View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Saskatchewan's Research Archive

IMPROVING EXPERTISE-SENSITIVE HELP SYSTEMS

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Mangalagouri Masarakal

©Copyright Mangalagouri Masarakal, January, 2010. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying, publication, or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science

176 Thorvaldson Building

110 Science Place

University of Saskatchewan

Saskatoon, SK, S7N 5C9

Canada

ABSTRACT

Given the complexity and functionality of today's software, task-specific, system-suggested help could be beneficial for users. Although system-suggested help assists users in completing their tasks quickly, user response to unsolicited advice from their applications has been lukewarm. One such problem is lack of knowledge of system-suggested help about the user's expertise with the task they are currently doing. This thesis examines the possibility of improving system-suggested help by adding knowledge about user expertise into the help system and eventually designing an expertise-sensitive help system. An expertise-sensitive help system would detect user expertise dynamically and regularly so that systems could recommend help overtly to novices, subtly to average and poor users, and not at all to experts.

This thesis makes several advances in this area through a series of four experiments. In the first experiment, we show that users respond differently to help interruptions depending on their expertise with a task. Having established that user response to helpful interruptions varies with expertise level, in the second experiment we create a four-level classifier of task expertise with an accuracy of 90%. To present helpful interruptions differently to novice, poor, and average users, we need to design three interrupting notifications that vary in their attentional draw. In experiment three, we investigate a number of options and choose three icons. Finally, in experiment four, we integrate the expertise model and three interrupting notifications into an expertise-sensitive system-suggested help program, and investigate the user response. Together, these four experiments show that users value helpful interruptions when their expertise with a task is low, and that an expertise-sensitive help system that presents helpful interruptions with attentional draw that matches user expertise is effective and valuable.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. Regan Mandryk for her guidance and input throughout my graduate career at the University of Saskatchewan and also I wish to thank her for introducing me to the world of HCI research. A big thank to the faculty and staff at the Department of Computer Science. Additionally, I would like to thank Brian de Alwis, Michael Lippold, Andre Doucette, David Flatla, and Aaron Genest for their help. Finally, thanks to the other students and staff in the HCI Lab for their help and their friendship.

DEDICATION

This thesis is dedicated to my husband Dr. Kiran Doranalli for supporting my decision to go back to school, and being an all round wonderful person.

Finally, I would like to thank my parents for providing inspiration in this and every part of my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MS Word – Microsoft Word

ARFF – Attribute – Relation File Format

CHAPTER 1

INTRODUCTION

Over the past four decades, computer interfaces have been comprised of basic visual interface elements like windows, menus, icons, and pointers. Present computer interfaces have improved functionality, which is accompanied by enhancements to the graphical computer interface. In addition to improving functionality, interface changes are also made to improve efficiency and to make the interface more attractive. Although improving functionality, these interface changes can also result in systems that are confusing and difficult to use. As today's software is increasingly complex, users may need help in order to complete unfamiliar tasks.

When a system is new to a user or its operation is confusing or difficult to master, instructors may help new users by monitoring their interactions and providing assistance when needed. This helps beginners increase their knowledge of a system or application. Similarly, new users of any computer application can gain knowledge easily if assistance is given [60]. However, it is difficult to monitor and assist users of computer applications as much of the learning of new applications is done outside of formal learning environments. When computer users try to complete a new and unfamiliar task without explicit assistance, they normally do so by mapping it to a previously solved problem [4], by asking others for help, or by using the built-in help system.

Built-in help systems are available in most software applications and provide solutions to the users' questions; however, it has been found that built-in help systems are the least-used commands compared to other commands [40]. Studies by Moreno et al. [46] and Atkinson [5] have shown that help agents within an application can help beginners perform tasks more quickly than in applications without help agents. Help agents serve to remind users about help systems.

Take for example the help agent provided in previous versions of Microsoft Word (2000): when the built-in help system in MS Word suffered from lack of use, a help agent called "Clippy" was included in MS Word that reminded users about the help system and provided assistance with common tasks. Not all users liked Cliippy though because not all users need help agents – some users have knowledge about the current task and know the steps to complete the task. These users are called experts. When the help agent interrupts an expert user, the user can become frustrated. Thus, while help agents can benefit non-expert users, they should be avoided for expert users who do not need the extra assistance.

Although a number of avenues could be explored for helping users in completing their task, knowing whether a user needs help or not is an important question. Also, it is important to know whether user expertise changes from one application to another and within a set of tasks in the same application. This thesis carries out research to answer some of these basic questions.

