] Orally asked for the participant's age</td>
</tr>
<tr>
<td>C3</td>
<td>[Sex] Observed (male / female)</td>
</tr>
<tr>
<td>C4</td>
<td>[Computer Experience] 0 = little / none, 1 = some, 2 = lots</td>
</tr>
</tbody>
</table>

2.2.2.1 How was Computer Experience determined?

Individuals were categorized with respect to computer experience in the aforementioned areas by asking a number of questions. The order of the questions varied somewhat in the attempt to foster a natural conversation, one that would ease the individual into the tasks ahead. Sample dialogues are as indicated below.

**Dialogue One**

<table>
<thead>
<tr>
<th>(WJ)</th>
<th>Hello, how are you today?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RS)</td>
<td>Fine thanks.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Do you have a computer at home?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Yes I do, but it's the kids'.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Do you ever use it?</td>
</tr>
<tr>
<td>(RS)</td>
<td>No, I just watch them play games on it - it looks fun, however.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>In the sort of work that you do, do you ever use a computer?</td>
</tr>
</tbody>
</table>
Placement - Computer Experience = 0, none

Dialogue Two

(RS) No, I can't even type.

(WJ) Hello, how are you today?
(RS) Oh, not bad - looks like you've got some good things here.
(WJ) Oh yes, these computers are for the study that I'm conducting.
(RS) They are very difficult - they cost much?
(WJ) Ah yes, they are expensive - do you have a computer?
(RS) No, I don't know them.

Placement - Computer Experience = 0, none [typical of language problem]

Dialogue Three

(WJ) Hello, how are you today?
(RS) Excellent.
(WJ) What sort of work are you in?
(RS) I'm a secretary.
(WJ) Do you use WordPerfect at your office?
(RS) Yes we do - version 5.1.
(WJ) Are there any other applications that you use the computer for?
(RS) Yes, we use Lotus 1-2-3 a fair bit, but I only use WordPerfect - but I would like to try Lotus 1-2-3.

Placement - Computer Experience = 1, some [some computer exposure, no GUI exposure]
Dialogue Four

<table>
<thead>
<tr>
<th>(WJ)</th>
<th>Hello, how are you today?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RS)</td>
<td>Oh, fine thanks. Looks like you’ve got some nice equipment there - 386s?</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Actually yes - I see you know your computers - do you have a computer of your own?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Yes, I do - a 386 DX/33 with a 120 Mb. Hard-drive - running windows 3.1. Do you have windows on these boxes?</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Yes, I do - so do you run mostly windows applications?</td>
</tr>
<tr>
<td>(RS)</td>
<td>No actually, I run both equally - it really depends on what I’m trying to do.</td>
</tr>
</tbody>
</table>

Placement - Computer Experience = 2, Lots [Lots of computer exposure, strong GUI exposure]

Dialogue Five

<table>
<thead>
<tr>
<th>(WJ)</th>
<th>Hello, how are you today?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RS)</td>
<td>Not bad - looks like you are a PC fan.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Well, I guess you may say that - do you have a PC?</td>
</tr>
<tr>
<td>(RS)</td>
<td>No - we’ve got Macs at work.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>I see - do you use the Mac?</td>
</tr>
<tr>
<td>(RS)</td>
<td>A bit - but mostly for word processing.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Do you have a PC or a Mac at home?</td>
</tr>
<tr>
<td>(RS)</td>
<td>No.</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Have you ever tried a PC?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Once or twice for Lotus.</td>
</tr>
</tbody>
</table>

Placement - Computer Experience = 1, Some, [Exposure to GUIs 2 (lots)]. No special statistics were kept on Mac vs. PC users. If a user had GUI experience on any platform, this would give at least a GUI score of 1 and depending on the nature of that exposure a 2.
2.2.3 Measurement Descriptions Continued

<table>
<thead>
<tr>
<th>C5</th>
<th>[GUI Experience] 0 = little/none, 1 = some, 2 = lots (determined from the previous conversations)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>C6</th>
<th>[Time to complete task]</th>
</tr>
</thead>
</table>

This measurement includes the total time (measured in decimal minutes) from the first keystroke (or mouse click) to the completion questions relevant to the content of the expert system (see Section 3 and following for a detailed description of these problems). In the event that a user cannot reach this terminal point in the CAAD system, task completion time would be measured to include the point where the user believes he/she has completed all that is within the scope of his/her abilities. Time was based on an TSR clock.

<table>
<thead>
<tr>
<th>C7</th>
<th>[Percentage of task completed]</th>
</tr>
</thead>
</table>

If completion of problems relevant to the content of the expert system (see Section 3.1 to Section 3.4; specifically problems 1A, 1B, 2A and 2B) represents 100% of the task completed, then completion of either problem group (1 or 2 exclusively) would represent 50% of the task completed, and so on. This was coded as a discrete value in the range (0, 1 or 2), where 0 represented little to none of the task completed, 1 represented some of the task, and 2 represented most of all of the task completed.
This measurement is defined as the percentage of the task completed divided by the total time to complete the task.

| C9 | Time spent in errors |

This measurement is defined as the total time expended while:

(a) Re-issuing failed commands.
(b) Pursuing an incorrect branch/options.
(c) Issuing regressive commands.

Note: The minimum spanning tree, i.e., the shortest path to complete either problems 1.A and 2.A or problems 1.B and 2.B is 22 selection options in either case.
2.2.3.1 How was an Error Determined (CUI / GUI) Sessions?

| Example One: | If the critical path was A -> B -> C and the user went A -> B -> A -> B -> C, then a regressive error was made and the time off the critical path was calculated. |
| Example Two: | If the critical path was A -> B -> C and the user went A -> A -> B -> C, then a command was re-issued and the time in the redundant command was calculated. |
| Example Three: | If the critical path was A -> B -> C and the user went A -> B -> D, then a misdirection was made and the time off the critical path was calculated (until the individual returned back to C). |

Note: With respect to GUI sessions, A, B, and C may represent activities such as icon selection, graphic selection, etc., as opposed to command line entries made under CUI sessions.

\[ CI_0 \] [Percentage time in errors]

This measurement is defined as the total time spent in errors divided by the total task completion time.

\[ CI_1 \] [Number of commands used]

This measurement is defined as the total number of selection options exercised. Selection options equal required commands plus number of errors.

\[ CI_2 \] [Frequency of help used]
This measurement is defined as the total number of times the user calls on the help facility.

C13 | [Time spent in help]

This measurement is defined as the cumulative time spent by the user accessing the help facility.

C14 | [Percentage of favourable/unfavourable user commands]

This measurement is defined as 22 (minimum span) divided by the total number of selection options exercised.

C15 | [Number of repetitions of failed commands]

This measurement is defined as the total number of times a user attempts to repeat a failed command.

C16 | [Number of times the interface misleads the user]

This measurement is defined as the total number of times the user embarks on a path completely unrelated to the nature of the problem under study.
This measurement is defined as the total number of regressive commands issued by the user.

Note: Given a critical path depicted by the progression M1 -> M2 -> M3 ... -> Mn, movement from M3 -> M2 would be defined as a regressive command.

This measurement is defined as the total number of times the user needs external help to reinstate the system.

2.2.3.3 How was Preference Determined?

Upon completion of the CAAD sessions, users were asked to express their views about the systems with the intent of finding out which UI they liked the best, should they wish to routinely solve these sorts of problems. Coded as 0 = expressed GUI preference, 1 = expressed CUI preference, and 2 no expressed preference.

Individuals were categorized in the aforementioned areas by way of administering a number of orally asked questions. The order was varied somewhat to foster a natural conversation. Sample dialogues are as indicated.
Most Common Sample Dialogue - Expressed Preference for GUI

Dialogue One

<table>
<thead>
<tr>
<th></th>
<th>[Common introduction for all dialogues one, two and three following]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(WJ)</td>
<td>Well, thank you so much for helping me out with my study.</td>
</tr>
<tr>
<td>(RS)</td>
<td>Some typical responses included:</td>
</tr>
<tr>
<td></td>
<td>&quot;No problem - it was actually quite fun.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;My pleasure.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I didn't know it was going to take that long.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I didn't know it was going to take that long, but that's okay.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;It must have taken you a long time to make those programs.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Are you going to sell these programs to the garage or anybody?&quot;</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Well, if you were going to use either of these computer systems to routinely solve similar mechanical problems, which one would it be?</td>
</tr>
<tr>
<td>(RS)</td>
<td>I'd choose the first one. (This assumes the individual has indicated the GUI; because the individuals alternated between running through the GUI or the CUI first, the alternative response would be &quot;I'd choose the second one.&quot;)</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Well, why did you choose that one?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Some typical responses included:</td>
</tr>
<tr>
<td></td>
<td>&quot;It was more exciting.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;It was easier to use.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I liked the car picture(s).&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I liked using the mouse.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;It was more like a game.&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I don't know - I just liked it better than the other one.&quot;</td>
</tr>
</tbody>
</table>
Most Common Sample Dialogue - Expressed Preference for CUI

Dialogue Two

<table>
<thead>
<tr>
<th></th>
<th>[Common introduction]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(WJ)</td>
<td>Well, if you were going to use either of these computer systems to routinely solve similar mechanical problems, which one would it be?</td>
<td></td>
</tr>
<tr>
<td>(RS)</td>
<td>I'd choose the first one. (This assumes the individual has indicated the CUI; because the individuals alternated between running through the GUI or the CUI first, the alternative response would be &quot;I'd choose the second one.&quot;)</td>
<td></td>
</tr>
<tr>
<td>(WJ)</td>
<td>Well, why did you choose that one?</td>
<td></td>
</tr>
<tr>
<td>(RS)</td>
<td>Some typical responses included:</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;I didn't like using the mouse.&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;It was difficult to position that square [cursor] on the car [or menus].&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;It looked better.&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;I think it was easier to use.&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;It seemed to be faster to use.&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;It seemed like most programs I've used.&quot;</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>&quot;I don't know - I just liked it better than the other one.&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Most Common Sample Dialogue - No Expressed Interface Preference

Dialogue Three

<table>
<thead>
<tr>
<th></th>
<th>[Common introduction]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(WJ)</td>
<td>Well, if you were going to use either of these computer systems to routinely solve similar mechanical problems, which one would it be?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Some typical responses included:</td>
</tr>
<tr>
<td>-</td>
<td>&quot;I don’t know, I liked them both equally.&quot;</td>
</tr>
<tr>
<td>-</td>
<td>&quot;They were both good, I can’t choose.&quot;</td>
</tr>
<tr>
<td>-</td>
<td>&quot;I don’t know.&quot;</td>
</tr>
<tr>
<td>(WJ)</td>
<td>Well, isn’t there one that you like just a bit better?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Some typical responses included:</td>
</tr>
<tr>
<td>-</td>
<td>&quot;Well I guess the first one.&quot; [Once again this would be appropriately interpreted and coded.]</td>
</tr>
<tr>
<td>-</td>
<td>&quot;Not really.&quot;</td>
</tr>
<tr>
<td>-</td>
<td>&quot;No.&quot;</td>
</tr>
<tr>
<td>-</td>
<td>&quot;No they are both good.&quot;</td>
</tr>
<tr>
<td>(WJ)</td>
<td>(Sometimes, out of curiosity and if the individual seemed eager to talk) - What did you like about the each program?</td>
</tr>
<tr>
<td>(RS)</td>
<td>Typical responses much like the previous comments cited.</td>
</tr>
</tbody>
</table>
2.3 Apparatus: Hardware and Software Description

In order to study every individual in a timely efficient fashion, the following hardware was employed:

Two, 80386-33 MHZ workstations (each) equipped with: VGA graphics, 8 megabytes RAM, 80 MB/12 ms hard drive, with 32 K cached controller and 4 MB disk cache.

All necessary software was loaded on each specific system's hard-drive. Additionally, one of the workstations was equipped with: Windows 3.0 and a Microsoft mouse.

On this system resided the GUI-based version of CAAD. No software installation was required by the end user. The systems both resided in a ready state for immediate end user interaction.

To facilitate the tracking of each specific user, a memory resident program called SuperKey was loaded into both systems' memory. SuperKey is normally used to create macros to allow the user to customize the working environment. These macros are most often employed to automate menu-driven selections.

Specific to this study, SuperKey can perform a more significant task. By using a counter in each system's autotrace program, each new user can be assigned a key from the keyboard. Thus, the first user to begin interaction with either of the workstations will be assigned the key "A". Essentially, all user inputs will then be trapped into a file called "A.mac". This cycle is repeated for users B, C, D, etc. The output is like a macro written in Lotus 1-2-3. The output is trapped to a standard ASCII file and is highly readable. Upon completion of the session, SuperKey issues a "stop recording" command to the macro, thus saving the text file. This procedure by itself cannot
track the time involved with respect to completing any specific task. To manage this activity, a short memory resident program was written to issue a start and stop time for any session. Unfortunately, specific procedural times could not be managed by this short program, so it was necessary to separately analyze time requirements for specific sub-tasks such as specific task-completion time, time spent requesting help, and time spent correcting errors or repeating unnecessary procedures.

2.4 Additional Details

Explanations of the basic operation of both user interfaces were given to all participants. Examples included the "double-click" (object/attribute paradigm) for the mouse used in the GUI and the menu-cycling capability used in the CUI system.

Note: The knowledge bases for CAAD - CUI and CAAD - GUI are identical. They are structured, they follow a top-down flow, and they are highly readable. Neither the 400 plus pages of heuristics nor the 700 plus additional pages of programming source code is supplied with this thesis.
3 Problems, Solutions and Quizzes

As was detailed in Section 2.2, participants were randomly assigned (by coin flip) a version of CAAD with which to work. Next, by random assignment (coin flip), individuals were given either problems 1.A and 2.A or 1.B and 2.B to solve with this version of CAAD. If for example, the coin toss required the individual to solve problems 1.A and 2.A with the CUI version of CAAD, during the next session, the individual would be asked to solve problems 1.B and 2.B with the GUI version of CAAD.

3.1 Problem 1.A

Driving in to work you notice that your oil pressure gauge has risen to read "high". Your mechanic informs you that "we'll have to remove and inspect the oil pressure relief valve assembly." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem.

ANTICIPATED SOLUTION (In Order of Likely Occurrence)

<table>
<thead>
<tr>
<th>POSSIBLE CAUSE</th>
<th>CORRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Improper oil viscosity.</td>
<td>Drain and refill crankcase with correct viscosity oil.</td>
</tr>
<tr>
<td>2) Oil pressure gauge or sending unit inaccurate.</td>
<td>Inspect and replace as necessary.</td>
</tr>
<tr>
<td>3) Oil pressure relief valve sticking closed.</td>
<td>Remove and inspect oil pressure relief valve assembly.</td>
</tr>
</tbody>
</table>
### 3.2 Problem 1.B

Driving in to work you notice your car "backfiring." Your mechanic informs you that "we'll have to test the diverter valve and replace it if necessary." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem.

**ANTICIPATED SOLUTION (In Order of Likely Occurrence)**

<table>
<thead>
<tr>
<th>POSSIBLE CAUSE</th>
<th>CORRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Air leak into manifold vacuum.</td>
<td>Check manifold vacuum and repair as necessary.</td>
</tr>
<tr>
<td>2) Faulty air injection diverter valve.</td>
<td>Test diverter valve and replace as necessary.</td>
</tr>
<tr>
<td>3) Exhaust leak.</td>
<td>Locate and eliminate leak.</td>
</tr>
</tbody>
</table>
Driving in to work you notice your Chevrolet Camaro appears to vibrate when driving at high speeds. Your mechanic informs you that "we'll have to balance your tires." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem. If the mechanic is correct in his diagnosis, describe the procedure that can be used to check tire balancing.

ANTICIPATED SOLUTION (In Order of Likely Occurrence)

<table>
<thead>
<tr>
<th>POSSIBLE CAUSE</th>
<th>CORRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Tire(s) are not balanced.</td>
<td>Rebalance tire(s).</td>
</tr>
<tr>
<td>2) Tire(s) are warped or defective.</td>
<td>Replace tire(s).</td>
</tr>
<tr>
<td>3) Wheel(s) are loose.</td>
<td>Tighten lug nut(s).</td>
</tr>
<tr>
<td>4) Bent axle or hub.</td>
<td>Inspect and replace if necessary.</td>
</tr>
</tbody>
</table>

ANTICIPATED SOLUTION re: TIRE BALANCING

Tire balance can be checked in the following way:

1) Rotate tires so that tire to be balanced becomes a drive wheel.
2) Jack up wheel to be balanced only. Make sure car is very steady.
3) Enter car, start engine, and place in DRIVE or 4th gear.
4) Drive car at approximately 70 kmph.
5) If car shakes tire is out of balance.
6) Repeat for rest of tires.
3.4 Problem 2.B

Prior to driving in to work, you notice that the front tires on your Chevrolet Camaro appear to be wearing irregularly. Your mechanic informs you that “we’ll have to adjust the Toe.” Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem. If the mechanic is correct in his diagnosis, describe the procedure that can be used to check Toe Adjustment.