In summary, before providing help suggestions to users, users' perceptions about help suggestions based on their expertise levels must be explored and expertise-sensitive help system must be built and tested.

1.1 Research Problem

The problem to be addressed in this thesis is that system-suggested help systems do not take into account the expertise level of the user, and thus have not been well-received by users with higher expertise levels.

In this thesis we define expertise level of a user as the knowledge of a user about a task or the ability of a user to complete a task. Most software applications have built-in help functions, but users may choose not to access system help, may not want to navigate through help menus, may want to try to figure out how to complete their task in the interface itself, or may forget that help is available to them. System-suggested help solves many of these issues because it reminds users of the help function, removes the complexity of navigating help dialogs, and allows users to receive instructions within the interface itself; however, user response to system-suggested help has been unfavorable.

The main reason that system-suggested help is viewed negatively is that in many cases the help provided is poorly matched to the user's level of expertise. While a novice user may feel grateful for a helpful interruption, an expert user may be frustrated by the same notification. The Lumiere system [43] is an example of a successful intelligent task model paired with a less successful help agent. The agent- commonly known as 'Clippy'- was appreciated by novices and disliked by expert computer users [13].

Motivation

The main motivation for this work is to overcome the limitations of system-suggested help systems by identifying the problems associated with them. There are many benefits of system-suggested help, especially for novice users. The main problem associated with system-suggested help [13] is that help suggestions are given to all users of the application without understanding their knowledge of the current task. Expert users do not like help suggestions, while novice users like them [13]. A help system that differentiates between the expertise of the user, and suggests help only when the user is completing an unfamiliar task will improve the quality of help in computer software by decreasing the frustration experienced by expert users, while providing help to novices. This will increase the productivity of all kinds of users, which ultimately will improve their interactions with computer software applications.

1.2 Solution

If computer systems could detect the expertise of a user for a given task, different help options could be provided. Systems could recommend help overtly to novice users, subtly to average users, and not at all to expert users. However, determining the expertise of a user is not a trivial problem, because the same user may be an expert with one application and a novice with others. Furthermore, the same user may be an expert with a single task in one application and a novice with a different task in that same application. Thus, expertise must be sensed automatically and regularly.

To create an effective expertise-sensitive help-suggestion system, there need to be advances in three key areas. First, we must determine whether users' perceptions of helpful interruptions differ based on their expertise; second, we need to have accurate, generalizable, and dynamic system models of user expertise; and finally, we must create helpful interruptions that vary in their assertiveness for users of varying expertise, rather than relying on an all-or-nothing approach.

Our solution is to build an expertise model that is independent of the task and provide help suggestions that vary in their assertiveness based on users' expertise. In particular, we construct and evaluate a help agent that is sensitive to user expertise level.

To provide this solution we conducted four experiments. Before providing help suggestions based on users' expertise, we determined whether users' perceptions of being interrupted with help suggestions differ based on their expertise levels. Empirical results and subjective rankings of users' expertise level found that users' perceptions of being interrupted with help suggestions vary with their expertise. This result was the motivation for conducting the second experiment to build an expertise user model that classifies user expertise into one of the four levels - expert, average, poor and novice. Empirical results and subjective rankings were used to build an expertise model with a classification accuracy of 90%. Our next step was to conduct an experiment to find three visual signals with different levels of attentional draw and different levels of annoyance. Subjective rankings were used to judge the most appropriate designs of three visual signals. Later, we also evaluated a complete expertise-sensitive help suggesting system using the expertise model adapted to run in real time and the three visual signals that varied in their attentional draw.

1.3 Steps in the Solution

To explore expertise-sensitive help systems, we conducted four experiments. These experiments were designed to systematically understand how users feel about helpful interruptions and to respond to user feedback by designing a help system capable of adapting to user expertise. We summarize the steps in our solution here.

Determine users' perceptions of being interrupted with help suggestions

We conducted an experiment with 12 tasks in Microsoft Word. Data collected from questionnaires were analysed and the results indicated that:

1. User perception of help interruptions varied with their expertise with a given task.

- User frustration with being interrupted increased with task familiarity.
- User frustration with being interrupted increased with task expertise.

2. On tasks where users accept the help option, they were less frustrated with the interruption and agreed less strongly with the statement that they did not need to be interrupted.

3. Users agreed more strongly that they did not need to be interrupted with increasing task expertise.

Build an expertise model

We conducted an experiment with 11 tasks in Microsoft Word. Results indicated that:

- User's expertise varies between tasks and user's expertise level increased with their familiarity with the task.
- If the users repeat the same task more than once, their expertise level with that task increased.
- A model was built for determining the approximate user's level of expertise and the model predicted expertise at a finer granularity than two levels. This model was built from features derived from mouse use, keyboard short cuts, and menu data. This model classifies the user's expertise level as one of the four levels. The four levels are Expert, Average, Poor and Novice.

Investigate three visual signals with different attentional draw

This experiment was conducted to design three visual signals with varied attentional draw, with varied annoyance, with varied ignorability and with varied intrusiveness. Initially a set of visual signals were designed and among them, "Colour", "Bounce Medium" and "Grow Fast" were chosen as a set of signals with different levels of attentional draw. These visual signals were used in our fourth experiment to provide help suggestions to non-expert users.

Evaluate the expertise-sensitive help system

We established subtly-suggestive help through a controlled user study. We deployed the full system in a laboratory trial to gather data on the efficacy and acceptability of expertise-sensitive subtly-suggestive help in real work environments. We found that most users found our help to be a useful tool and that system-suggested help occurred at the appropriate time.

1.4 Evaluation

The goal of this thesis is to assist users in completing their tasks by providing expertise-sensitive help suggestions. This means that help suggestions will be provided overtly to novice users, subtly to average users and not at all to expert users. Our hypothesis was that expertise-sensitive system-suggested help would be liked by users.

The hypothesis was tested in two ways. Before providing help suggestions to users that vary in their assertiveness, we needed to know whether user's insight varies about help suggestions depending upon knowledge about tasks. Participants completed a set of tasks, rated their expertise level and were interrupted with help suggestions for each task. The expertise-sensitive help system was judged to be required if expert users did not like help suggestions and non-expert users liked help systems. Also a four level user model was judged to be valuable if participants varied their self-reported expertise level between multiple levels such as expert, average, poor and novice.

The second evaluation was done on the developed expertise-sensitive subtly-suggested help system. We recorded subjective ratings about the help system. The expertise-sensitive help suggesting system was judged to be helpful if users belonging to four expertise levels appreciated the helpful interruptions that varied in their subtlety.

1.5 Contributions

The major contribution of this research is empirical evidence that an expertise-sensitive help suggesting system is a valuable concept for improving available assistance to the users in interfaces.

In particular, our contributions are:

- A better understanding of users' perceptions of being interrupted with help interruptions in interfaces. Additional evidence that help interruptions are perceived differently by users depending upon their knowledge about a task. Research indicated that experts dislike help interruptions and non-experts feel grateful for the same.

- A four-level classifier of user expertise is valuable.
- A four-level classifier of user expertise requiring no prior knowledge about the user or their task, which was created and validated with real tasks of variable familiarity in a familiar software application.
- A better understanding of users' expertise levels. Users' expertise levels vary from one application to other and also vary from one task to another within an application.
- Additional evidence that user's expertise level increases as he/she repeats the task. Eventually, a time taken to complete the task will be decreased as he/she repeats the task.
- The design of intelligent help interruptions that vary in their assertiveness and their subtlety.
- A better understanding of the limits of expertise-sensitive help systems. This thesis discusses issues that ought to be considered.

1.6 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter Two presents a survey of related research which form the foundation for the research presented in this thesis. First, knowledge acquisition capabilities of users, types of users' expertise levels and differences between experts and non-experts are discussed. Second, adaptive interfaces are discussed. Third, we discuss existing and previous versions of help systems in computer interfaces like built-in help systems and system-suggested help systems. Fourth, we discuss users' perceptions about interruptions and better design approaches for interruptions. Finally, we discuss existing expertise models and other related models.

Chapter Three presents our research with Microsoft Word (2003) and AppMonitor (for logging mouse and keyboard events) for investigating whether users' perceptions about help interruptions vary with their expertise levels. Finding the answer for this question validates a need for building an expertise model that classifies user expertise level into one of the four levels (novice, poor, average and expert) as a solution to the problem of system-suggested help system with an all-or-nothing approach. In Chapter Four we build an expertise model that classifies user expertise into one of the four levels.

Chapter Four presents a research to demonstrate the feasibility of a multi-level expertise model based on real tasks within a familiar application. An informal study is conducted using multiple tasks of variable familiarity in Microsoft Word 2003. The model that classifies user expertise into one of four levels (novice, poor, average and expert) is developed and discussed. Design recommendations for future expertise models are presented.

Chapter Five presents research that investigates three visual signals for providing system-suggested help interruptions. An informal study is conducted using multiple tasks of variable familiarity in Microsoft Word 2003. Three visual signals with varied attentional draw are found.