**ANTICIPATED SOLUTION (In Order of Likely Occurrence)**

<table>
<thead>
<tr>
<th>POSSIBLE CAUSE</th>
<th>CORRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Toe not to specification</td>
<td>Make Toe adjustment.</td>
</tr>
<tr>
<td>2) Camber not to specification</td>
<td>Make Camber adjustment.</td>
</tr>
<tr>
<td>3) Tire(s) out of balance</td>
<td>Rebalance tire(s).</td>
</tr>
<tr>
<td>4) Improper tire inflation</td>
<td>Check tire inflation.</td>
</tr>
</tbody>
</table>

**ANTICIPATED SOLUTION re: TOE ADJUSTMENT**

**TOE ADJUSTMENT**

Toe-out is controlled by tie rod position. Adjustment is made by loosening the clamp bolts at the steering knuckle end of the tie rods and rotating them to obtain proper toe setting. After correct toe setting is obtained, torque clamp bolts to 14 ft.lbs.

**NOTE:** Toe setting is the only adjustment normally required. However, in special circumstances, such as damage due to road hazard or collision, camber may need to be adjusted by modifying the strut assembly.
Driving into work you notice that your oil pressure gauge has risen to read "high". Your mechanic informs you that "we'll have to remove and inspect the oil pressure relief valve assembly." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem.

Complete the following questions.

1) Is the mechanic's diagnosis reasonable (i.e., possible)? Check either:
   - Yes,   - No,   - Unsure

2) Should the mechanic consider alternative causes for this problem FIRST? Check either:
   - Yes,   - No,   - Unsure

3) If Yes to question 2 above, list the other possible causes and corrections for this problem.

<table>
<thead>
<tr>
<th>CAUSES</th>
<th>CORRECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Driving in to work you notice your car "backfiring". Your mechanic informs you that "we'll have to test the diverter valve and replace it if necessary." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem.

Complete the following questions.

1) Is the mechanic’s diagnosis reasonable (i.e., possible)? Check either:
   __ Yes, __ No, __ Unsure

2) Should the mechanic consider alternative causes for this problem FIRST? Check either:
   __ Yes, __ No, __ Unsure

3) If Yes to question 2 above, list the other possible causes and corrections for this problem.

   **CAUSES**     **CORRECTIONS**
   __________________________  __________________________
   __________________________  __________________________
   __________________________  __________________________
   __________________________  __________________________
Driving in to work you notice your Chevrolet Camaro appears to vibrate when driving at high speeds. Your mechanic informs you that “we’ll have to balance your tires.” Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem. If the mechanic is correct in his diagnosis, describe the procedure that can be used to check tire balancing.

Check the following questions.

1) Is the mechanic's diagnosis reasonable (i.e., possible)? Check either:
   - Yes, _____ No, _____ Unsure

2) Should the mechanic consider alternative causes for this problem FIRST? Check either:
   - Yes, _____ No, _____ Unsure

3) If Yes to question 2 above, list the other possible causes and corrections for this problem.

<table>
<thead>
<tr>
<th>CAUSES</th>
<th>CORRECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4) If the mechanic is correct in his diagnosis, i.e., the tires need to be balanced, describe the procedure that can be used to check tire balancing.
3.8 Quiz Problem 2.B

Problem

Prior to driving in to work, you notice that the front tires on your Chevrolet Camaro appear to be wearing irregularly. Your mechanic informs you that "we'll have to adjust the Toe." Is this a reasonable diagnosis? Should the mechanic consider alternative causes for this problem? If so, determine the other possible causes and corrections for this problem.

If the mechanic is correct in his diagnosis, describe the procedure that can be used to check Toe Adjustment.

Check the following questions.

1) Is the mechanic's diagnosis reasonable (i.e., possible)? Check either:
   ___ Yes, ___ No, ___ Unsure

2) Should the mechanic consider alternative causes for this problem FIRST? Check either:
   ___ Yes, ___ No, ___ Unsure

3) If Yes to question 2 above, list the other possible causes and corrections for this problem.

<table>
<thead>
<tr>
<th>CAUSES</th>
<th>CORRECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4) If the mechanic is correct in his diagnosis, i.e., the Toes needs to be adjusted, describe the procedure that can be used to check Toe Adjustment.
3.9 Screen Captures - CAAD CUI & CAAD GUI

Screen captures for CAAD CUI are presented in the appendix that follows. These are actual screen captures for problems 1.A and 2.A. Because the captures for problems 1.B and 2.B are similar in nature, they are not reproduced here.

Screen captures for CAAD GUI are also presented in the appendix that follows. These are actual local window screen captures for problems 1.A and 2.A. Because the captures for problems 1.B and 2.B are similar in nature, they are not reproduced here.
4 GOALS, OPERATORS, METHODS, SELECTION (GOMS) Analysis

A GOMS model provides a complete dynamic description of user behaviour, measured at the level of goals, methods, and operators. As such, a GOMS model consists of four components:

1. a set of GOALS,
2. a set of OPERATORS,
3. a set of METHODS for achieving goals, and
4. a set of SELECTION rules for choosing among competing methods for goals.

All of the above, under this analysis, are represented through a hierarchical series of statements (see Sections 4.1 and 4.2 following). The GOMS analysis, is in some ways, very similar to a system data flow diagram of user behaviour that employs words rather than data symbols as descriptors.

In terms of a general GOMS analysis, GOALS are organized into a hierarchy. Eventually, each goal terminates on a set of operators. Associated with each goal is a set of alternative methods by which the goal can be achieved, as well as a set of selection rules for selecting among the methods.

Some goals are analyzed by parameters, or memory chunks that must be maintained in the user's "working memory" at the time the goal is executed. Methods can decompose a specific problem in UNIT-TASKS. Method selection is handled by a set of selection rules. Each selection rule is of the form "if such-and-such is true in the current task situation, then use method M", etc.
As the detail of the GOMS model increases, it is more difficult to predict the precise operator sequence that a user would employ on any specific occasion. In the CAAD system, the user, in effect, attempts to accomplish a series of goals in order to complete the larger end-goal (that being the solution to an automotive failure). In the pages that follow, every attempt has been made to decompose user activity into precise operation sequences.

Results of the GOMS analysis are detailed in Section 4.3.
4.1 GOMS Analysis - Character User Interface

GOMS ANALYSIS - CHARACTER-BASED USER INTERFACE

<NGOMSL FORMAT>

STEP 1. ACHIEVE GOAL OF DIAGNOSING PROBLEM 1A

Method to accomplish goal of diagnosing PROBLEM 1A

Step 1. GET TASK INFORMATION FROM DOCUMENT.

Selection Rule Set for goal of diagnosing PROBLEM 1A

- If the problem is car-specific, then accomplish goal of diagnosis using the Detailed Diagnostics.
- If the problem is not car-specific, then accomplish goal of diagnosis using the Quick Diagnostics.

Report Goal Accomplished.

Step 2. ACHIEVE GOAL OF SELECTING QUICK DIAGNOSTICS.

Method of accomplishing goal of selecting QUICK DIAGNOSTICS.

Step 1. Determine option number of the option QUICK DIAGNOSTICS.

Step 2. Locate number on keyboard.

Step 3. Enter number of option QUICK DIAGNOSTICS.

Method to enter number of option QUICK DIAGNOSTICS.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.
Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for text “QUICK DIAGNOSTICS”.

Step 2. If “QUICK DIAGNOSTICS” is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 3. ACCOMPLISH GOAL OF SELECTING PROBLEM AREA IN THE CAR.

Method of accomplishing goal of selecting PROBLEM AREA in the car.

Step 1. Retain <PROBLEM AREA> and accomplish goal of selecting PROBLEM AREA in the car.

Step 2. If <PROBLEM AREA> is on-screen, then determine letter of option <PROBLEM AREA>.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the option <PROBLEM AREA>.

Method to enter letter of option <PROBLEM AREA>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.
Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <PROBLEM AREA>.
Step 2. If <PROBLEM AREA> is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that <X>EXIT CAAD, P=PREVIOUS SCREEN, F1HELP> are located at the bottom of the screen.


Step 4. ACCOMPLISH GOAL OF SELECTING SPECIFIC DETAILS OF PROBLEM.

Method of accomplishing goal of selecting SPECIFIC DETAILS OF PROBLEM.

Step 1. Retain <SPECIFIC DETAIL> and accomplish goal of selecting SPECIFIC DETAILS OF THE PROBLEM.
Step 2. If <SPECIFIC DETAIL> is on-screen, then determine letter of option <SPECIFIC DETAIL>.
Step 3. Locate letter on keyboard.
Step 4. Enter the letter of option <SPECIFIC DETAIL>.

Method to enter letter of option <SPECIFIC DETAIL>.

Step 1. Place finger over an appropriate key.
Step 2. Depress key once.
Step 3. Report goal accomplished.
Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <MECHANIC'S DIAGNOSIS>.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 5. ACCOMPLISH GOAL OF SELECTING MECHANIC'S DIAGNOSIS.

Method of accomplishing goal of selecting <MECHANIC'S DIAGNOSIS>.

Step 1. Retain the diagnosis was <MECHANIC'S DIAGNOSIS> and accomplish goal of selecting MECHANIC'S DIAGNOSIS.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, determine the letter of option <MECHANIC'S DIAGNOSIS>.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the option <MECHANIC'S DIAGNOSIS>.

Method to enter letter of option <MECHANIC'S DIAGNOSIS>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
Method to verify that correct option has been selected.

Step 1. Scan next screen for <MECHANIC'S DIAGNOSIS>.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 6. ACCOMPLISH GOAL OF ANALYSING OTHER SOLUTIONS.

Method to accomplish goal of ANALYSING OTHER SOLUTIONS.

Step 1. Return to previous screen.

Method of returning to previous screen.

Step 1. Retain that <X=EXIT, P=PREVIOUS SCREEN, F1=HELP> are located at bottom of the screen.

Step 2. Determine letter of option PREVIOUS SCREEN.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of option PREVIOUS SCREEN.

Method to enter letter of option PREVIOUS SCREEN.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.
Step 1. Scan screen for <SOLUTIONS>.

Step 2. If <SOLUTIONS> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 2. Decide: If no more <OTHER SOLUTIONS> exist, then report goal accomplished.

Step 3. Determine the letter of the option <SOLUTION #N>.

Step 4. Locate letter on keyboard.

Step 5. Enter the letter of the option <SOLUTION #N>.
   Method to enter letter of option <SOLUTION #N>.
   Step 1. Place finger over appropriate key.
   Step 2. Depress key once.
   Step 3. Report goal accomplished.

Step 6. Verify that correct option has been selected.
   Method to verify that correct option has been selected.
   Step 1. Scan next screen for <SOLUTION #N>.
   Step 2. If <SOLUTION #N> is on-screen, then report goal accomplished, else return to previous screen.
   Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN,
Step 7. Go to 1.

Step 7. ACCOMPLISH GOAL OF RETURNING TO THE MAIN MENU.

Method to accomplish goal of RETURNING TO THE MAIN MENU.

Step 1. Retain that 'X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP' are located at the bottom of the screen.

Step 2. Determine the letter of the option EXIT.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the option EXIT.

Method to enter letter of option EXIT.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for MAIN MENU.

Step 2. If MAIN MENU is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that 'X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP' are located at the bottom of the screen.


STEP 2. ACCOMPLISH GOAL OF DIAGNOSING PROBLEM 2A

Method to accomplish goal of diagnosing PROBLEM 2A.

Step 1. GET TASK INFORMATION FROM DOCUMENT.

Selection Rule Set for goal of diagnosing PROBLEM 2A.

If the problem is car-specific, then accomplish goal of diagnosis using the Detailed Diagnostics.

If the problem is not car-specific, then accomplish goal of diagnosis using the Quick Diagnostics.

Report Goal Accomplished.

Step 2. ACCOMPLISH GOAL OF SELECTING DETAILED DIAGNOSTICS.

Method to accomplish goal of selecting DETAILED DIAGNOSTICS.

Step 1. Determine option number of the option DETAILED DIAGNOSTICS.

Step 2. Locate number on keyboard.

Step 3. Enter the number of the option DETAILED DIAGNOSTICS.

Method to enter number of option DETAILED DIAGNOSTICS.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for text "DETAILED DIAGNOSTICS".
Step 2. If "DETAILED DIAGNOSTICS" is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 3. ACCOMPLISH GOAL OF SELECTING AUTOMOTIVE MANUFACTURER.

Method to accomplish goal of selecting AUTOMOTIVE MANUFACTURER.

Step 1. Retain that car is <MANUFACTURER, MAKE, MODEL> and accomplish goal of selecting AUTOMOTIVE MANUFACTURER.

Step 2. If <MANUFACTURER> is on-screen, then determine the letter of the option <MANUFACTURER>.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the option <MANUFACTURER>.

Method to enter letter of option <MANUFACTURER>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.
Step 1. Scan next screen for MANUFACTURER.
Step 2. If MANUFACTURER is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP are located at the bottom of the screen.

Step 4. ACCOMPLISH GOAL OF SELECTING MAKE OF AUTOMOBILE.
Method to accomplish goal of selecting MAKE OF AUTOMOBILE.
Step 1. Retain that car is MANUFACTURER, MAKE, MODEL and accomplish goal of selecting MAKE OF AUTOMOBILE.
Step 2. If MAKE is on-screen, then determine option number of MAKE.
Step 3. Locate number on keyboard.
Step 4. Enter the number of the option MAKE.
Method to enter number of option MAKE.
Step 1. Place finger over appropriate key.
Step 2. Depress key once.
Step 3. Report goal accomplished.
Step 5. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for MAKE.
Step 2. If MAKE is on-screen, then report goal accomplished,
else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 5. ACCOMPLISH GOAL OF SELECTING AUTOMOBILE MODEL.

Method to accomplish goal of selecting AUTOMOBILE MODEL.

Step 1. Retain that car is <MANUFACTURER, MAKE, MODEL> and accomplish goal of selecting AUTOMOBILE MODEL.

Step 2. If <MODEL> is on-screen, then determine letter of the option <MODEL>.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the option <MODEL>.

Method to enter letter of option <MODEL>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <MODEL>.

Step 2. If <MODEL> is on-screen, then report goal
accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 6. ACCOMPLISH GOAL OF FINDING PROBLEM AREA IN THE CAR.

Method to accomplish goal of finding PROBLEM AREA in the car.

Step 1. Retain <PROBLEM AREA> and accomplish goal of finding PROBLEM AREA in the car.

Step 2. If <PROBLEM AREA> is on-screen, then determine option number of <PROBLEM AREA>.

Step 3. Locate number on keyboard.

Step 4. Enter the number of the option listing <PROBLEM AREA>.

Method to enter number of option <PROBLEM AREA>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.
Step 1. Scan next screen for <PROBLEM AREA>.

Step 2. If <PROBLEM AREA> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X>EXIT CAAD, <P>PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 7. ACHIEVE GOAL OF SELECTING SPECIFIC DETAILS OF PROBLEM.

Method to accomplish goal of selecting SPECIFIC DETAILS OF THE PROBLEM.

Step 1. Retain <SPECIFIC DETAIL> and accomplish goal of selecting SPECIFIC DETAILS OF THE PROBLEM.

Step 2. If <SPECIFIC DETAIL> is on-screen, then determine the letter of the option <SPECIFIC DETAIL>.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of the <SPECIFIC DETAIL>.

Method to enter letter of option <SPECIFIC DETAIL>.

Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
Method to verify that correct option has been selected.

Step 1. Scan next screen for <SPECIFIC DETAIL>.

Step 2. If <SPECIFIC DETAIL> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 8. ACCOMPLISH GOAL OF SELECTING MECHANIC'S DIAGNOSIS.

Method to accomplish goal of selecting MECHANIC'S DIAGNOSIS.

Step 1. Retain the diagnosis was <MECHANIC'S DIAGNOSIS> and accomplish goal of selecting MECHANIC'S DIAGNOSIS.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then determine the option number of <MECHANIC'S DIAGNOSIS>.

Step 3. Locate the number on the keyboard.

Step 4. Enter the number of the option <MECHANIC'S DIAGNOSIS>.

Method to enter number of option <MECHANIC'S
Step 1. Place finger over appropriate key.

Step 2. Depress key once.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
   Method to verify that correct option has been selected.
   Step 1. Scan next screen for `<MECHANIC'S DIAGNOSIS>`.
   Step 2. If `<MECHANIC'S DIAGNOSIS>` is on-screen, the report goal accomplished, else return to previous screen.
   Step 3. Retain that `X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP` are located at the bottom of the screen.


Step 9. ACCOMPLISH GOAL OF ANALYSING OTHER SOLUTIONS.
   Method to accomplish goal of ANALYSING OTHER SOLUTIONS.
   Step 1. Return to previous screen.
   Method of returning to previous screen.
   Step 1. Retain that `X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP` are located at bottom of
Step 2. Determine letter of option PREVIOUS SCREEN.

Step 3. Locate letter on keyboard.

Step 4. Enter the letter of option PREVIOUS SCREEN.
   Method to enter letter of option PREVIOUS SCREEN.
   Step 1. Place finger over appropriate key.
   Step 2. Depress key once.
   Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
   Method to verify that correct option has been selected.
   Step 1. Scan screen for <SOLUTIONS>.
   Step 2. If <SOLUTIONS> is on-screen, then report goal accomplished, else return to previous screen.
   Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.


Step 2. Decide: If no more <OTHER SOLUTIONS> exist, then report goal accomplished.
Step 3. Determine option number of <SOLUTION #N>.

Step 4. Locate number on keyboard.

Step 5. Enter the number of the option <SOLUTION #N>.
Method to enter number of option <SOLUTION #N>.
Step 1. Place finger over appropriate key.
Step 2. Depress key once.
Step 3. Report goal accomplished.

Step 6. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for <SOLUTION #N>.
Step 2. If <SOLUTION #N> is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F1=HELP> are located at the bottom of the screen.

Step 7. Go to 1.

Step 10. REPORT GOAL ACCOMPLISHED.

STEP 3. REPORT GOAL ACCOMPLISHED.
4.2 GOMS Analysis - Graphical User Interface

GOMS ANALYSIS - GRAPHICAL USER INTERFACE

<NGOMSL FORMAT>

STEP 1. ACHIEVE GOAL OF DIAGNOSES PROBLEM 1A

Method to accomplish goal of diagnosing PROBLEM 1A.

Step 1. GET TASK INFORMATION FROM DOCUMENT.

Selection Rule Set for goal of diagnosing PROBLEM 1A

If the problem is car-specific, then accomplish goal of diagnosis using the
Detailed Diagnostics.

If the problem is not car-specific, then accomplish goal of diagnosis using
the Quick Diagnostics.

Report Goal Accomplished.

Step 2. ACHIEVE GOAL OF SELECTING QUICK DIAGNOSTICS.

Method of accomplishing goal of selecting QUICK DIAGNOSTICS.

Step 1. Determine option number of the option QUICK DIAGNOSTICS.

Step 2. Move mouse pointer to the position of the menu option QUICK
DIAGNOSTICS.

Method to enter number of option QUICK DIAGNOSTICS

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches

destination.

Step 3. Report goal accomplished.
Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for text “QUICK DIAGNOSTICS”.

Step 2. If “QUICK DIAGNOSTICS” is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 3. ACCOMPLISH GOAL OF SELECTING PROBLEM AREA IN THE CAR.

Method of accomplishing goal of selecting PROBLEM AREA in the car.

Step 1. Retain <PROBLEM AREA> and accomplish goal of selecting PROBLEM AREA in the car.

Step 2. If <PROBLEM AREA> is on-screen, then place mouse pointer over menu option listing <PROBLEM AREA>.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <PROBLEM AREA>.

Step 2. If <PROBLEM AREA> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.
Step 4. ACCOMPLISH GOAL OF SELECTING SPECIFIC DETAILS OF PROBLEM.
Method of accomplishing goal of selecting SPECIFIC DETAILS OF PROBLEM.
Step 1. Retain <SPECIFIC DETAIL> and accomplish goal of selecting SPECIFIC DETAILS OF PROBLEMS.
Step 2. If <SPECIFIC DETAIL> is on-screen, then place mouse pointer over menu option listing <SPECIFIC DETAIL>.
Method of moving mouse pointer to correct position.
Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer screen reaches destination.
Step 3. Report goal accomplished.
Step 3. Double-click left mouse button.
Method of double-clicking left mouse button.
Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.
Step 4. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for <SPECIFIC DETAIL>.
Step 2. If <SPECIFIC DETAIL> is on-screen, report goal accomplished.
accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 5. ACCOMPLISH GOAL OF SELECTING MECHANIC'S DIAGNOSIS.

Method of accomplishing goal of selecting MECHANIC'S DIAGNOSIS.

Step 1. Retain the diagnosis was <MECHANIC'S DIAGNOSIS> and accomplished goal of selecting MECHANIC'S DIAGNOSIS.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then place mouse pointer over menu option listing <MECHANIC'S DIAGNOSIS>.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.
Step 1. Scan next screen for <MECHANIC'S DIAGNOSIS>.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <X=EXIT CAAD, P=PREVIOUS SCREEN, F=HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 6. ACCOMPLISH GOAL OF ANALYSING OTHER SOLUTIONS.

Method of accomplishing goal of ANALYSING OTHER SOLUTIONS.

Step 1. Return to previous screen.

Method of returning to previous screen.

Step 1. Retain that <EXIT, PREVIOUS SCREEN, HELP> are located at bottom of screen.

Step 2. Determine position of PREVIOUS SCREEN option.

Step 3. Place mouse pointer over PREVIOUS SCREEN option.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 4. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan screen for <SOLUTIONS>.
Step 2. If <SOLUTIONS> is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.


Step 2. Decide: If no more <OTHER SOLUTIONS> exist, the report goal accomplished.

Step 3. Place mouse pointer over <SOLUTION #N>.
Method of moving mouse pointer to correct position.
Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.
Step 3. Report goal accomplished.

Step 4. Double-click left mouse button.
Method of double-clicking left mouse button
Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.
Step 5. Verify that correct option has been selected.
   Method to verify that correct option has been selected.
   Step 1. Scan next screen for <SOLUTION #N>.
   Step 2. If <SOLUTION #N> is on-screen, the report goal accomplished,
   else return to previous screen.
   Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are
   located at the bottom of the screen.

Step 6. Go to 1.

Step 7. ACCOMPLISH GOAL OF RETURNING TO THE MAIN MENU.
   Method to accomplish goal of RETURNING TO THE MAIN MENU.
   Step 1. Determine position of EXIT option on-screen.
   Step 2. Move mouse pointer to position of EXIT option on-screen.
   Method of moving mouse pointer to correct position.
   Step 1. Place hand on mouse.
   Step 2. Slide mouse across pad until pointer on screen reaches destination.
   Step 3. Report goal accomplished.
   Step 3. Double-click left mouse button.
   Method of double-clicking left mouse button.
   Step 1. Place index finger on left mouse button.
   Step 2. Depress and release left mouse button twice.
   Step 3. Report goal accomplished.
Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for MAIN MENU.

Step 2. If MAIN MENU is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that "EXIT CAAD, PREVIOUS SCREEN, HE LP" are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 8. REPORT GOAL ACCOMPLISHED.
STEP 2. ACCOMPLISH GOAL OF DIAGNOSING PROBLEM 2A
Method to accomplish goal of diagnosing PROBLEM 2A.

Step 1. GET TASK INFORMATION FROM DOCUMENT.
Selection Rule Set for goal of diagnosing PROBLEM 2A
If the problem is car-specific, then accomplish goal of diagnosis using
the Detailed Diagnostics.
If the problem is not car-specific, then accomplish goal of diagnosis
using the Quick Diagnostics.
Report Goal Accomplished.

Step 2. ACCOMPLISH GOAL OF SELECTING DETAILED DIAGNOSTICS.
Method to accomplish goal of selecting DETAILED DIAGNOSTICS.

Step 1. Determine option number of the option DETAILED
DIAGNOSTICS.

Step 2. Move mouse pointer to the position of the menu option
DETAILED DIAGNOSTICS.
Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches
destination.
Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.
Method of double-clicking left mouse button.
Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

   Method to verify that correct option has been selected.
   
   Step 1. Scan next screen for text "DETAILED DIAGNOSTICS".
   
   Step 2. If "DETAILED DIAGNOSTICS" is on-screen, then report goal accomplished, else return to previous screen.
   
   Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 3. ACCOMPLISH GOAL OF SELECTING AUTOMOTIVE MANUFACTURER.

   Method to accomplish goal of selecting AUTOMOTIVE MANUFACTURER.

   Step 1. Retain that car is <MANUFACTURER, MAKE, MODEL> and accomplish goal of selecting AUTOMOTIVE MANUFACTURER.

   Step 2. If <MANUFACTURER>’s logo is on-screen, then place mouse pointer over the icon.

   Method of moving mouse pointer to correct position.

   Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <MANUFACTURER>.

Step 2. If <MANUFACTURER> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 4. ACCOMPLISH GOAL OF SELECTING MAKE OF AUTOMOBILE

Method to accomplish goal of selecting MAKE OF AUTOMOBILE.

Step 1. Retain that car is <MANUFACTURER, MAKE, MODEL> and accomplish goal of selecting MAKE OF AUTOMOBILE.

Step 2. If <MAKE> is on-screen, then place mouse pointer over the menu option of <MAKE>.
Method of moving mouse pointer to correct position.
Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.
Step 3. Report goal accomplished.
Step 3. Double-click left mouse button.
Method of double-clicking left mouse button.
Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.
Step 4. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for <MAKE>.
Step 2. If <MAKE> is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.
Step 5. Report goal accomplished.
Step 5. ACCOMPLISH GOAL OF SELECTING AUTOMOBILE MODEL.
Method to accomplish goal of selecting AUTOMOBILE MODEL.
Step 1. Retain that car is <MANUFACTURER, MAKE, MODEL> and accomplish goal of selecting AUTOMOBILE MODEL.
Step 2. If <MODEL> is on-screen, then place mouse pointer over the menu option of <MODEL>.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <MODEL>.

Step 2. If <MODEL> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 6. ACCOMPLISH GOAL OF FINDING PROBLEM AREA IN THE CAR.

Method to accomplish goal of finding PROBLEM AREA in the car.
Step 1. Retain <PROBLEM AREA> and accomplish goal of finding PROBLEM AREA in the car.

Step 2. If PROBLEM AREA is on-screen, then place mouse pointer over the section of the car where PROBLEM AREA is located. Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.
Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.
Method of double-clicking left mouse button.
Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for PROBLEM AREA.
Step 2. If PROBLEM AREA is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that EXIT CAAD, PREVIOUS SCREEN, HELP are located at the bottom of the screen.

Step 5. Report goal accomplished.
Step 7. ACCOMPLISH GOAL OF SELECTING SPECIFIC DETAILS OF PROBLEM.

Method to accomplish goal of selecting SPECIFIC DETAILS OF THE PROBLEM.

Step 1. Retain <SPECIFIC DETAIL> and accomplish goal of selecting SPECIFIC DETAILS OF THE PROBLEM.

Step 2. If <SPECIFIC DETAIL> is on-screen, then place mouse pointer over menu option listing <SPECIFIC DETAIL>.
   Method of moving mouse pointer to correct position.
   Step 1. Place hand on mouse.
   Step 2. Slide mouse across pad until pointer on screen reaches destination.
   Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.
   Method of double-clicking left mouse button.
   Step 1. Place index finger on left mouse button.
   Step 2. Depress and release left mouse button twice.
   Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.
   Method to verify that correct option has been selected.
   Step 1. Scan next screen for <SPECIFIC DETAIL>.
   Step 2. If <SPECIFIC DETAIL> is on-screen, then report goal
accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 8. ACCOMPLISH GOAL OF SELECTING MECHANIC'S DIAGNOSIS.

Method to accomplish goal of selecting MECHANIC'S DIAGNOSIS.

Step 1. Retain the diagnosis was <MECHANIC'S DIAGNOSIS> and accomplish goal of selecting MECHANIC'S DIAGNOSIS.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then place mouse pointer over menu option listing <MECHANIC'S DIAGNOSIS>.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 3. Double-click left mouse button.

Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.

Step 2. Depress and release left mouse button twice.

Step 3. Report goal accomplished.

Step 4. Verify that correct option has been selected.
Method to verify that correct option has been selected.

Step 1. Scan next screen for <MECHANIC'S DIAGNOSIS>.

Step 2. If <MECHANIC'S DIAGNOSIS> is on-screen, then report goal accomplished, else return to previous screen.

Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 5. Report goal accomplished.

Step 9. ACHIEVE GOAL OF ANALYSING OTHER SOLUTIONS.

Method to accomplish goal of ANALYSING OTHER SOLUTIONS.

Step 1. Return to previous screen.

Method of returning to previous screen.

Step 1. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at bottom of screen.

Step 2. Determine position of PREVIOUS SCREEN option.

Step 3. Place mouse pointer over PREVIOUS SCREEN option.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.

Step 2. Slide mouse across pad until pointer on screen reaches destination.

Step 3. Report goal accomplished.

Step 4. Double-click left mouse button.
Method of double-clicking left mouse button.

Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.
Step 5. Verify that correct option has been selected.

Method to verify that correct option has been selected.

Step 1. Scan next screen for <SOLUTIONS>.
Step 2. If <SOLUTIONS> is on-screen, then report goal accomplished, else return to previous screen.
Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.


Step 2: Decide: If no more <OTHER SOLUTIONS> exist, then report goal accomplished.

Step 3. Place mouse pointer over <SOLUTION IN>.

Method of moving mouse pointer to correct position.

Step 1. Place hand on mouse.
Step 2. Slide mouse across pad until pointer on screen reaches destination.
Step 3. Report goal accomplished.

Step 4. Double-click left mouse button.

Method of double-clicking left mouse button.
Step 1. Place index finger on left mouse button.
Step 2. Depress and release left mouse button twice.
Step 3. Report goal accomplished.

Step 5. Verify that correct option has been selected.
Method to verify that correct option has been selected.
Step 1. Scan next screen for <SOLUTION #N>.
Step 2. If <SOLUTION #N> is on-screen, the report goal accomplished, else return to previous screen.
Step 3. Retain that <EXIT CAAD, PREVIOUS SCREEN, HELP> are located at the bottom of the screen.

Step 6. Go to 1.

Step 10. REPORT GOAL ACCOMPLISHED.

STEP 3. REPORT GOAL ACCOMPLISHED.
4.3 Results of GOMS Analysis

DIAGNOSING PROBLEMS 1A AND 2A

STEP 1:

The step of obtaining task information from the document (see Section 3.5 & Section 3.7) was the same under both the CUI and the GUI problem sessions. There were no differences in this step that would lead to a variation in user performance.

STEP 2 TO COMPLETION:

The main goals of each of these steps was the same under both UIs. However, the methods needed to accomplish the goals differed.

It took more steps to accomplish the goals with the GUI than with the CUI. The main reason for this difference was the task of pointing and clicking with the mouse. This task needed to be broken down into many subgoals for the user. For this reason, the accomplishment of each of these subgoals added additional time to the user’s overall session.

The pointing and clicking of the mouse required additional physical movement by the user, and depending on the user’s experience, the use of the GUI may have proved cumbersome and difficult.

The use of the CUI involved first visually scanning the keyboard for the proper key and then depressing it; with the GUI, the user needed to watch the mouse pointer pan across the screen while moving the hand to place the pointer over the proper area of the screen.

The GUI user had to concentrate on two movements at the same time, while the CUI user did not need to do so. Depending on the abilities of the user, he/she may not have been able to maintain these two movements independently and therefore separated these steps into further subgoals. If this
was the case, then the overall session time would have increased because separation of the steps required more time than if the user had worked on the two processes in parallel.

The process of depressing the necessary key(s) was different under the CUI and GUI sessions. Under the CUI, the user simply had to depress the desired key once in order to activate the next screen, while the GUI user needed to depress the left mouse button twice in ("rapid") sequence. Although the extra depression of the key requires a negligible amount of time for one step, this time builds over the course of the session and would contribute to an explanation of why on average the GUI session time took longer to complete. As well, users do not always depress the mouse button twice in a rapid sequence, and therefore the screens do not advance, often requiring the users to repeat the process. This will again cause delays in session time and would further explain the relatively high number of regressive commands performed by GUI users over CUI users.

Depressing the necessary keys results in different hand positions for the users under the different UIs. Under the CUI, the user simply needs to place one finger over the key and depress it. Under the GUI, however, the user may often have all fingers placed around the mouse while the index finger acts to depress the appropriate button. This hand positioning makes the task decidedly more complex. This difference may affect user performance for those who lack GUI experience and/or manual dexterity.

It should be noted that once a user has selected his/her option under the GUI, the hand operating the mouse may inadvertently slide to a different physical position. Therefore, depending on the location of the next option to be selected, the user may require additional time to place the pointer over the necessary item. This may not happen for all users. However, it may be a natural tendency subject to the location of the option selected, i.e., if the option is in the top portion of the screen, then
the user's hand is often high up on the mouse pad which is not a normally comfortable resting position. Therefore the user may slide the mouse downwards after accomplishing the goal of selecting the correct option. Should the next option to be selected also be at the top portion of the screen, the user must again slide the mouse upwards along the mouse pad (which requires more time than if the hand was resting in the same position on the mouse pad as in the previous screen).

Such subgoals as scanning the screen for proper textual information and for verifications of decisions were the same under both UIs, hence the user was not advantaged or disadvantaged using one UI over the other in this respect. Information that would provide the user with verification of the selections was located in the same position on the screen with respect to either UI, and therefore once again, this would not have been a contributing factor with respect to user performance.

The locations of the required information were the same under both the CUI and the GUI, i.e., the UIs were consistent, although the methods for presentation of the material varied (e.g., text-based options vs. icons). Therefore, the screen layouts of options were not a factor in the user's overall session time under both UIs.

Overall, the tasks under both UIs were the same, with differences occurring only in methodologies for goal-completion. As the GOMS analysis shows, the tasks are decidedly more complex under the GUI, due to the many movements required to accomplish the goals and subgoals.

The GOMS analysis does not, however, present any insights as to why the user would prefer a particular UI. The greater number of subgoals under the GUI would lead one to conclude that users would not prefer this interface due to the increased complexity. However, this was not so, as the GUI was preferred by most users.
5 Results

Two hundred and fifty people participated in this study. Each participant was asked to solve two sets of two automobile failure problems using the two systems described in this thesis. People from varied fields, ranging in age from 16 to 66, of almost equal gender representation, were participants in this study (see Table 5.0-1).