Chapter Six presents research to investigate the feasibility and efficacy of an expertise-sensitive help-suggestion system. The results of this study are discussed and design recommendations for future expertise-sensitive help systems are presented.

Chapter Seven presents a discussion of the important results from our four experiments. Higher-level implications of our findings and issues related to the work as a whole are addressed. Lessons that have been learned over the course of our work are discussed.

Chapter Eight summarizes the research presented in this thesis, discussing the main contributions of our work and highlighting avenues of future work that have been opened as a result of this thesis.

CHAPTER 2

RELATED WORK

To improve expertise-sensitive help suggestions that vary in their assertiveness according to users' expertise level, five areas of research literature must be examined.

First, to provide expertise-sensitive help suggestions to users with varied expertise levels, it is important to understand users' expertise levels and their knowledge acquisition capabilities. Hence, we examined research that has determined the knowledge acquisition differences between experts and non-experts and skill acquisition features of expert and non-expert users while performing tasks.

Second, we examined related work in the area of intelligent or adaptive user interfaces. Intelligent interfaces are designed to adapt the interfaces intelligently according to user needs. Intelligent adaptation of interfaces can be a way to assist users to accomplish tasks. We examined this research to give insights into the ways which intelligent adaptation can be used to help users and also to know the benefits and drawbacks of intelligent adaptation or intelligent user interfaces.

Third, to work on augmenting system-suggested help systems, we must understand the existing and previous versions of help systems, such as built-in help systems and system-suggested help, in order to find their capabilities and defects. This research aided us in understanding better design rules for an effective help system.

Fourth, to design expertise-sensitive help suggestions that interrupt users to assist them and vary in their assertiveness, we must understand users' perceptions about interruptions. Previous literature about interruptions aided in our understanding of users' perceptions about interruptions, and helped us to adapt a better design approach for help interruptions.

Finally, we examined other predictive expertise models which gave us insight into the ways we can build predictive models to determine user expertise level to provide help suggestions.

2.1 Differentiating Experts' and Non-experts' Behaviour

Identifying expert and non-expert computer users is a main objective of this project because we wish to provide help differently to users of different levels of expertise. Therefore, in the following subsections, we explain about related work that differentiates expert and non-expert users. Differentiation is accomplished through the manner of users' knowledge acquisition. By identifying steps of knowledge acquisition and the accompanying features that can aid in differentiating expert and non-expert users' behaviour, we can find features of expert and non-expert users in order to build an expertise model.

2.1.1 Stages of Skill Acquisition

Normally, users spend some time to master any new or unknown software application. This applies to any new technology or to any real world tool. As a person uses an unknown tool or software application, his expertise level can change. There might be several levels of skill acquisition before he masters that tool or software application. According to Dreyfus [21], skill acquisitions of computer users have five main stages: novice, advanced beginner, competent, proficient and expert [21].

Novice

Most computer applications contain several menu items that are used in different combinations or a particular menu item itself to complete tasks. Using different combination of menu items might help in completing different tasks.

Users who are new to an application or the task they are currently performing are called novice users. Novice users try to identify menu items of an unknown application without any previous experience and try to remember names of menu items for use the next time.

Advanced Beginner

Novice users become advanced beginners when they use their experiences to begin to remember features to complete tasks. As users interact with an application more than once, they begin to remember the steps required to complete the task.

Competent

Advanced beginners become competent users when they make decisions or consciously assess the required steps to complete the task that they are currently doing. Competent users make independent decisions in choosing the required steps to complete the current task. There are chances to fail, and users might feel frustrated when they make errors.

Proficient

Competent users become proficient users when they stop forming the rules (that are a combination of a set of menu items) before actual actions taken are to complete the current task.

Expert

Proficient users become experts with an application when they experience all tasks repeatedly in an application. Expert users respond dynamically and complete any given tasks as they would be familiar with all tasks in an application.

Expertise levels based on time per menu item search

Dreyfus's [21] work on skill acquisition is generalized to computer users, while others focus more on specific applications. Our research focuses on expertise in Microsoft Word.

Norman [51] defined four levels of expertise for users of Microsoft Word based on menu item search time, based on literature [36, 38]. The four types of expertise levels are named as novice, intermediate, expert, and extreme expert [51].

1. ***Novice***: Users in this category perform an exhaustive linear search with an average time per item of 1.0 sec [38, 51].

2. ***Intermediate***: Users in this category examine every menu item until they find the desired menu item. Average time required per menu item in Microsoft Word is 0.5 sec [38, 51].