<table>
<thead>
<tr>
<th>Gender</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Females</td>
<td>52.4</td>
</tr>
<tr>
<td>Males</td>
<td>47.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4.0</td>
</tr>
<tr>
<td>17 - 26</td>
<td>18.8</td>
</tr>
<tr>
<td>27 - 36</td>
<td>20.0</td>
</tr>
<tr>
<td>37 - 46</td>
<td>23.2</td>
</tr>
<tr>
<td>47-56</td>
<td>17.6</td>
</tr>
<tr>
<td>57-66</td>
<td>16.4</td>
</tr>
</tbody>
</table>

This study was conducted in a local garage in Mississauga, Ontario (Canada), geographically situated near several large automobile dealerships (both domestic [GM, Chrysler] and import [Honda, Toyota]). For the most part, though not crucial to the findings presented in this thesis, Ford products represented the bulk of the vehicles being serviced in this garage.
The participants in this study came from the local community. The responses of people with different experiences, backgrounds, and computer skills (see Figure 5.0-2) were studied. Some of these individuals had English as their second language, with limited reading skills — though in the pre-interview stage, if language skills would pose a problem, the individual's participation was not sought.

For the most part, the group of people studied is typical of the individual who would likely be using a commercial release of CAAD CUI / CAAD GUI. Moreover, during the outset of the conversation with a prospective participant, the author could readily identify those individuals who were interested in spending some of their time investigating these systems. This latter point is important in the sense that committed individuals did not give way to the many distractions that are often present as background noise, customer conversation, etc., in a busy garage.

Figure 5.0-2 People's Level of Computer Experience Participating in the CAAD Study (n=250)
Additionally, participation on average (with some interviewing, explanation and questions by the individual) would consume more than an hour of time (see Section 5), consequently, it was important to get user commitment to ensure accurate findings.

About 1/3 of the people in this study had a significant amount of computing experience (see Figure 5.0-2). As expected, the group that had the most computer experience tended to have the most knowledge of GUIs (see Figure 5.0-4).

Figure 5.0-3 Distribution of People's GUI Familiarity (n=250)
The other 2/3 of the participants had little to no computer experience to some computer experience with “spotted” knowledge of GUIs (see Table 5.0-4). At the time this study was conducted, most of an individual’s knowledge of GUIs would have stemmed from exposure to either Macintosh OS, Amiga GEOS, or perhaps Microsoft Windows 3.0 (or an earlier pre-evaluation Windows OS).

During the pre-interview conversation and interview session, individuals were classified (see Section 2). This study was conducted during one summer (three months). Based on the consistency of the data analysis presented in this thesis, it is highly unlikely that the findings would vary significantly if additional users were studied, save the fact that with each month passing, more users would have gained some exposure to GUIs (either through the rapid proliferation of the Windows OS, or Macintosh OS).

<table>
<thead>
<tr>
<th>Computer Experience</th>
<th>None</th>
<th>Some/Little</th>
<th>Lots</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>32.8%</td>
<td>18.0%</td>
<td>9.2%</td>
</tr>
<tr>
<td>GUI Familiarity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Some/Little</td>
<td>0%</td>
<td>16.4%</td>
<td>8.8%</td>
</tr>
<tr>
<td>Lots</td>
<td>0%</td>
<td>0%</td>
<td>14.8%</td>
</tr>
</tbody>
</table>
5.1 Observations

On average, people completed the problem-solving tasks in approximately one hour, i.e., 29.08 minutes for the CUI session, and 31.20 minutes for the GUI session. By removing the quickest 5% from each session and the slowest 5%, the results do not change dramatically, i.e., 28.70 minutes for the CUI session and 30.77 minutes for the GUI session.

The measure of dispersion (standard deviation) around a mean session time of 30 minutes was approximately 12 minutes.

Figure 5.1 Composite Time Summary (Modified Box-Plot) for CAAD CUI Sessions and CAAD GUI Sessions
About 1 out of 5 experienced computer users achieved shorter session times using the CUI version of CAAD. This is not a surprising finding because of the users’ histories of CUI experience. At the other end of the spectrum, the slowest users had longer session times using the GUI version of CAAD. Using the CUI version of CAAD, 6.4% of the people had session times between 50 and 60 minutes while 10.4% had similar times using the GUI version of CAAD. Many factors may account for these times. As previously cited, GUIs may promote browsing and browsing does take time.

Moreover, individuals who had longer session times with the GUI did issue more commands during their session (see Figure 5.1-1) or made additional requests for on-line help (see Figure 5.1-2). Overall, there was less of a pronounced range of commands issued using the CUI version of CAAD (see Figure 5.1-1).

Figure 5.1-1 Frequency Distribution - Number of Commands Issued Under Each UI
On average, people used fewer commands with the CUI version of CAAD than the GUI version.

Also interesting are the number of times the UI mislead the user (see Figure 5.1-4). People can be misdirected for many reasons, e.g., failure to read a menu correctly, misinterpreting a menu item, clicking an object that they believe represents one thing when it represents something quite different. More people were misdirected more frequently using GUI version of CAAD. Again, misdirection represents any movement not immediately directed with the end of reaching a decision in the most economic amount of keystrokes. Yet learning styles may influence goal immediacy, i.e., what constitutes a misdirected rule path to be initiated from the system may, from the perspective of the user, reflect goal progression.

Figure 5.1-2 Frequency Distribution - Number of Times Help was Sought During Each CAAD Session
Figure 5.1-3 Frequency Distribution - Number of Repetition of Failed Commands Issued Under Each CAAD UI Session

Figure 5.1-4 Frequency Distribution - Number of Times the User was Mislead by the UI Under Both CAAD Sessions
Fewer individuals required any help using the CUI version of CAAD (see Figure 5.1-2) than the GUI version. However, of those individuals who required help, more help was increasingly required of the CUI version of CAAD as the sessions progressed. More help was requested more frequently of the GUI version of CAAD. There appears to be a pattern based on the material presented in Figure 5.1-2. Essentially, a user is more likely not to wander (or browse) using the CUI version of CAAD (see Figure 5.1-5). Fewer commands, less misdirection and fewer requests for help seem to categorize most of the CUI user sessions. However, if the user's work within the CUI version of CAAD lacks goal progression, then increasingly help will be sought. This is suggested by the fact that as problems are encountered with the CUI version of CAAD, increasingly the user will continue to process repeated failed commands (see Figure 5.1-3). Regardless of the UI, people tended to make repeated attempts of failed commands, even in light of the fact that no immediate goal progress is achieved.

Figure 5.1-5 Frequency Distribution - Number of Regressive Commands Issued Under Each CAAD Session

![Frequency Distribution Chart](image-url)
Figure 5.1-6 Frequency Distribution - Number of Times User Lost Control While Operating Under CAAD CUI and CAAD GUI

Proportion of People (n=260)

0.0
0.2
0.4
0.6
0.8
1.0

Number of Times User Lost Control Under Each CAAD UI Session

- CUT
- GUI
5.2 Summary of User Performance

Computer Experience vs. Regressive Commands

Most users (160 out of 250), did not enter any regressive commands while operating under the CUI version of CAAD. Of those that did enter 1 regressive command (i.e., 89 out of 250), 82% had little to no computer experience.

More users tended to make regressive commands while operating under the GUI version of CAAD (i.e., 192 out of 250). Of those who made these regressive commands, 89% had little to no GUI experience.

Computer Experience vs. Loss of Control

Of the population 4% lost control of the CUI version of CAAD, while only 1.6% of the population lost control of the GUI version of CAAD.

Of the 4% of the population that lost control of the CUI version of CAAD, all had no computer experience. Similarly, of the 1.6% of the population who lost control of the GUI version of CAAD, neither had any computer experience nor GUI experience. Those individuals who lost control of the GUI version of CAAD similarly lost control of the CUI version of CAAD.

Percentage Time in Errors

Under the CUI version of CAAD, the mean time spent in error was 4.1 minutes. Under the GUI version of CAAD, the mean time spent in error was 8.4 minutes.
Computer Experience vs. Time (CUI Session)

Individuals with more computer experience required less time to complete the problems posed to them (see Figure 5.2-1).

Figure 5.2-1 Computer Experience vs. CUI Maximum / Minimum Session Times

Similarly, individuals with GUI experience required less session time to complete the problems posed to them.

Out of the population 37 individuals had lots of computer experience and GUI experience. This group took an average 18.8 minutes of session time under the CUI version of CAAD and 21 minutes under the GUI version of CAAD.

Computer Experience vs. Number of Times Mislead by UI

In all categories of computer experience and GUI experience, more individuals were mislead operating under the GUI (see Figure 5.1-4).
Computer Experience vs. Number of Failed Commands

As computer experience and GUI experience increased, fewer failed commands were issued operating under both the CUI version of CAAD and the GUI version of CAAD. Of those with lots of experience, 68.3% had performed at least one failed command under the CUI version of CAAD, while 86% of those with lots of experience entered at least one failed command under the GUI version of CAAD. Regardless of computer experience, most users under both interfaces performed at least one failed command.

Computer Experience vs. Number of Times Accessed Help

As both computer experience and GUI experience increased, help was required less often under both versions of CAAD. Typically, those with no computer experience or with very little experience had to search the on-line help anywhere from 2 to 9 times. Moreover, if an individual had to access help at least 2 times, there was a 94% chance that he/she would continue to access on-line help for as many as 4 more times.

Percentage of Session Time Spent in Error

More percentage of total session time was spent in error operating under the GUI version of CAAD. Of all users, 70.4% spent 20% or more of their session time in errors operating under errors, as opposed to 26% for CUI sessions.
5.3 Summary of User Preferences

User Preferences vs. Computer Experience and GUI Experience

People with no computer experience and no GUI experience overwhelmingly preferred the GUI version of CAAD (see Table 5.3-1).

<table>
<thead>
<tr>
<th>Control: Computer Experience</th>
<th>No GUI Experience</th>
<th>Some GUI Experience</th>
<th>Lots of GUI Experience</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI Preferred</td>
<td>98.78%</td>
<td>0%</td>
<td>0%</td>
<td>98.78%</td>
</tr>
<tr>
<td>CUI Preferred</td>
<td>1.22%</td>
<td>0%</td>
<td>0%</td>
<td>1.22%</td>
</tr>
<tr>
<td>All</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

People with some computer experience with either no GUI or little GUI experience prefer the GUI version of CAAD about two to one over the CUI version of CAAD (see Table 5.3-2).

<table>
<thead>
<tr>
<th>Control: Computer Experience</th>
<th>No GUI Experience</th>
<th>Some GUI Experience</th>
<th>Lots of GUI Experience</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI Preferred</td>
<td>33.72%</td>
<td>31.40%</td>
<td>0%</td>
<td>65.12%</td>
</tr>
<tr>
<td>CUI Preferred</td>
<td>18.60%</td>
<td>16.28%</td>
<td>0%</td>
<td>34.88%</td>
</tr>
<tr>
<td>All</td>
<td>53.33%</td>
<td>47.87%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>
People with lots of computer experience regardless of GUI experience prefer the CUI version of CAAD to the GUI version, else have no expressed UI preference (see Table 5.3-3).

Table 5.3-3 GUI experience vs. UI preference

<table>
<thead>
<tr>
<th>Control: Computer Experience = Lots</th>
<th>No GUI Experience</th>
<th>Some GUI Experience</th>
<th>Lots of GUI Experience</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI Preferred</td>
<td>4.88%</td>
<td>6.10%</td>
<td>0%</td>
<td>10.98%</td>
</tr>
<tr>
<td>CUI Preferred</td>
<td>23.17%</td>
<td>20.73%</td>
<td>25.61%</td>
<td>69.51%</td>
</tr>
<tr>
<td>No Preference</td>
<td>0%</td>
<td>0%</td>
<td>19.51%</td>
<td>19.51%</td>
</tr>
<tr>
<td>All</td>
<td>28.03%</td>
<td>26.83%</td>
<td>45.12%</td>
<td>100%</td>
</tr>
</tbody>
</table>

About one out of five of the "power" users, or very experienced computer users, those individuals with lots of GUI experience had no interface preference. Essentially, this group of individuals was able to pursue the solutions to the problems sited in this study equally well under either interface.

Expressed UI Preference vs. Session Time

People who took a longer time to solve the problems posed to them (i.e., where CUI session time exceeded 30 minutes) preferred the GUI version of CAAD 97.5% of the time. They tended to be the inexperienced computer users.

Expressed UI Preference vs. Percentage Session Time in Error

More people showed a preference towards the GUI version of CAAD to the CUI version of CAAD (see Figure 5.3-4) regardless of their percentage time in error operating under CAAD CUI.
Figure 5.3-4 Expressed UI Preference vs. Percentage Session Time in Error
(For CAAD CUI Session Times Considering Individual Session Times Which are
Less / Greater Than the Overall Mean CAAD Session Times)

Expressed UI Preferences vs. Times Accessing Help and Number of Failed Commands

Among individuals with no computer experience, 98.78% of individuals expressed a preference for the GUI version of CAAD, in spite of the number of times they needed to access help during their GUI session.

Individuals needing some assistance while operating under the GUI version of CAAD, who had some level of computer experience, preferred the CUI version of CAAD 65.12% to the GUI version 34.88%.

Individuals with lots of computer experience preferred the CUI version of CAAD 69.51% to the GUI version 10.98%. The remaining 19.51% had no UI preference.
Expressed UI Preferences vs. Number of Times Mislead by CUI

Individuals who were misled by the CUI interface tended to prefer the GUI version of CAAD (see Table 5.3-5).

Table 5.3-5 UI Preference vs. Number of Times Mislead During CAAD CUI Session

<table>
<thead>
<tr>
<th></th>
<th>Mislead 0 Times</th>
<th>Mislead 1 Time</th>
<th>Mislead 2 Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred GUI</td>
<td>10.5%</td>
<td>71.2%</td>
<td>100%</td>
</tr>
<tr>
<td>Preferred CUI</td>
<td>61.4%</td>
<td>28.8%</td>
<td>0%</td>
</tr>
<tr>
<td>No Preference</td>
<td>28.1%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Expressed UI Preference vs. Number of Regressive Commands Issued

Individuals, while operating under the CUI version of CAAD who made one or more regressive commands, tended to overwhelmingly prefer the GUI version of CAAD in the ratio of 3 to 1. In 100% of the cases where 2 or more regressive commands were made, users preferred the GUI version of CAAD.

Individuals with some to lots of computer experience, who tended to make one or more regressive commands while operating under the GUI version of CAAD, either preferred the CUI version of CAAD or had no expressed UI preference.

Expressed UI Preferences vs. Loss of Control Under Both UIs

In all instances, individuals operating under the CUI version of CAAD who lost control of the system preferred the GUI version of CAAD. Individuals who lost control of the GUI version of CAAD still preferred the GUI version of CAAD 96.4% of the time.
In both cases, individuals that lost control of either system were inexperienced computer users, and accordingly, these inexperienced users (on the whole) tended to prefer the GUI version of CAAD.

5.4 Summary of Preference vs. Age and Gender

There is no clear pattern that suggests that as age increases there is a marked preference toward one or the other UI. In general, the GUI was preferred across all age categories (see Figure 5.4-1).

Figure 5.4-1 Distribution Analysis - UI Preference vs. Age
UI Preference vs. Gender

Females generally preferred the GUI version of CAAD (i.e., 61.1% GUI preference, 30.5% CUI preference, and 8.4% no preference). This distinction was also true among males but not to the same degree (i.e., 55.3% GUI preference, 40.3% CUI preference, and 4.2% no preference). Comparing UI preference against gender (taking computer experience as a control variable) suggests that: (1) Individuals with little computer experience prefer the GUI regardless of gender (see Table 5.4-2); (2) Females with some computer experience prefer the GUI version of CAAD (see Table 5.4-3), whereas this preference is not as prominent with males; (3) Individuals with lots of computer experience prefer the CUI version of CAAD to the GUI version of CAAD regardless of gender (see Table 5.4-4) or had no expressed UI preference.

Table 5.4-2 UI Preference vs. Gender (n=250)
(No Computer Experience As a Control Variable)

<table>
<thead>
<tr>
<th></th>
<th>Female</th>
<th>Male</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>52.44%</td>
<td>46.34%</td>
<td>98.78%</td>
</tr>
<tr>
<td>CUI</td>
<td>1.22%</td>
<td>0%</td>
<td>1.22%</td>
</tr>
<tr>
<td>None</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>All</td>
<td>53.66%</td>
<td>46.34%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Table 5.4-3 UI Preference vs. Gender (n=250)
(Some Computer Experience As a Control Variable)

<table>
<thead>
<tr>
<th></th>
<th>Female</th>
<th>Male</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>39.53%</td>
<td>25.58%</td>
<td>65.12%</td>
</tr>
<tr>
<td>CUI</td>
<td>15.12%</td>
<td>19.77%</td>
<td>34.88%</td>
</tr>
<tr>
<td>None</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>All</td>
<td>54.65%</td>
<td>45.35%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 5.4-4 UI Preference vs. Gender (n=250)
(Lots of Computer Experience As a Control Variable)

<table>
<thead>
<tr>
<th></th>
<th>Female</th>
<th>Male</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>3.66%</td>
<td>7.32%</td>
<td>10.98%</td>
</tr>
<tr>
<td>CUI</td>
<td>31.71%</td>
<td>37.80%</td>
<td>69.51%</td>
</tr>
<tr>
<td>None</td>
<td>13.41%</td>
<td>6.10%</td>
<td>19.51%</td>
</tr>
<tr>
<td>All</td>
<td>48.78%</td>
<td>51.22%</td>
<td>100%</td>
</tr>
</tbody>
</table>
6 General Discussion

6.1 Summary of Findings

The average length of time an individual took to complete the CUI version of CAAD was approximately 29 minutes, with a standard deviation of about 12 minutes, and a standard error of approximately one minute. The average length of time an individual took to complete the GUI version of CAAD was approximately 31 minutes, with a standard deviation of about 13 minutes and a standard error of approximately one minute (Reference: Figure 5.1).

In summary, the average individual session time increased by approximately 7% using the GUI system. Why should this be the case? A number of factors help explain this difference:

1. Overall, since the tasks under both UIs were the same (see GOMs analysis, Section 4.2 and 4.3), real differences occur only in methodologies for goal-completion. As the GOMS analysis shows, the tasks are decidedly more complex under the GUI, due to the many movements required to accomplish the goals and subgoals.

2. The process of scanning the car graphic (see Appendix GUI Screen Captures) may have required additional time to focus on any desired action. Moreover, users may well have been intrigued (hence distracted) by complex graphical representations, and therefore their productivity slows down.

3. Selection of actions by pointing requires the user to locate the current position of the pointing device (cursor), locate the position of the desired alternative, and plan a trajectory from the current position of the pointing device to the desired alternative. To the extent, however, that any transformation or translation is required, time will be increased.
(4) CUs in general promote the scanning of menus, while GUIs may well promote browsing,
exploring, and hence wandering around the application as opposed to finding the shortest path
to accomplish an end. This may well have been the case with the CAAD study.

(5) Accidental error -- more than twice as much time was spent in error (8.4 minutes on average for
the GUI sessions as against 4.1 minutes on average for the CUI sessions). Making errors and
the recovery process take additional session time. This in turn contributes to the longer overall
operation time under the GUI. The key question really is, “Why were these errors made?” Did
they result from increased browsing and an inability to return control back to the goal path? Or
were errors made because of icon/graphic ambiguity? Was the car drawing (as depicted in the
Appendix) partially responsible for these longer session times? Furthermore, consider an
individual who has an ownership history with rear-engine automobiles. This sort of history may
lead the individual to scan the car graphic for engine related problems at the rear portion of the
vehicle. Not seeing any text appearing when the mouse is correspondingly placed, may lend
the individual to erroneously select an alternative “hot” region of the image map, thereby putting
the user several steps off the control path. At this stage, recovery may entail several visits to
the on-line help, once again needlessly adding to overall session time.

(6) Confusion -- some individuals, those with the least amount of GUI familiarity, may have had
problems co-ordinating movement of the mouse in the XZ plane against the corresponding
actions/movements of the cursor in the XY plane. Though it is hard to imagine that this
problem would plague the user throughout his/her entire GUI session (as people naturally begin
making the association between mouse pointing in the XY plane and the corresponding hand
movement in the XZ plane), it nonetheless could account for some additional GUI session time.
Yet one should expect the user's active learning throughout the session to minimize this effect, thereby marginalizing these delays.

On average, it took about 28 commands to complete a CUI session with a standard deviation of about 3 commands. It took an individual an average of 31 commands to complete a GUI session, with the standard deviation being about 4 commands (Reference: Figure 5.1). On-line help was accessed an average of 2 times per CUI session with a standard deviation of approximately 2. GUI users accessed help an average of 3 times per session with a standard deviation of approximately 2 (Reference: Figure 5.1-2). Operators of both the CUI and GUI versions repeated failed commands an average of 2 times per session, with a standard deviation of 1 (Reference: Figure 5.1-3). The CUI misled users once per session with a standard deviation of less than 1 time. The GUI misled users twice per session with a standard deviation of less than 1 time (Reference: Figure 5.1-4). The majority of users did not enter any regressive commands with the CUI version of CAAD. A GUI user entered an average of 2 regressive commands per session with a standard deviation of approximately 1 (Reference: Figure 5.1-5). In summary, these statistics tend to indicate that users were more productive using the CUI version of CAAD.
6.2 Summary of Hypotheses Tested

What may be concluded with respect to the original four hypotheses (listed below) from the CAAD study?

1. People with no computer experience will prefer using a GUI equipped expert system over the same CUI equipped expert system.
2. People with considerable computer experience will prefer using a CUI equipped expert system over the same GUI equipped expert system.
3. People with GUI experience will be more productive using a GUI equipped expert system over the same CUI equipped expert system.
4. People with GUI experience will make fewer errors using a GUI equipped expert system over the same CUI equipped expert system.

Examining each of these hypotheses in order, the findings presented in this study show that inexperienced computer users do have an interface preference (i.e., GUI). There is a dependent relationship ($\chi^2=142.247$, $p<.005$) between computer experience and interface preference. As experience increases there is a shift in preference to the CUI. Why would this be the case? Again a number of factors may contribute to this finding:

1. A number of inexperienced computer users either do not type or are largely unfamiliar with keyboard layouts. Hunting for keys may be more difficult for this group, hence there is a preference for the GUI version of CAAD. Yet if typing skills (or lack thereof) were the reason for an individual’s shift toward the GUI, then the fault may be attributed to the functionality of the input device as opposed to the user interface. Imagine if the individual could still operate the CUI version of CAAD without being hampered by keyboarding constraints, i.e., consider...
voice-scripting technology as an alternative to this (type of) input device. Would the results of this study prove to be otherwise? At the time this study was undertaken, these alternative technologies were in their infancy. Today, many observers claim that the technology has seen little improvement, and in fairness to the CUI version of CAAD, it is clearly an issue with which to reckon. As such, this same study could be recast to take voice-scripting into account and, for that matter, perhaps other input devices as alternatives to CUI interaction. If this were the case, would the data still show a marked preference toward the GUI? For balance, it would be necessary to apply voice-scripting to GUI sessions as well as to CUI sessions. Given this offset, testing would focus on variations of different input devices against both interfaces, eventually passing judgement on the preference issue.

(2) For new users of computer technology, the visual interest of a GUI may offer an added dimension of appeal. Users may have found less visual interest in the “vanilla” flavouring of the CUI version of CAAD.

(3) To some extent, new users of technology may have a preconceived idea that GUIs are easier to use. This is often reinforced by the media. This “preconditioning” may factor into play with respect to UI preference. Prior conditioning may well lay the foundation for an “inherent” preference toward GUIs. This preference may well be shaped by the user’s past educational experiences - experiences that are often fashioned around “models”. These models, like building a puzzle, or performing a chemistry experiment, etc., are often expressed pictorially, and this reinforces the innate preference towards the GUI.

(4) Inexperienced computer users focused their attention on two tasks with respect to the CAAD study - one being the application problem, i.e., automotive failure diagnostics and the other
using the computer. This task separation is not as apparent with respect to the experienced user of technology. Here the focus is the application problem area, as opposed to "struggling" with the computer system. As a result, the quest to solve the problems posed them was much more the driving force for this group of individuals. Consequently, efficiency of effort may have contributed to their preference towards the CUI version of CAAD.

People with lots of computer experience (regardless of their level of GUI familiarity) preferred the CUI version of CAAD to the GUI version ($\chi^2_{10}=136.03, p<.005$) in a ratio of 7 to 1. There are a number of explanations that tend to account for this finding. Perhaps the most likely explanation falls along the lines of the adage "Old habits die hard." These experienced users carry a history of past CUI system interactions from a period quite possibly extending back two or more decades. Over this period of time, individuals typically acquiesce to many of the shortcomings inherent in CUI interaction to the point of eventual proficiency within the interface. Like the saying "You can't teach an old dog new tricks," these people may have some disdain for system navigation by way of what was then a relatively new technology. More interesting, perhaps, is the fact that 20% of all the experienced users had no expressed UI preference. Why would this be the case? Perhaps they recognized the merits of both interfaces to the extent that they could easily solve any problems so presented. For this group of users, productivity towards an end goal may not be hampered by even a poorly designed interface. By their very nature they typically skirt around the intricacies of the UI to reach the desired objective. This group of power users represented only 5% of the population studied. Should this statistic hold true in the larger populations of system users, developers may in fact be able to market their wares largely based on content as opposed to context and presentation. This could be an interesting area for future researchers to pursue.
Due to the highly correlated relationship between GUI experience and computer experience (see Appendix: Pearson Correlation Coefficients), hypothesis (3), which states that people with GUI experience will be more productive (e.g., greater variable C8 scores), completing a using a GUI equipped expert system over the same CUI equipped expert system, is really suggesting that people with lots of computer experience will be more productive using a GUI equipped expert system over the same CUI equipped expert system. As a result of the CAAD study, based on the result from the analysis of variance [$F(2,247) = 123.34, p<.01$] this is clearly not the case. Moreover, this conclusion is further evidenced in light of the GOMs UI analysis provided in this study (see Figure 6.2.1).

One Way ANOVA (GUI Experience, log of the % of task completed per unit of time. (Legend: 0 = no GUI experience, 1 = some / little GUI experience, 2 = lots of GUI experience).

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI-exp</td>
<td>2</td>
<td>89.607</td>
<td>44.804</td>
<td>123.34</td>
<td>0.01</td>
</tr>
<tr>
<td>ERROR</td>
<td>247</td>
<td>89.722</td>
<td>0.363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>249</td>
<td>179.329</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**INDIVIDUAL 95% PCT CI'S FOR MEAN BASED ON POOLED SDEIV**

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>N</th>
<th>MEAN</th>
<th>SDEV</th>
<th>__________</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150</td>
<td>2.543</td>
<td>0.644</td>
<td>(- - -)</td>
</tr>
<tr>
<td>1</td>
<td>63</td>
<td>3.346</td>
<td>0.560</td>
<td>(- - -)</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>3.660</td>
<td>0.484</td>
<td>(-----)</td>
</tr>
</tbody>
</table>

**POOLED SDEV** = 0.6027

mean expressed as, $e^{\text{mean}}$ for each level of GUI experience equals $e^{2.543} = 9.52$ (no GUI experience), $e^{3.346} = 20.29$ (some / little GUI experience), $e^{3.660} = 39.67$ (lots of GUI experience).
Moreover, hypothesis (4), which states that people with GUI experience will make fewer errors (e.g., lower combined C15, C16 and C17 values) using a GUI equipped expert system over the same CUI equipped expert system, similarly did not hold true. Based on the result from the analysis of variance, we should reject the null hypothesis that the amount of errors does not vary among people with different levels of GUI experience \( [F(2,247) = 57.27, p<0.01] \).

Calculation details:
Three types of errors are considered here, i.e.,
total errors = a) failed commands (c15) + b) UI mislead (c16) + c) regressive commands (c17).

Taking a one-way ANOVA of total errors (response) against GUI experience (c5) (factor) leads to a stdev for level 0 (no GUI experience), level 1 (some / little GUI experience), and level 2 (lots of GUI experience) as level 0, stdev = 1.288,
level 1, stdev = 1.445,
level 2, stdev = 0.505 (respectively).

Associating these stdev (above) with the respective GUI experience (c5) values, i.e., if the value in c5 (GUI experience equals 0, then the associated new value (coded as C23) would be 1.288. Similarly, if the GUI experience value was 1, then the associated value in C23 would be 1.445 etc.).
The following further elaborates these findings:

1. All the individuals having lots of computer experience were able to complete the CUI based version of CAAD in less than the population mean time.

2. On average, users spent a greater percentage of their session time in error under the GUI.

3. More experienced users entered failed commands with the GUI system than with the CUI system.

4. Help was accessed by many experienced users within the GUI system (over 3/4) while only about 1/3 of experienced users had to access help in the CUI system.

These findings are significant for future application designers. Essentially, if application developers expect their application to be employed by users who have little computer experience, then it is wise to make this group of users feel comfortable with the application by providing a GUI.

Standardising this value by dividing by the respective stdev (C23), and taking a One Way ANOVA on this standardized value (coded as C27 response) against C5 (factor) leads to the information presented below (GUI Experience, Total errors).

(Legend: 0 = no GUI experience, 1 = some / little GUI experience, 2 = lots of GUI experience).

ANALYSIS OF VARIANCE ON c27 standardized error

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>gui-exp</td>
<td>2</td>
<td>114.54</td>
<td>57.27</td>
<td>57.29</td>
<td>0.01</td>
</tr>
<tr>
<td>ERROR</td>
<td>247</td>
<td>246.90</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>249</td>
<td>361.45</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

INDIVIDUAL 95 PCT CI'S FOR MEAN

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>N</th>
<th>MEAN</th>
<th>STDEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150</td>
<td>2.8520</td>
<td>0.9996</td>
</tr>
<tr>
<td>1</td>
<td>63</td>
<td>2.7023</td>
<td>0.9999</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>0.9098</td>
<td>1.0005</td>
</tr>
</tbody>
</table>

POOLED STDEV = 0.9998

<table>
<thead>
<tr>
<th></th>
<th>0.70</th>
<th>1.40</th>
<th>2.10</th>
<th>2.80</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This is a key point in the sense that expense, by way of application development time, is greater in a GUI environment than in a CUI environment. Hence, if an application developer knows something about the intended target audience, i.e., that if there is a specific level of computer knowledge and experience, then providing a CUI may prove to be all that is required.

In short, if user “comfort” is an issue, then it is worth the time, money and effort to develop a GUI front-end if not for the sole reason that the inexperienced user requires it, whether or not the application developer is producing a diagnostics system, an executive information system, or an information system provided to shoppers in a shopping mall, etc.

Furthermore, given a certain level of computer experience, it is possible for people to be equally productive with one interface or the other. From the application developer’s perspective, if the application layer of an existing application is going to be used locally, then one need not develop a (potentially expensive) GUI front-end. Perhaps resources should be directed toward user training to offset the user comfort level given to the inexperienced computer user by the GUI.

It might be argued that with the appropriate amount of user training, interface is not the most critical factor with respect to the user’s future acceptance of a system. If there is ensured user training, interface differences can possibly be overcome.
6.3 Preference, Learning and Performance

The GOMS analysis (as detailed in Section 4) helps to provide a dynamic description of user behaviour throughout the individual CAAD sessions. It depicts the specific learning opportunities through the larger goals and subgoals pertaining to the problems sets shown in Section 3 of this study. At the metalevel, the GOMS analysis provides a "roadmap" which demarcates the learning path for individual users of the CAAD systems. Clearly, a significant number of individuals were able to learn and demonstrate their understanding of complex automotive mechanics through the use of the CAAD systems. Perhaps consideration should be given to the estimate of the value of a system broader than the rule of thumb that "a system is only as good as it is useful"; estimate of a system's value should include a consideration of the learning experiences it provides.

In the present study, 46 people (Group One) of the 250 people studied were able to answer all of the questions from the problem sets with 100% accuracy under both versions of the CAAD system. With respect to this group of people, 61% had lots of GUI experience and 39% had some GUI experience. This suggests that about 1/5 of the population gathered a significant amount of knowledge in the area of automotive mechanics, although conclusions are limited because there are no measures of prior understanding of automotive mechanics. A fair assumption is that most participants were not automotive mechanics, for if they were, it is likely that they would be engaging in the repair themselves. Individuals of Group One were equally good working under either interface. At the same time, their productivity in either session is comparable with respect to errors that were made and session times.

One hundred and twenty nine people (Group Two) were able to answer at least three out of four problems with a high degree of accuracy. The mistakes that they made were primarily in the area
of causes and corrections. This group of individuals was able to determine correctly whether or not the mechanic’s diagnosis was reasonable, as well as able to suggest whether or not the mechanic should consider different causes for the problem before proceeding to do the work. This suggests that a significant amount of learning took place for Group Two.

Typically, participants falling into this Group Two category had difficulty assigning a priority to the automotive corrections for the problem under consideration. In simple terms, if a tire is flat it is reasonable to assume that there is a hole or puncture in the tire and that this would likely result from running over some foreign object, glass, nail, etc. The correction would be simple - to repair the tire using the appropriate patching systems for bias or radial systems. A less likely correction for such a problem would be to order a new rim to ensure proper tire seating. Essentially, if patching the tire by using the proper patching system is the most likely course of action to pursue (call this Correction A), and ordering a new rim were an alternative path of action to pursue (call this Correction B), then Group Two individuals may end up with an answer advancing Correction B as the best course to follow, even in light of realizing that Correction A is also a potential candidate.

In summary, 175 individuals out of the 250 people studied (i.e., Group One and Group Two combined) demonstrated considerable understanding of automotive mechanics after the use of CAAD CUI and CAAD GUI. Considering the 129 people who made some form of minor diagnostic error / failure correction, were more incorrect diagnoses made under the GUI version or the CUI version? Is it possible to know if greater learning took place under one version of the CAAD system or the other? Statistics reveal that there was a greater chance of a more complete diagnosis / more accurate diagnosis under the CUI version than under the GUI version. Of these individuals, 4.6% had slightly better test scores under the CUI version. Again, this is not surprising, for Group Two
was also misdirected more often using the GUI version, suggesting a possible reason for greater reversals of priorities in sequences. Moreover, some of the reversed priorities may be explained when one examines other sorts of related errors made under the GUI version, i.e., regressive and repeated commands. This is consistent with the rest of the findings of this study.