3. ***Expert***: Users belonging to this category find any menu item with a time equal to a logarithmic function of the number of alternatives. The average time is 0.15 seconds for each menu item in Microsoft Word [36, 51].

4. ***Extreme Expert**:* A user belonging to this category finds any menu item with a same constant time [51].

In Norman's work, the classification of user expertise-level is only based on time required to locate any menu item [51]. However, there may be other features which can contribute to better classifying users' expertise-levels.

2.1.2 Effect of Practice on Expertise Level

Most novice users try to solve a new problem by mapping it to a previously-solved problem [4]. If a user is an expert with any application then it indicates that the user has experience using that application [32]. More experience with an application results in more user memories of using that application [35]. Normally, expert users have greater episodic memory than novice users. Episodic memory is a past memory along with an event related to it [62]. For example, if you see any particular word then you might remember when you first learned it. This helps the expert users to complete the task sooner by easily locating menu items in an application compared to novice users [35]. Also expert users do not need to read all of the menu items' names; they remember the menu items' locations. In contrast, novice users remember menu items' names [29] and must search for their locations. Users remember menu items' locations as they use an application more often [29].

Also, a user's level of expertise with a particular task varies dynamically as the user repeats that task within an application [34]. However, user expertise level also varies from task to task within an application because the user may be familiar with only a particular set of functions of that application [44].

As a user performs the same task repeatedly, it is likely that time required to complete the task will decrease, while the performance rate increases [34, 41]. However, according to the power law of learning [34], as a user repeats the task, the time required to complete a task (T) decreases until a certain point and then there would be no decrease in task completion time. According to the power law: $T = a + bN^{c}$,

Where, T- The time required to complete the task;

a- The limit on performance;

b- The difference between the initial and the limit on the performance;

N- The number of practices;

c- The learning rate.

2.1.3 Features to distinguish experts and non-experts

Until this point, we established that user expertise-level varies from one application to another and from one task to another within an application. User expertise depends upon one's experience with an application. Now, we examine related work that describes features that aid in differentiating the behaviour of experts and non-experts. We describe qualitative and quantitative differences between experts and non-experts [24].

Qualitative Differences

A user familiar with an application environment would be mentally prepared about the actions to be performed. On the contrary, a user who is new to an application would be new to its environment and need to prepare or need to respond dynamically [24, 51]. Users might differ in their planning strategies before reacting to their goal [48, 17]. A user familiar with an application might need to recall the menu item location but a user who is not familiar with this same application might try to transfer his knowledge from previously-solved problems [4, 24, and 29]. Thus, the experience of a user, their knowledge with other applications, and their idea or familiarity with the current application task contributes to user expertise-level.

Users differ in their verbal as well as graphical ability [58]. Verbal ability refers to the ability to understand names of menu items and graphical ability refers to the ability to understand icons of menu items. However, if the interface could provide verbal or graphical information based on their cognitive ability then users could understand the information easily [58].

Quantitative Differences

Quantitative differences can aid us in differentiating experts and non-experts when expertise models need to be independent of the task and knowledge about the tasks. Features can be derived from user emotional states but it is difficult to find the correlation between emotional states and actions in the interface [59]. Features could also be extracted from keyboard or mouse events along with the ways the user presses the keys (i.e hardness) and clicks mouse buttons (i.e. quickness) [23]. Features could also contain usage of keyboard short-cuts (like ctrl + C) to copy the required text or image and help option usage which might indicate that the user is non-expert. Other potential features include deleting typed characters and moving between two applications. Taking pauses in between while completing a task could also aid in differentiating experts and non-experts [23].

Hurst et al. [24] refer to quantitative features to differentiate experts and non-experts in a specific application called GIMP (Image Manipulation Program); however, they did not consider either Fitts' law or the Steering law for finding features. Fitts' law states that time required to reach any target is the function of the distance to the target and the size of the target [63].

The Steering law predicts time required to navigate through a two dimensional tunnel [64]. Hurst et al. [24] did not consider these two laws to derive features because the sizes of menu items are not fixed. However, an expertise predictive model was built with other quantitative features (and is explained in section 2.5.1). Quantitative features considered to build an expertise model include following [24]:

1. *Features from low-level gestures*

Features derived from low-level gestures include speed in pixels per second and time spent on dwelling in seconds.

2. *Features from interaction techniques*

Interaction features include the number of opened submenus, the count of unique menu items visited, and the time spent dwelling with each selected menu item.