Perhaps more interesting is the suggestion that people may enjoy working with iconic-based interfaces more than with character-based ones, even though session error rate may be more frequent and learning outcomes may be less satisfactory. Yet these learning outcomes were only about 5% better under CUI sessions. It would be interesting to run this study again using domain knowledge from a less technical area of study, i.e., using an expert system not in the area of automotive failure, to consider if learning outcomes would still be better under the CUI mode of interaction. Possibly one could redo this study considering information in the Health Sciences regarding dietary planning as it pertains to wellness. However, most ES-based applications have been constructed in highly technical areas; for the high level of knowledge consistency, completeness and depth make this a more definable problem to approach from the knowledge engineer’s perspective and from the perspective of the system developers. Would the results achieved in non-technical areas be the same as those presented in this study, or does subject matter influence the sort of learning that takes place? By analogy, do children who enjoy a subject in school do better in that subject than those who do not enjoy that subject?

Seventy five individuals of the 250 studied (Group Three) had low scores on the questions presented in this study. For the most part, these individuals were unable to assess the mechanic’s solution to their car malfunction. Yet some of the individuals in Group Three were able to assess whether or not the mechanic’s diagnosis was reasonable, yet from the questions that followed, they
apparently could not consider the alternatives to the problem presented, nor could they come up with possible causes and their associated corrections. Simply put, an individual may have guessed correctly as to whether or not the mechanic's solution to their car problem was correct. The first question was answered correctly, but incorrect answers to questions two or three showed a failure to traverse the appropriate rule path from the CAAD system.

With respect to Group Three, was there a greater chance of failure under the CUI version or the GUI version? The results are inconclusive. Generally, this group experienced equal difficulty with both versions of CAAD. Presenting material in pictorial, point-and-click format proved to be no better in terms of learning outcomes than interaction through command line response. It would appear that if someone was to achieve a less than acceptable solution to the problem presented in the set of questions presented in this study, this individual would likely do no better using one interface over the other. Interestingly enough, 39 of these 75 individuals had no computer experience whatsoever; i.e., better than half the individuals who were unable to proceed successfully in the question section of this study had no computer experience. Of the remaining 36 individuals, 23 had just little to some computer experience. The interesting question to consider is the following: Why would some individuals do so poorly on these tests? Can this be attributed to (a) lack of computer skill; (b) lack of facility with the subject matter; (c) reading difficulties; (d) general malaise with respect to the study or so forth? The statistical evidence does not address all of these issues. It does, however, seem reasonable to attribute some lack of performance to a lack of computer experience, though it is also the case that 43 of 129 individuals who did reasonably well on the question section of this study had no computer experience. The positive finding is that individuals with no prior experience were able to learn a great deal. Was learning made any easier, operating under either
version of the UI? Results show such slight differences in performance associated with UI (favoring CUI) that learning, as measured by problems embedded in the expert system, was not affected by the UI. The sample size is small, however, so relationships between learning and UI are largely speculative.

Are learning and preference related? Do individuals who perform well on one system tend to prefer that system? The results of the chi-square test ($\chi^2_{df}=31.394, p<0.005$) suggest this. It can be concluded that there is a dependent relationship between test performance and UI preference; however, this finding is not surprising in light of the previous discussion.
A Reappraisal of the Findings from the CAAD Study

7.1 Two Interfaces for the Same Expert System

A number of issues arise from the attempt to study two alternate interface designs for the same underlying knowledge base, i.e.,

(a) How do I know the results of my study are valid?
(b) Maybe my CUI is just significantly better than my GUI (or vice versa)?
(c) Perhaps I’ve benefited by having designed the CUI first?

Response:

(a) There was a consistency in terms of design with respect to both interfaces. One UI did not provide added features or choices that were not available in the other UI. There is no structural difference in the design of CAAD CUI and CAAD GUI. They are built on the same heuristics, have the same inference engine, and use the same rule base. The only difference between these systems is the interface. The same problems were put forth for users under both UIs; i.e., the problem in the GUI session was not more complex than the problem in the CUI session. At the time this study was conducted, most GUIs were accessed via a 2-button, non-tracking enhanced, non-ballistic mouse. Given today’s input technology, which may be superior or at least different from yesterday’s input technology, it would be interesting to rerun the entire study using all or any of the following input devices on both UIs to compare the findings, i.e.:

1. Voice scripting (cf. IBM Aptiva)
2. Ballistic, tracking enhanced, 2-button mice (cf. Logitech)
3. Ballistic, tracking enhanced, 2-button mice with movement controlling wheel (cf. Microsoft Intelligent mouse)
4. Glide pad (cf. NEC Versa Notebooks)
5. Accu-stick (or track sticks) (cf. IBM Thinkpads, Toshiba Notebooks)
6. 3-axis Joysticks (Thrustmaster, Microsoft Sidewinder)

If anything, current research seems to suggest that these technologies may in fact reduce overall session times (Rutledge and Selker, 1990). Using any of these new technologies, given an input sensitive application, or an application that is reliant on large amounts and/or frequent amounts of input data (say 30 plus inputs per minute), a significant time savings would probably be realized using the former input devices. However, with respect to this specific study, users could solve this problem set in as little as 22 entries. Given even the maximum session time (i.e., 60 minutes), the frequency equals about 1 data entry every 3 minutes. This is certainly not heavily input driven. At the same time, one may argue that these newer technologies tend to be more “error free” in their use and application. Hence fewer user errors would ensue, there would be less time spent in error recovery, and the result would be shorter overall session times. This latter statement is perhaps questionable. If an error was accountable to the input device, it is unlikely this would alter the findings of this study to a significant degree either way, with respect to any of the previously cited input devices.

(b) It is probably difficult in general to find a GUI environment that is more user friendly than a GUI environment. However, more specifically, in terms of CAAD study, the same HELP features and information were provided to the user under both UIs.
(c) Both UIs were developed with the purpose of consistency in terms of the information conveyed to the user, task specifications, and task selection. These UIs were not developed in isolation one from another or as completely separate entities. The information they provide is completely interchangeable in all aspects.
7.2 Factors Affecting UI Preferences

Given the naturalistic context in which this study was conducted, and attendant limits in time and measures that could be imposed on participants, issues arise regarding factors that were not systematically varied, yet could influence results. These factors include prior knowledge of automotive mechanics, reading ability, motivation, the physical environment, personality, and so forth. As preface to this section, it is important to recognize that factors that did affect this study are, for the most part, randomly distributed across both CAAD sessions. Also, while factors may be differentially represented in the different populations sampled in this study, thus possibly confounding or obscuring important variables, the likelihood is that these factors are similarly represented in populations outside the volunteer group employed in this study. Accordingly, while there is a great deal additional to learn about variables affecting UI design, the present study represents an important early step in exploring two distinct and important interface access routes to a common body of knowledge.

There were also the many distracting factors associated with the garage setting that may have influenced results. Given an ideal environment, fewer errors and shorter session times could be expected. Yet users of a commercial release of these packages will employ them in garages similar to the one that was used for this study.

The individuals who came to the shop were asked if they had time to participate in this study. In light of their specific automotive failure, they could possibly make use of CAAD's knowledge base. At first glance, it would seem that they would be suitable if not excellent candidates for this study because of the meaningfulness of the problems to be solved relative to their own automobile failure. Hence, the general method for selecting people was based on whether (a) the individual had
time for the study, (b) there was an expressed interest, and (c) the individual had a problem that could be solved by CAAD. However, depending on the nature and severity of their automotive problems, their focus may not have been on the CAAD systems, with adverse affects on session times and frequency of errors made. Would candidates who were less distracted or preoccupied provide the same types of results? This seems quite likely, with these individuals perhaps making fewer errors, and experiencing shorter session times. Nonetheless, such effects would be randomly distributed across the population and across both systems.

Are the recorded session times accurate? Automobile mechanics occasionally entered the front office and asked to talk with the owner while he/she was operating the system. Such interruptions influenced the session's progress and goal focus. These interruptions make it difficult to return to the study. With interruptions, timing had to be stopped and other adjustments had to be calculated to attempt to overcome this break in continuity. Though this nuisance factor is randomly distributed, it could affect and presumably did add to the overall session times.

Moreover, session times are probably lengthened because of the general surrounding/environmental conditions and the deafening noise levels characteristic of garages. These circumstances do affect the user's ability to read and to pay attention to the problems at hand. Hammers drop, compressed air ratchets are started, cars go up and down on hoists, doors slam, cars drive over signal lines, doors on the garage bay open and close -- all these actions generate loud noise that is often intrusive. For that matter, during certain times of the day, when traffic is busier than usual, noise levels become significant considerations. Such disturbances are factored randomly against both systems.

Several other factors probably contributed to the time it took individuals to proceed through the
CAAD systems. These factors include (a) sensory capabilities, (b) lighting, and (c) sunshine on the screen due to the time of day.

The effect of the type of input device is one factor not assessed by this study. On session performance, user productivity, and preference would the results be the same had the input device had an impact on use interface preference? Would this have impacted on the overall session times, the types of errors that were made, and the frequency of the sort of errors that were made? If so, this is not as much of a problem as it might seem, given each of these devices can be tested on both CAAD sessions, thereby controlling for the unique performance variations inherent in each input device.

Another criticism might be that the cars drawn in the GUI version of CAAD are in 3D perspective but the mode of user interaction is in 2D. This rendering may have resulted in spatial ambiguity for some users and may consequently have thrown off their session times, navigational accuracy, and corresponding UI preference. If this is a problem, it is a minor one because as the user spends more time on the GUI version of CAAD, the impact of the spatial ambiguity would be minimized as learning and familiarization took place.

The GUI version of CAAD suffers from the problem of object recognition with respect to the constituent parts of the automobile. What if the individual had no idea where to look for the area of concern with respect to the manner in which the car was drawn? Just knowing the location of a part in the rendering may have been a difficult challenge for some individuals.

What about the problem of “lag” between screens with respect to user navigation through the GUI system? Did this predispose the GUI to longer session times? The problem did not affect session times because it was factored out using two different programming features, i.e., “quick”
screen refresh (based on image caching) and delay loops that pulled back the CUI when necessary.

Are end-users more productive over the long run when given better tools? The results of this study might have been different had individuals continued to use these systems over a period of three days, three weeks, or three months. Would session times still fall in the same line? Would expressed UI preference still be the same as it was on the first day of the study? What kind of recommendations are then to be offered to the developers of ES-based applications? Should these recommendations focus in on today’s users (i.e., the purchasers of the system) or tomorrow’s acclimatized user?

How do aesthetics factor into UI preference? If people could have customized their UI, yet maintained all of the consistent functionality between both versions of CAAD, would this impact on user preference? If the user could design and customize the interface to facilitate his/her personal needs, would the results of this study change dramatically?

Better technology would likely reduce all session times across both systems. Neither system was significantly advantaged by any of the factors described here. The problems that are presented by the GUI in particular are symptomatic of all GUIs.
8 Conclusions

Although graphics-based systems are gaining widespread popularity, with producers boasting of their ease of use and efficiency, it appears the actual benefits to user productivity and performance are not substantiated by studies.

Those users with little to no GUI or CUI experience preferred the GUI version of CAAD. However, as these levels of experience increased, the preference shifted to the CUI system. With respect to the GUI system usage, all statistics compiled on time, errors, and regressive commands showed a negative impact on user productivity and performance. Nevertheless, this did not seem to turn users away from the GUI system. Yet those users who encountered problems in the CUI system were quick to change their loyalties. Users may have been predisposed toward the graphics-based system. Analyzing this predisposition (if in fact it exists) could prove to be a topic for future research.

8.1 Recommendations for Developing Effective Interface Design

There are clearly many advantages that can be realized through the use of an object-oriented, graphical-user interface, but high reliance on this factor alone may not guarantee usability, end-user productivity, satisfaction and the acceptance of the system. The results of the CAAD study tend to demonstrate the potential benefits of object-based concepts with respect to the process of developing expert systems technology. Though not totally transparent in light of the statistical material presented in this thesis, much of the qualitative discussion offered by the participants involved in this study indicates that end-user acceptance lies in the designer’s ability to produce “intuitive” interfaces, ones that center around the specific of each user’s individual needs. Success and
productivity in an application seem to go hand in hand with interface preference.

Today, many applications are centered around the concept of users working together to produce "compound documents" or "team-based solutions." Though this trend is most apparent in many of the suite-based products, e.g., Lotus Smartsuite 97, and MS Office 97, significant acceptance continues in other less commercial areas as well. Though not as prevalent in the current batch of expert systems technology, the idea of team-focused solutions made possible by this technology is appealing and is reaching a number of diverse application areas (mostly notably in medical diagnostics but also as far reaching as transaction processing via the Internet). Given the many benefits associated with "user-centered" applications, especially with respect to system acceptance, the process of developing interfaces that suit the varied nature of group usage is a problem of immediate concern. Although this concern may be immediate, it is difficult to conceive of what the solution to this problem would entail. This may prove to be a major hurdle, one that must be overcome by developers of all information technology and system streams. One may find some solace in the hope of more user-centric operating system tools, those which would furnish developers with the means necessary to achieve these ends. Although, in the absence of

---

9IBM/Lotus make reference to their spreadsheet 123 v.5 as "the only spreadsheet for you and your team." Moreover, the word processor in this suit, e.g., WordPro 97, is promoted as "your team-based wordprocessor." Microsoft uses much the same promotional strategies and design framework in the Office 97 suite.

10Such systems include Micro Data Base Systems GURU (an expert system development environment for the PC); Harlequin's KNOWLEDGEWORKS (a knowledge visualization tool for Unix platforms including Sun, Sparc, RS/6000, Alpha, and SGI); Neuron Data's NE Expert OBJECT (a knowledge acquisition tool for PCs, Macintosh, Workstations, Minicomputers, and mainframes) to cite just a few of the many commercial products.

11Such tools include Acquired Intelligence's ACQUIRE-DSK (a software development kit for SCO Unix, Windows 3.1, Windows 95, and Windows NT); Talknowledge Corporation's M4 (GUI integration software modeling environment for Windows 95, Windows NT and System 7) among a wide variety of others.
these aids, and based on the study conducted in this thesis, what recommendations may be offered to developers of ES technology to facilitate user satisfaction, productivity and acceptance?

Designers of expert systems technology must, at the outset, determine what the end-users' objectives and tasks will be within the scope of the anticipated inference processing framed by the ES application. Preliminary task analysis should lead to a functional decomposition of the basic components and objects used in the expert systems, and to those tasks/associated behaviours and attributes that differentiate one kind of object from another, including those relationships of the objects to each other and to the user. Once this analysis is complete, developers may proceed to identify potential objects for users of the interface. Characteristics of the objects identified may be presented as either icons or data elements (encapsulated in either a form or a menu). Developers should attempt to map possible user behaviours and operations to specific kinds of interactions, such as menu commands or direct object manipulations.

Effective interface design is more than just following a set of rules. It begins with the realization that each user is unique, and even though there are attempts to classify users (as was in the case in the CAAD study), these classifications may be too broad to noticeably detail the interface. Interface development and implementation are best accomplished with a user-centered focus and design methodology. The success of this latter activity is contingent upon early planning and continued work through the software development process.

A "user-centered" focus must facilitate the user's sense of interface: control, directness, consistency, forgiveness, feedback, aesthetics, and simplicity.
8.1.1 Control

In all but a few instances, users did not lose control with either implementation of CAAD, though losing control and being in control are very different states. Actively empowering users by placing them in control of their applications, rather than passively subjecting their efforts to pre-programmed software scripts, makes for the most engaging ES design strategies. Although a seemingly compelling principle, interface design with active empowerment is not without difficulty. One noteworthy implication is the operational assumption that the user initiates system action based on his/her current knowledge of the problem. In this sense, the user plays an active rather than a reactive role. This makes sense if users have even the most rudimentary knowledge of the problem at hand, but little sense given a user with absolutely no domain knowledge. This denotes a shortcoming associated with the GUI implementation of CAAD. In the simplest of terms, it is easy for a user to actively scan a menu looking for such words as “brakes” or “air-conditioner” and consequently trace the problems that may be associated with the area of the automobile’s failure, but it is quite another matter to expect a user to be visually cognizant of the association between a car’s braking system and its wheels. Visually, users may associate a braking problem with the wheels of their car. But alternatively, they may possibly relate this sort of failure to other graphical objects presented in the user interface, such as the brake pedal, handbrake, or parking brake. System usage is traditionally in 3 dimensional space (moving a mouse, e.g., left, right, forward, reverse) diagrammatic aids (objects, icons, pictures, etc.) are in 2 dimensional space (i.e., X/Y plane) or in 2 ½ dimensional space (a simulated 3D mode). This shift in perspective can complicate GUI navigation for a number of individuals. Whether or not limited domain knowledge disadvantages the user’s ability to navigate within a graphically focused ES application, this is of significant
concern to developers. If command line entry was device restricted (i.e., touch screen, or otherwise), many menu options would have to be implemented on a mapping system that would allow the user to choose the shortest route from point "A" on a map to point "B". The programming cost of this sort interface would be considerable, to say the least.

8.1.2 Directness

Effective interface design should provide a vehicle for users to directly manipulate software representations of information. Whether dragging an object to relocate it from one part of an application to another, or navigating from screen to screen, users should see how the actions they take affect their progression toward their goals. On-screen visibility of information presented through the interface reduces the user's mental workload. It may be easier for some users to recognize on-screen commands rather than recall their syntax.