3. *Features from performance models*

The time taken to make selection for any menu item per the depth of that menu item was considered. Here, the performance model used was KLM (Keystroke-Level Model) that predicts the time required for an experienced user to complete an operation [65]. If a sequence of operations is given then the result would be sum of time required to perform each operation in the sequence [65]. This KLM model was used to find the difference between the KLM predicted time and the time taken to perform the action.

2.2 Adaptive Interfaces

After examining past research to find out features which might aid in differentiating the behaviour of computer experts and non-experts, our next step is to examine past work done to assist computer users in overcoming the difficulties of human-computer interaction. We can consider assistance in two main ways. The first is to provide interface assistants to help to complete a task, and the second is to adapt interfaces to user needs. Understanding these ways might aid us in better understanding the advantages and disadvantages of adaptive interfaces and help us to provide better assistance to users. In this subsection, we explain past work done in the area of adaptive interfaces.

Human-computer interfaces can be made to adapt in two ways [48]; first, adaptable interfaces allow the user to modify the interface features according to her need. Alternatively, adaptive interfaces predict user needs and modify the available features according to her need. The latter type of adaptive interfaces should consider the current user expertise level and the current user goal before adapting the interface feature list [48].

2.2.1 Adaptable Interfaces

If users are provided with an option to customize their interfaces, users rarely do [43] because of lack of time and the feeling that time spent in customizing the interface is time spent not working. Less experience with an application might not explain why users do not customize because users who have experience with an application know only a few commands which they used often [49]. So, users often learn new things in any application including customizing when they are in need.

Macky [43] collected a set of reasons that trigger users to customize the interface and barriers preventing users from customizing. The list of triggers that discouraged users from customizing include: 33% of users found it too difficult to customize, 63% of users did not have time to customize, 12% were not interested, and 6% were scared to risk customizing the interface. The list of triggers that encouraged users to customize the interface include: 43% of users customized the interface when they began to use same repeated set of features, 22% of users customized when they thought of a new feature, 25% of users customized when they wanted to switch the environment, 18% of users customized out of curiosity and 29% customized when something went wrong.

2.2.2 Adaptive Interfaces

As opposed to adaptable interfaces- where users can customize their interface – adaptive interfaces are those where the interface changes based on a system model of what the user wants, needs, has used recently, or uses frequently.

A study [13] was done to find the effects of the predictability and accuracy of adaptive interfaces on task performance and satisfaction of user. Here, the term “predictability” refers to finding what is going on in the user’s mind or to find out about the strategy used by the user to complete the task, and the term “accuracy” refers to the percentage of time that the required user elements are present in the adaptive area [13]. The authors conducted a study with a modified version of Microsoft Word. An adaptive toolbar was used with two conditions. In the unpredictable condition, the adaptive toolbar contained randomly chosen elements and in the predictable condition, the adaptive toolbar contained the toolbar elements that were predicted to be used next. From the results, increased accuracy has greater beneficial effects on users than increased predictability. However, adaptive interfaces suffer from a lack of clearness and predictability [19] [28].

Although adaptive interfaces were designed to assist users, they also suffer from disadvantages. If users are allowed to see only the required features chosen by the system, they might not seek to learn other possible features or menu items. If the system interface changes frequently, the user may not develop a clear picture of the system and this might reduce performance and self-confidence with the system, which subsequently reduces user efficiency [18]. Users often remember frequently-used menu items by location but, in the adaptive interfaces, menu item location may often change according to the predicted user goal or for some other reason. This disturbs the user’s mental model of the application [22], [14]. Here mental model refers to how a user understands an application and its functionality.

In addition, adaptive interface implementation is not cost-effective compared to normal user interfaces and might require more computational time [48]. Also, adaptive interfaces should be designed in such a way that they should not have power over users, and users should be allowed to turn off the adaptive feature of the interface [56].

2.2.3 Adaptive and Adaptable Interfaces

To overcome these disadvantages, a combination of adaptive and adaptable interfaces was introduced [10], which could be beneficial if users are not able to customise their interfaces efficiently. In this case, a user is provided with two options. The first option allows a user to customize her interface. In the second option, the interface is customized according to a user's needs by the system; however, for this system to customize it needs to know the user's goals.

Adaptive and adaptable interfaces both have advantages and disadvantages, but they can replace or improve help systems if they are designed appropriately by considering users' expertise levels. Detecting the user's goal along with their expertise level is an important step before adapting the interface because users' approaches to complete a required task might be different and users' expertise levels change as they use it. The latter problem can be solved by building an accurate user expertise model.