Familiar metaphors provide a direct and intuitive interface to user tasks. By allowing users to transfer their knowledge and experience, metaphors make it easier to predict and learn the behaviours of the ES application. In the case of the CAAD GUI system, the use of a single automobile as a representation of "all" automobiles was done for programming efficiency and convenience, as opposed to the precept of "intuitive" modelling. In some ways a "generic" representation of a car may run against the user's past experiences and serve as a source of confusion as opposed to utility. This is not to say that with some level of training and repetitive usage this problem would not be overcome. Metaphors provide a cognitive bridge between past user experiences and future system usage. Historically, developers outside the ES area have marketed their metaphors as ends in themselves (Microsoft vis. Windows Desktop).
Metaphors support user recognition rather than user recollection. They promote user comfort within an ES application, as employed in CAAD GUI.

8.1.3 Consistency

Consistency allows users to transfer existing knowledge to new tasks, learn new information more quickly, and focus more on tasks. Users need not spend time trying to remember the differences in interaction. By providing a sense of stability, consistency makes the interface familiar and predictable.

Consistency is important through all aspects of ES interface design, including but not limited to names of commands, visual presentations of information, and operational behaviour. To design consistency into an ES application, developers should examine consistency within the product, consistency with the operating environment, and consistency with metaphors.

8.1.4 Forgiveness

Many users like to explore the features of an interface and learn through trial and error. Discovery learning is a well-accepted methodology for a number of subject areas. It is particularly suited to ES technology, especially in light of the user’s ability to do scenario management and sensitivity analysis in the context of decision-making. Providing “forgiveness,” or interface tolerance, offers only the appropriate sets of choices to the user in the context of current application residence. Warnings to users should be generously advanced regarding potential situations where actions are irreversible or unrecoverable, or where damage to the system or data could ensue based on a specific input. Given the nature of ES technology, with its primary use completely directed to
an eventual end, e.g., a decision or recommendation, data integrity must be of the highest regard if informed decision-making is to be reached.

Even within the best designed interface, users may make mistakes. These mistakes can be both physical (accidentally pointing to the wrong object, command, or menu) and mental (making a wrong decision about which command or data to select). An effective ES interface design strategy avoids placing the user in a situation in which this is likely to occur. Furthermore, effective ES interfaces should accommodate potential user errors by making recovery easy and intuitive.

8.1.5 Feedback

ES developers should always strive to provide feedback for a user’s actions. Visual and sometimes audio cues should be presented with every user interaction to confirm that the application is responding to the user’s input and to communicate details that distinguish the nature of the action.

Effective feedback is timely, and should be presented as close to the point of the user’s interaction as possible. Even when a call to a database is invoked by the inference engine, possibly creating a system wait state, feedback should be provided to the user regarding the state of the process and the way to cancel that process if that is an option that does not impede system functionality. Nothing is more disconcerting to a user, especially to a new user of ES technology, than to encounter a screen that is unresponsive to input. Typically, users will tolerate only a few seconds of an unresponsive interface. Although CAAD was optimized to reduce any wait states through its highly structured database, users on several occasions appeared somewhat impatient when decisions were being reached. One can only speculate that long-term usage in this user group may make this frustration more pronounced, possibly leading to lower system acceptance.
8.1.6 Aesthetics

Much has been written in terms of the application of aesthetics to system design. Visual design is an important part of the ES interface. Visual attributes can provide valuable impressions and communicate important cues to the interaction behaviour of particular objects. At the same time, every visual element that appears on the screen potentially competes for the user’s attention. Of a more general nature, ES developers should attempt to provide a “pleasant” environment that clearly contributes to the user’s understanding of the information presented. A graphics or visual designer may be an invaluable resource to the developer with this aspect of the design. CAAD GUI used a fleshed-out, wire-frame generic car to assist users in terms of understanding the source of their automotive failure. These drawings were somewhat “lacking” in graphic qualities and this shortcoming could have influenced user preference between the two interfaces. Given the best of today’s 3D rendering technology, this drawing could have been significantly improved, and this improvement could have cast an even greater number of individuals to favour the GUI-based product. At the time CAAD CUI and CAAD GUI were developed, very little effective drawing software was available. (Packages like AutoCad were extremely difficult to run on an i286 with 1 Mb. of RAM). Clearly the effort required to produce a comparable image to today’s technology would be more extensive than the time required to write the actual system’s source code. As a percentage of this total effort, it is unlikely any designer would consider this to be a prudent investment.
8.1.7 Simplicity

An interface should be simple (not simplistic), easy to learn, and easy to use. It should also provide access to all functionality provided by an application. Maximizing functionality and maintaining simplicity work against each other in the interface. An effective design balances these objectives.

One way to support simplicity is to reduce the presentation of information to the minimum required for adequate communication. If at all possible, avoid wordy descriptions for command names and messages. This is important in light of the CAAD study, especially given the fact that some users experienced difficulty with reading comprehension and literacy. Excessive verbiage not only clutters the interface, but makes it difficult for users to easily extract essential information.

Another important design strategy is to use natural mappings, groups and semantics to structure interaction within the expert system. Imagine how difficult it would be to scan a menu of days of the week if the words were arranged alphabetically. The same holds true with respect to logically grouping areas associated with failure diagnostics. The arrangement and presentation of such elements may affect their meaning and association.

ES developers may also help users manage complexity by using a concept called progressive disclosure. Progressive disclosure involves the careful organization of all information so that it is shown only at its appropriate time. By “hiding” information presented to the user, developers can reduce the amount of information to process. Clicking a menu may display a number of options (whether grayed out or not). This still takes some time on the part of the user to scan, process and eventually act upon. As an alternative, one carefully orchestrated dialogue box has the potential to reduce the number of menu options dramatically.
Progressive disclosure does not imply using unconventional techniques for revealing information, such as requiring modifier keys as the only way to access basic functions, or forcing users down an extended sequence of hierarchical interactions. It is a technique that has enjoyed considerable success in a number of application programming environments and developer tools today.

8.1.8 Costs to the Developer: Dollars vs. Design Tradeoffs

The development of expert systems today is guided by commercial concerns. Most developers of information technology find themselves operating within "windows of opportunity." The faster an application can be effectively brought to market, the greater the competitive edge, and ideally the longer the selling life cycle. Design is at best a tradeoff between sales and customer acceptance. In light of this tension, a number of additional factors influence the design of an ES application specifically in terms of its interface development. Marketing considerations may require the developer to deliver a product with a minimal design process. For that matter, competitive analysis may force the developer to consider additional features. Shortcuts and additional functionality can and generally do affect the quality of the product. Last-minute changes often cannot be properly coded within the scope of the interface, and as such a potential source of user frustration is bound to transpire with system usage. There is no simple equation to determine when a design tradeoff is appropriate. Every additional feature potentially affects the application's performance, complexity, stability, tolerance with the scope of the operating system, maintenance, and the support costs of an application. Moreover, it is typically more difficult to fix a design problem after the release of a product because users may adapt or even become dependent on a peculiarity in the design. Though
developers may attempt to provide as much application simplicity to the end-user as possible, making something simple to use often requires a good deal of work and code. Features implemented by a small extension in the application code do not necessarily have a proportional effect in a user interface. For example, if the primary task is selecting a single object, extending it to support the selection of multiple objects could make the frequent, simple task more difficult to carry out.

An effective user-centered design process involves a well-formulated conceptual underpinning and user testing at salient development points. The initial work on the ES interface’s design can be the most critical because during this phase the developer must decide the general shape of the product, considering the domain knowledge to be warehoused within the various repositories.

If this foundational work is flawed, it will prove most difficult to correct afterwards. In light of building user-centered designs, the importance of the interface with respect to end-user system acceptance cannot be stressed enough. This part of the process involves not only defining the objectives and features of the interface, but also understanding who the potential users are of the product, their tasks, intentions, goals, and expectations. Developers cannot learn too much with respect to such factors as the users’ background, age, gender, expertise, experience level, physical limitations, and special needs; their work environment, social and cultural influences, and physical surroundings; and their current task organization -- the steps required, the dependencies, redundant activities, and the output objective. An order-entry system ES may have very different users and requirements than does an information kiosk ES.

At this point, the developer must define the conceptual framework to represent his/her product with the knowledge and experience of the target audience. Ideally, one should create a design model that fits the user’s conceptual view of the tasks to be performed. One should consider the basic
organization and different types of metaphors that can be employed. Observing users at their current tasks can provide ideas on effective metaphors to use.

Ultimately, there are no magic recipes that will guarantee user acceptance of any ES-based application. What ultimately makes a system useful are the benefits the users derive from it. What constitutes these benefits is fuel for another paper.
9 Developing Expert Systems: Using the Results Presented in This Study

ES-based applications are being developed in a wide variety of areas today. Most of these systems now come equipped with a GUI. GUIs continue to evolve. People for a number of different reasons tend to prefer GUIs over similar functionality presented by CUIs. Although this preference is clear, the material developed in this thesis, with respect to productivity and errors, make it all the more important to construct the GUI on a base of well-founded set of principles.

ES-based technology is also claiming a share of the Internet marketplace. Essentially, transaction management systems -- which provide businesses with the ability to negotiate among themselves and their customers via secure electronic commerce servers -- place significant demands on the developers of ES-based technology.

Whether these systems are at the back-end of an HTML page, acting as a transaction intermediary to an SQL server, or simply as a component facilitating an on-line search, this technology continues to make incursions into the ever-increasing Information Age.

ES-based applications are unique in the sense that they heavily tax the cognitive powers of most of their users, especially where the subject domain is in a highly technical area (as the case in the study presented in this thesis). Failure recognition in the automotive industry is just one potential application of this technology. Many other opportunities continue to present themselves in a globally shared information repository. Accessing this critical mass of information will be made all the more effective using the processing capabilities of future ES-based applications.

ES-based technology may be able to provide the buffer between today's growing mass of information and much of the chaos that many experience when surfing the Internet. Intelligent parsing technology can allow users to make sentence-based requests from search engines throughout
the Information Superhighway. When this technology is coupled with the many advances that have been made in the area of fuzzy logic, a new generation of search engines will be born.

Tomorrow's users will likely face a new breed of mixed-modal interfaces for their use. These types of interfaces will likely exhibit the best characteristics of both the CUI and the GUI; they will be designed for the purpose of providing one exceptionally powerful "super" interface. This "super" interface will be dynamic in nature, customizable "on-the-fly," and based on the user's varied and distinct preferences.

Designers of tomorrow's ES-based applications will play an integral part in the development of such user interfaces. Ideally the findings presented in this study will facilitate that end.
References


Appendices
Appendices

(a) CUI Screen Captures
(b) GUI Screen Captures
(c) CAAD Experiment Data (All Subjects)
(d) Pearson Correlation Coefficients
The goal of CAAD is to provide instruction to the customer or do-it-yourself mechanic in the area of automotive trouble shooting and diagnosis.

Using the principles of expert systems based DEX, this system integrates a complex series of menus (in the form of a rule base), along with a database to instruct the user and provide decision making capabilities for repairing possible automotive malfunctions.

INSTRUCTIONS:

When asked a question respond by choosing the appropriate letter: (A, B, C, etc.) or move the highlighted box using the arrow keys) over the selection and press the ENTER key.

Press the ENTER key now to continue.

CUSTOMER LICENSE AGREEMENT:

You may use this product on any compatible hardware that you own or use.

You may not:

- Use this product in a multiple site or multiple user environment.
- Use this product in an interactive cable television system without addition copies of CAAD for each system or a multi-user license from the developer, Wallace John Whistans-Smith.
- Make copies of the CAAD User Manual or program disks.
- Alter the CAAD software in any way, including modification for use on non-compatible hardware.

Press the ENTER key now to continue.
MAIN MENU

1) DETAILED DIAGNOSTICS
2) QUICK DIAGNOSTICS

Press the number of the option you wish to choose.

X = Exit    F1 = Help

SCREEN-04

CAAD QUICK TROUBLE SHOOTING

A) BRAKE SYSTEM
B) ENGINE MECHANICAL
C) ENGINE PERFORMANCE
D) EXTERIOR LIGHTS
E) HEADLIGHT SWITCH
F) RACK & PINION STEERING GEAR
G) STEERING GEAR EXCEPT RACK & PINION

Enter the letter of the applicable problem area.

X = Exit    P = Previous Screen    F1 = Help
Problem: HIGH OIL PRESSURE

Possible Causes (listed in order of likelihood of occurrence):
A) Improper oil viscosity
B) Oil pressure gauge or sending unit inaccurate
C) Oil pressure relief valve sticking closed

X * Exit  P * Previous Screen  F1 * Help
Problem: HIGH OIL PRESSURE
Details: IMPROPER OIL VISCOSITY

Corrections to problem are as listed below.

The following steps should be taken:
Drain and refill crankcase with correct viscosity oil.

Possible causes (listed in order of likelihood of occurrence):
A) Improper oil viscosity.
B) Oil pressure gauge or sending unit inaccurate.
C) Oil pressure relief valve sticking closed.

X = Exit  P = Previous Screen  F1 = Help
Problem: HIGH OIL PRESSURE
Details: OIL PRESSURE GAUGE OR SENDING UNIT INACCURATE

Corrections to problem are as listed below.
The following steps should be taken:
Inspect oil pressure gauge or sending unit and replace as necessary.

SCREEN-09

Problem: HIGH OIL PRESSURE
Details: OIL PRESSURE GAUGE OR SENDING UNIT INACCURATE

The following steps should be taken:
Inspect oil pressure gauge or sending unit and replace as necessary.

SCREEN-10

Problem: HIGH OIL PRESSURE

Enter the letter of the possible cause.

POSSIBLE CAUSES (listed in order of likelihood of occurrence):
A) Improper oil viscosity.
B) Oil pressure gauge or sending unit inaccurate.
C) Oil pressure relief valve sticking closed.

Y = Exit    P = Previous Screen    FI = Help
Problem: HIGH OIL PRESSURE
Details: OIL PRESSURE RELIEF VALVE STICKING CLOSED

 Corrections to problem are as listed below.

The following steps should be taken:
Remove and inspect oil pressure relief valve assembly.

SCREEN-11

SCREEN-12

CAAD MAIN MENU
1) DETAILED DIAGNOSTICS
2) QUICK DIAGNOSTICS
Press the number of the option you wish to choose.
Choose any of the following options by typing the associated letter with it.

POSSIBLE CAUSES:
A) Vehicle vibrates at high speeds;
B) Vehicle vibrates at low speeds;
C) Vehicle pulls in one direction;
D) Tire wear;
E) Vehicle wavers.

X = Exit  P = Previous Screen  F1 = Help
Solution: Check tire balance.

Tire balance can be checked in the following ways:
1) Rotate tires so that the tire to be balanced becomes a drive wheel.
2) Jack up wheel to be balanced only. Make sure car is very steady.
3) Enter car, start engine, and place in DRIVES.
4) Drive car at approximately 70 kph.
5) If car shakes then the tire is out of balance.
6) Repeat procedure for the rest of the tires.

Possible Causes of problem (in order of likelihood):
1) Check tire balance
2) Warped tires
3) Loose wheels
4) Bent axle or hub

Enter the number of the appropriate option (i.e., 1, 2, 3, or 4).
Model: CAMARO
Problem: FRONT END
Details: Vehicle Vibrates at High Speeds

Solution: Warped Tires.
Warped tires can be checked in the following ways:
1) Drive car slowly.
2) If car moves steadily from side to side, one or more tires are warped.
3) Check up one wheel (if it is a drive wheel, place the car in neutral).
4) While looking directly at tire tread, spin the wheel.
5) If treads move from side to side, tire is warped.

Possible Causes of problem (in order of likelihood):
1) Check tire balance
2) Warped tires
3) Loose wheel(s)
4) Bent axle or hub

Enter the number of the appropriate option (i.e., 1, 2, 3, or 4).
**Model:** CANARD
**Problem:** FRONT END
**Details:** Vehicle Vibrates at High Speeds

**Solution:** Loose Wheels.
Loose wheels can be checked in the following ways:
1. Do not jack up car.
2. While car is on the ground, tighten each lug nut.
3. Repeat procedure for each wheel.
4. Care should be exercised so as not to over tighten each lug nut.

X = Exit  P = Previous Screen  F1 = Help

**SCREEN-24**

**Model:** CANARD
**Problem:** FRONT END
**Details:** Vehicle Vibrates at High Speeds

**Possible Causes of Problem (in order of likelihood):**
1. Check tire balance
2. Worn tire(s)
3. Loose wheel(s)
4. Bent axle or hub

Enter the number of the appropriate option (i.e., 1, 2, 3, or 4).

X = Exit  P = Previous Screen  F1 = Help
Model: CAMARO
Problem: FRONT END
Details: Vehicle Vibrates at High Speeds

Solution: Bent axle or hub

A bent axle can be checked in the following way:
1) Raise the vehicle and remove the wheels and brake drums.
2) Remove the differential cover.
3) Remove the differential pinion shaft lock-screw and pinion shaft.
4) Push the flanged end of the axle shaft toward the center and remove.
5) Inspect by rolling shaft on a flat surface to see if bent.
Appendix B - GUI Screen Captures
CAAD

1.

The goal of CAAD is to provide instruction to the customer and/or do-it-yourself mechanic in the area of automotive trouble shooting and diagnostics.