2.3 Help Systems

After examining the past work done about adaptive interfaces, we found out that adaptive interfaces are good in assisting users; however they are effective only if they are able to detect the users' goal which is difficult for the system to know. Next, we examine past work in help systems, done to assist computer users to overcome the difficulties of human-computer interaction.

As today's software is sufficiently complex, users need help in order to complete unfamiliar tasks. In many learning environments, instructors help beginners of a system by monitoring them, and providing feedback or hints with an explanation. This helps the beginners to improve their knowledge about a system. Similarly, beginners of a software application need instructors to learn new things [60]. Other than adapting the interface, another way to assist users is to provide a help system. In the next sub-sections we examine work regarding the types of help systems, the problems associated with them and the design guidelines for them.

Other than adapting interfaces, users can be assisted by providing:

1. Visual clues [55].
2. A built-in help system [12, 31, 40].
3. A system suggested help system [5].

2.3.1 Assisting users by providing visual clues

Today's technology is growing so fast that users need to quickly learn skills to adapt to new software. Interface design can help in this regard by providing visual clues like familiar icon shapes, icon styles or toolbars. Providing familiar visual clues that are similar to the previous version of an application or similar to commonly-used icons help users transfer skills from previously-used applications [55]. This helps users be more confident when performing a new task in an application. For example, Microsoft Word and Microsoft Power Point contain quite similar visual icons for menu items [55] and this makes users feel more confident while performing tasks in Microsoft Power Point after having learned Microsoft Word.

2.3.2 Assisting users by providing built-in help systems

Providing visual clues might work well for users who are already familiar with similar applications, but other users without that familiarity should be provided with help to complete their tasks. Many applications provide help through access to static or online manuals, through tutorials, or through a built-in help function. The question remains whether people use these helpful tools.

Researchers have collected data for command usage by type in Microsoft Word 1997 [40]. They found out that the "File" command was the highest-used command (48%) and built-in "help" command was the least-used command (0.9%). Many applications have built-in help systems but they suffer from the following problems:

- Users may not be aware of the existing help;

- Users may forget that help is available to them;
- Users may not want to navigate through help menu;
- Users may want to try to figure out in the interface itself;
- Solutions provided by the built-in help system may be difficult to understand as it usually contains a list of solutions or users may not be able to find the menu items which are specified in the help solution [12, 31].

System-suggested help has been introduced to overcome these problems of a built-in help system.

2.3.3 Assisting users by providing system suggested help

System-suggested help reminds users about the help option by popping up a small window or icon. The frequency of help suggestions might depend upon a particular strategy or time and might be independent of a user's knowledge of an application. Two studies by Moreno.et.al. [46] and Atkinson [5], have explained that help agents within an application can help beginners to perform tasks faster than applications without help agents. However, there are issues with help agents too. For example "Clippy", a help agent in Microsoft Office 97, was perceived by users to not be very helpful, rather it was perceived to be annoying and distracting [42, 11]. Novice users may have enjoyed the help provided by "Clippy"; however, "Clippy" did not take into account user expertise and provided tips which were distracting to expert users.

There is a need for more effective pedagogical agents. Pedagogical agents are also a kind of help systems which promote the learning process without distracting the expert users. Intelligent pedagogical agents should not interrupt users at inappropriate times [50]. Research has shown that pedagogical agents can increase student motivation and attention [39] and suggest that considering keyboard input, mouse positioning and mouse clicks while designing agents could help design better help agents.

2.3.4 Designing Effective Help Systems

Although help systems –whether built-in or system-suggested – are beneficial to users, we have shown that they also have some drawbacks. Ellison proposes guidelines to improve help systems [61].

First, he states that using pictures or graphics in the help system solutions can make users better understand the solution. Second, he proposes that providing help by considering possible questions and that considering varied users' knowledge will help to cover all possible questions. Finally, he states that we should avoid extra content in the solution, which is not needed to be known by users, and might cause solutions to be overwhelming.

In summary, system-suggested help would help to overcome some of the disadvantages of built-in help systems; however, system-suggested help is effective when it suggests help without interrupting users or being annoying to users. This can be done by identifying possible moments which are appropriate to provide suggestions or by recognising users' expertise with their current task.

2.4 Interruptions

After examining the related work done on differentiating user expertise, we examined related work on adaptive interfaces and help systems. Now, we are continuing to investigate interface assistants. We are interested in using interface assistants to help users rather than adaptive interfaces because adaptive interfaces do not allow users to explore the interface, which we have shown is important for skill acquisition. However, we found that a major drawback of system-suggested help was the annoyance and distraction of being interrupted. We chose to provide help suggestions that vary in their assertiveness depending upon users' expertise levels. Our next step is to examine related work on interruptions in human computer interaction. This is done because, before providing the help interruptions, we must analyse the cost of redundant help interruptions, the benefits of help interruptions [23], and the design guidelines for efficient help interruptions. We will examine research done for finding the effect of interruptions on task performance, for finding the reasons for disruptive effects of the interruptions, and finally design guidelines for improving interruptions in human-computer interaction.