Using the principles of expert systems based ICA, this system integrates a complex series of menus (in the form of a rule base) along with a database to instruct the user and provide decision making capabilities for repairing possible automotive malfunctions.

INSTRUCTIONS:

When asked a question respond by choosing the appropriate letter (A, B, C, etc.) or move the highlighted box (using the arrow keys) over the selection and press the ENTER key.

Please mouse pointer over start option and double click left mouse button.

2.

CAAD

COMPUTER ASSISTED AUTOMOTIVE DIAGNOSTICS

CUSTOMER LICENSE AGREEMENT:

You may use this product on any compatible hardware that you own or use.

You may not:

- Use this product in a multiple site or multiple user environment.
- Use this product in an interactive television system without addition copies of CAAD for each system or a multi user license from the developer, Wallace John Whistance-Scots.
- Make copies of the CAAD User Manual or program disks.
- Alter the CAAD software in any way, including modification for use on non-compatible hardware.

Please mouse pointer over desired option and double click left mouse button.
GUI SCREEN 3.

CAAD MAIN MENU

1) DETAIL DIAGNOSTICS
2) QUICK DIAGNOSTICS

Place mouse pointer over desired option and double click left mouse button...

Exit Help

GUI SCREEN 4.

CAAD Quick Trouble Shooting

1) FRAME SYSTEM
2) ENGINE PERFORMANCE
3) DECELERATION
4) BRAKE SYSTEM
5) REAR AXLE SYSTEM
6) TAILGEAR SYSTEM
7) TAILGEAR SYSTEM TAIL & PIVOT

Place mouse pointer over any problem area and double click left mouse button...

Exit Previous Screen Help
GUI SCREEN 7.

Problem: ENGINE MECHANICAL
Details: HIGH OIL PRESSURE
Cause: IMPROPER OIL VISCOSITY

1. Drain and refill crankcase with correct viscosity oil.

GUI SCREEN 8.

Problem: Engine Mechanical
Details: Oil Pressure Gauge High
Solution: Possible causes of problem (in order of likelihood)

1. Improper oil viscosity
2. Oil pressure gauge or sender unit inaccurate
3. Oil pressure relief valve sticking closed

Mouse pointer over any of the solutions above and double click left mouse button...
GUI SCREEN 9.

Problem: ENGINE MECHANICAL
details: HIGH OIL PRESSURE
cause: OIL PRESSURE GAUGE / SENDING UNIT INACCURATE

Corrections to problem are as listed below.
The following steps should be taken:

1. Inspect oil pressure gauge or sending unit and replace as necessary.

GUI SCREEN 10.

Problem: Engine Mechanical
details: Oil Pressure Gauge High
solution: Possible causes of problem (in order of likelihood)

1) Improper oil capacity
2) Oil pressure gauge or sending unit inaccurate
3) Oil pressure relief valve sticking closed

Please review whether any of the solutions above and decide click left mouse button...

Exit Previous Screen Help
GUI SCREEN 11.

Problem: ENGINE MECHANICAL
Details: HIGH OIL PRESSURE
Cause: OIL PRESSURE RELIEF VALVE STICKING CLOSED

Corrections to problem are as listed below. The following steps should be taken:

1. Remove and inspect oil pressure relief valve assembly

GUI SCREEN 12.

CAAD MAIN MENU

1) DETAILED DIAGNOSTICS
2) QUICK DIAGNOSTICS

Place mouse pointer over desired option and double click left mouse button...
GUI SCREEN 13.

Automotive Manufacturer

Place mouse pointer over applicable manufacturer and double click left mouse button...

Exit  Previous-Screen  Help

GUI SCREEN 14.

GM

Place mouse pointer over applicable option and double click left mouse button...

Exit  Previous-Screen  Help
<table>
<thead>
<tr>
<th>1) Bump</th>
<th>2) Front</th>
<th>3) Center</th>
<th>4) Rear</th>
</tr>
</thead>
<tbody>
<tr>
<td>5) Left Side</td>
<td>6) Right Side</td>
<td>7) Left Rear</td>
<td>8) Right Rear</td>
</tr>
</tbody>
</table>

Place mouse pointer over appropriate area and double-click left mouse button.

---

**GM Chevrolet Camaro**

Place mouse pointer over area where problem is occurring and double-click left mouse button.

---

**GUI SCREEN 15.**

**GUI SCREEN 16.**
16a. GUI SCREEN

GM Chevrolet Camaro

Place mouse pointer over area where problem is occurring and double click left mouse button.

Exit Previous Screen Help

16b. GUI SCREEN

GM Chevrolet Camaro

Place mouse pointer over area where problem is occurring and double click left mouse button.

Exit Previous Screen Help
GUI SCREEN 16c.

![GUI SCREEN 16c.](image)

GUI SCREEN 16d.

![GUI SCREEN 16d.](image)
GUI SCREEN 17.

GUI SCREEN 18.
GUI SCREEN 19.

Problem: Front End
Details: Vehicle Vibes at High Speeds
Solution: Check tire balance.

Tire balance can be checked in the following ways:

1. Park vehicle. Turn off engine. Do not lock the wheels.
2. Level the car on a level surface. Make sure the car is parked on a level surface.
3. Check tire pressures. Ensure that tire pressures are within the recommended range.
4. Check for any unbalanced accessories or items in the car.
5. Check for any loose or damaged components.
6. Check for any unbalanced or damaged components.
7. Check for any unbalanced or damaged components.

Place mouse pointer over any of the solutions above and double click on left mouse button...

Exit 
Previous Screen 
Help

GUI SCREEN 20.

Problem: Front End
Details: Vehicle Vibes at High Speeds
Solution: Possible causes of problem (in order of likelihood)

1. Check tire balance.
2. Worn front tires.
3. Loose wheel(s).
4. Bent axle or hub.

Place mouse pointer over any of the solutions above and double click on left mouse button...

Exit 
Previous Screen 
Help
GUI SCREEN 21.

Problem: Front End
Details: Vehicle Vibration at High Speeds
Solution: Warped Tires

Warped tires can be checked in the following ways:

1. Check tie rod ends
2. Check shock absorbers for trouble or noise
3. Check steering linkage for trouble or noise
4. Check tie rod ends for trouble or noise

Place mouse pointer over any of the solutions above and double click left mouse button.

GUI SCREEN 22.

Problem: Front End
Details: Vehicle Vibration at High Speeds
Solution: Possible causes of problem (in order of likelihood)

1. Check tie rod ends
2. Check shock absorbers for trouble or noise
3. Check steering linkage for trouble or noise
4. Check tie rod ends for trouble or noise

Place mouse pointer over any of the solutions above and double click left mouse button.
GUI SCREEN 23.

Problem: Front End
Details: Vehicle Vibration at High Speeds
Solution: Loose Wheel
Loose wheels can be checked in the following ways:

1. Check for loose bolt
2. Check for loose nut
3. Check for loose brake caliper

Place mouse pointer over any of the solutions above and double click left mouse button...

Exit Previous Screen Help

GUI SCREEN 24.

Problem: Front End
Details: Vehicle Vibration at High Speeds
Solution: Possible causes of problem (in order of likelihood)

1. Check for loose bolt
2. Check for loose nut
3. Check for loose brake caliper

Place mouse pointer over any of the solutions above and double click left mouse button...

Exit Previous Screen Help
Problem: Front End
Cause: Wheel Vibration at High Speeds
Solution: Bent Axle or Hub

Loose wheels can be checked in the following ways:

1. Loosen the wheel lug nuts six to eight times.
2. Check the wheel flange for cracks.
3. Make sure the brake lines and hoses are properly connected.
4. Check for loose brake pads or rotors.

Please mouse pointer over any of the solution steps and double-click left mouse button.

Options:
- Exit
- Previous Screen
- Help
Appendix C - CAAD Experiment Data (All Subjects)
<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
<th>C15</th>
<th>C16</th>
<th>C17</th>
<th>C18</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>59</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>31</td>
<td>11</td>
<td>24</td>
<td>3.5</td>
<td>18</td>
<td>23</td>
<td>1</td>
<td>1.3</td>
<td>81</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>134</td>
<td>35</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>31</td>
<td>0</td>
<td>15</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>130</td>
<td>57</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>10</td>
<td>3.5</td>
<td>77</td>
<td>34</td>
<td>0</td>
<td>0.7</td>
<td>79</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>129</td>
<td>57</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>10</td>
<td>4.8</td>
<td>10</td>
<td>33</td>
<td>2</td>
<td>0.5</td>
<td>77</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>141</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>1</td>
<td>14</td>
<td>5.5</td>
<td>25</td>
<td>32</td>
<td>6</td>
<td>3.5</td>
<td>66</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>5.3</td>
<td>11</td>
<td>27</td>
<td>2</td>
<td>2.1</td>
<td>81</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>5.3</td>
<td>14</td>
<td>27</td>
<td>2</td>
<td>2.3</td>
<td>81</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>2</td>
<td>17</td>
<td>4.3</td>
<td>19</td>
<td>22</td>
<td>0</td>
<td>0.9</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Subject Data (CUU)
<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
<th>C15</th>
<th>C16</th>
<th>C17</th>
<th>C18</th>
<th>C19</th>
<th>C20</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>26</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>27</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>28</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>29</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>31</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>32</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>33</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>34</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>35</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Subject Data (CU)
<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
<th>C15</th>
<th>C16</th>
<th>C17</th>
<th>C18</th>
<th>C19</th>
<th>C20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>1</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.8</td>
<td>83</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>5</td>
<td>27</td>
<td>32</td>
<td>2</td>
<td>4.4</td>
<td>89</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>26</td>
<td>32</td>
<td>2</td>
<td>4.6</td>
<td>84</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>4</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>1</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>4</td>
<td>6.5</td>
<td>2</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>1</td>
<td>36</td>
<td>7</td>
<td>2</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>4</td>
<td>6.5</td>
<td>2</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>1</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>26</td>
<td>32</td>
<td>2</td>
<td>4.6</td>
<td>84</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>26</td>
<td>32</td>
<td>2</td>
<td>4.6</td>
<td>84</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>3</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>3</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>46</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>4</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>46</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>4</td>
<td>13</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>28</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>9</td>
<td>18</td>
<td>6</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>4.6</td>
<td>87</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>3</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>5.4</td>
<td>1</td>
<td>36</td>
<td>34</td>
<td>3</td>
<td>2.5</td>
<td>60</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Subject Date (GUI)
Appendix D - Pearson Correlation Coefficients
## Pearson Correlation Coefficients for CUI Sessions

<table>
<thead>
<tr>
<th></th>
<th>age</th>
<th>gender</th>
<th>comp-exp</th>
<th>gui-exp</th>
<th>time</th>
<th>t-tah</th>
<th>t-time</th>
<th>t-time-err</th>
<th>t-time-err Commands</th>
<th>Help</th>
<th>time-bhp</th>
<th>trav/unf</th>
<th>failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>0.007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comp-exp</td>
<td>0.041</td>
<td>0.640</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gui-exp</td>
<td>0.004</td>
<td>0.619</td>
<td>0.643</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>0.006</td>
<td>0.615</td>
<td>0.654</td>
<td>0.443</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t-tah</td>
<td>0.015</td>
<td>0.049</td>
<td>0.309</td>
<td>0.424</td>
<td>0.740</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t-time</td>
<td>0.049</td>
<td>0.041</td>
<td>0.576</td>
<td>0.652</td>
<td>0.624</td>
<td>0.700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t-time-err</td>
<td>0.046</td>
<td>0.039</td>
<td>0.552</td>
<td>0.475</td>
<td>0.416</td>
<td>0.449</td>
<td>0.463</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t-time-err Commands</td>
<td>0.017</td>
<td>0.024</td>
<td>0.781</td>
<td>0.650</td>
<td>0.649</td>
<td>0.655</td>
<td>0.614</td>
<td>0.620</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help</td>
<td>0.007</td>
<td>0.019</td>
<td>0.806</td>
<td>0.593</td>
<td>0.790</td>
<td>0.672</td>
<td>0.624</td>
<td>0.645</td>
<td>0.910</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time-bhp</td>
<td>0.023</td>
<td>0.048</td>
<td>0.644</td>
<td>0.536</td>
<td>0.492</td>
<td>0.304</td>
<td>0.443</td>
<td>0.477</td>
<td>0.577</td>
<td>0.772</td>
<td>0.904</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trav/unf</td>
<td>0.020</td>
<td>0.036</td>
<td>0.765</td>
<td>0.671</td>
<td>0.750</td>
<td>0.444</td>
<td>0.605</td>
<td>0.472</td>
<td>0.706</td>
<td>0.812</td>
<td>0.850</td>
<td></td>
<td></td>
</tr>
<tr>
<td>failed</td>
<td>0.040</td>
<td>0.020</td>
<td>0.627</td>
<td>0.510</td>
<td>0.544</td>
<td>0.214</td>
<td>0.454</td>
<td>0.492</td>
<td>0.197</td>
<td>0.794</td>
<td>0.517</td>
<td>0.699</td>
<td>0.824</td>
</tr>
<tr>
<td>missed</td>
<td>0.022</td>
<td>0.019</td>
<td>0.025</td>
<td>0.566</td>
<td>0.439</td>
<td>0.239</td>
<td>0.422</td>
<td>0.459</td>
<td>0.214</td>
<td>0.787</td>
<td>0.593</td>
<td>0.651</td>
<td>0.806</td>
</tr>
<tr>
<td>control</td>
<td>0.036</td>
<td>0.072</td>
<td>0.752</td>
<td>0.152</td>
<td>0.302</td>
<td>0.114</td>
<td>0.158</td>
<td>0.052</td>
<td>-2.223</td>
<td>0.217</td>
<td>0.284</td>
<td>0.002</td>
<td>0.105</td>
</tr>
<tr>
<td>pref</td>
<td>0.006</td>
<td>0.011</td>
<td>0.707</td>
<td>0.629</td>
<td>0.730</td>
<td>0.122</td>
<td>0.637</td>
<td>0.445</td>
<td>-0.005</td>
<td>-0.700</td>
<td>0.726</td>
<td>0.811</td>
<td>0.659</td>
</tr>
</tbody>
</table>

### Missed, Regress, Control

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>missed</td>
<td>0.403</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regress</td>
<td>0.043</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>control</td>
<td>-0.193</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pref</td>
<td>-0.600</td>
<td>-0.326</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.159</td>
</tr>
</tbody>
</table>
### Pearson Correlation Coefficients for GUI Sessions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.007</td>
<td>0.067</td>
<td>0.041</td>
<td>0.040</td>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.049</td>
<td>0.046</td>
<td>0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
</tr>
<tr>
<td>0.016</td>
<td>0.041</td>
<td>0.040</td>
<td>0.009</td>
<td>-0.090</td>
<td>-0.643</td>
<td>-0.445</td>
<td>-0.507</td>
<td>-0.007</td>
<td>-0.035</td>
<td>0.055</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.055</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
</tr>
<tr>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.040</td>
<td>0.007</td>
<td>0.033</td>
<td>0.172</td>
<td>0.173</td>
<td>0.462</td>
<td>0.375</td>
<td>0.012</td>
<td>0.044</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
</tr>
<tr>
<td>0.007</td>
<td>0.067</td>
<td>0.041</td>
<td>0.040</td>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.049</td>
<td>0.046</td>
<td>0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
</tr>
<tr>
<td>0.016</td>
<td>0.041</td>
<td>0.040</td>
<td>0.009</td>
<td>-0.090</td>
<td>-0.643</td>
<td>-0.445</td>
<td>-0.507</td>
<td>-0.007</td>
<td>-0.035</td>
<td>0.055</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.055</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
</tr>
<tr>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.040</td>
<td>0.007</td>
<td>0.033</td>
<td>0.172</td>
<td>0.173</td>
<td>0.462</td>
<td>0.375</td>
<td>0.012</td>
<td>0.044</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
</tr>
<tr>
<td>0.007</td>
<td>0.067</td>
<td>0.041</td>
<td>0.040</td>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.049</td>
<td>0.046</td>
<td>0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
</tr>
<tr>
<td>0.016</td>
<td>0.041</td>
<td>0.040</td>
<td>0.009</td>
<td>-0.090</td>
<td>-0.643</td>
<td>-0.445</td>
<td>-0.507</td>
<td>-0.007</td>
<td>-0.035</td>
<td>0.055</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.055</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
</tr>
<tr>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.040</td>
<td>0.007</td>
<td>0.033</td>
<td>0.172</td>
<td>0.173</td>
<td>0.462</td>
<td>0.375</td>
<td>0.012</td>
<td>0.044</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
<td>0.012</td>
<td>0.170</td>
<td>0.375</td>
</tr>
<tr>
<td>0.007</td>
<td>0.067</td>
<td>0.041</td>
<td>0.040</td>
<td>0.006</td>
<td>0.096</td>
<td>0.046</td>
<td>0.049</td>
<td>0.046</td>
<td>0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
<td>-0.045</td>
<td>-0.064</td>
</tr>
<tr>
<td>0.016</td>
<td>0.041</td>
<td>0.040</td>
<td>0.009</td>
<td>-0.090</td>
<td>-0.643</td>
<td>-0.445</td>
<td>-0.507</td>
<td>-0.007</td>
<td>-0.035</td>
<td>0.055</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.055</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
<td>0.046</td>
<td>0.056</td>
</tr>
</tbody>
</table>