Although it is true that the nature of people vary and their experience with interruptions also varies, if interruptions are designed and popped up in an appropriate moment [1] and varied in their attentional draw depending upon the importance of an interruption message [16], frustration with interruptions might reduce. Here the attentional draw refers to “the amount of attention attracted by an interruption's notification [16]”. In the following subsections, we examine related work on interruptions in human-computer interaction and design guidelines for providing effective interruptions in human-computer interaction.

2.4.1 Disruptive Effects of Interruptions in Human-Computer Interaction

Although interruptions in human-computer interaction have been used to inform users regarding various important information, users feel distracted, interrupted and annoyed if interruptions appear at an inappropriate time. The time needed to complete a task with interruptions is greater than the time needed to complete the same task without an interruption [7], even when the time taken by the interruption itself is not included.

The disruptive effect of interruptions depends on the concentration required for completing the primary task [7]. Kreifeldt and McCarthy [33] found the disruptive effect of an interruption increases with the similarity of the current task and found the disruptive effect of interruptions is independent of memory load at the time of an interruption. Here memory load refers to memory required to restart or continue the previously left work after an interruption. In contrast, Gillie et al. [15] found that subjects did not have disruptive effects in both high and low memory loads. Some of the interruptions might be disruptive even if they are shorter and the similarity of the interruption to the main task does not play a crucial role in determining the disruptive effect of an interruption.

As a summary, the nature of the interruption (i.e. disruptive rate of an interruption) plays a crucial role in determining the disruptive effect of an interruption [15]. Disruptive rate of an interruption might be because of its length, screen location, and attentional draw.

2.4.2 Improving Interruptions in Human-Computer Interaction

Past research related to interruptions has shown the negative or disruptive effects of interruptions [6, 25]. Disruptive effects of interruptions might be caused by the resumption lag required to restart the previous task after an interruption task is finished or ignored. Resumption lag after the interrupted task is needed to recollect the thoughts in memory regarding the previous task. External cues could be provided just before the interruption (to indicate the awaiting interruption) to reduce the disruptive effect [3]. An example of an awaiting interruption is a ringing phone that indicates an upcoming interruption of a phone conversation. Providing this hint could reduce the resumption lag. But, Miller`s [45] work suggests that interruptions with external cues decreased the task performance with complex tasks.

A study [1] was conducted to measure the effects of interruptions on task performance and on emotional states of users. Results recommended the design of an attention manager system that identifies appropriate interruption times to reduce the negative effects of an interruption.

A guideline suggests that positive perceptions of interruption increase if the interruption's attentional draw is varied according to the importance of its content [52]. This was examined by establishing three visual signals with different levels of noticeability, and results showed that participants found greater benefits with context-sensitive interruptions with varying attentional draw compared to interruptions with a static level of attentional draw [16]. Also it was shown that context-sensitive interruptions with varying attentional draw decreased the annoyance by increasing the positive perception of context sensitive interruptions [16, 52].

In summary, after examining related work about interruptions in human-computer interaction, we find that if interruptions are made to vary in their attentional draw along with the importance of information they provide, then users might feel less distracted and annoyed compared to interruptions with a static level of attention draw.

2.5 Expertise Models

We have seen that expert and non-expert behaviour is different and we found that help systems are better in many ways than adapting the interface; however, providing help suggestions which are less annoying and less distractive to non-expert users could be beneficial.

Our next step is to examine expertise models that are mainly built to identify experts and non-experts in any application. Past work done in the area of building expertise models must be examined because it could aid us in finding the defects in existing expertise models and avoiding those defects in our expertise model, as our goal is to improve expertise models to provide assistance to users in the user interface.

User modeling is needed either to provide intelligent help suggestions or to adapt the interfaces according to user need. One possible way to assist users [48] is to predict user plans to adapt interfaces according to the user's goal or to provide help suggestions according to their goal.

Expertise models can be built either by identifying features to differentiate experts and non-experts [24, 27] or by observing user behaviour and pooling the knowledge from which real-time comparison would be done to provide help [40].

2.5.1 Expertise Models Based on Features to Differentiate Experts and Non-Experts

Expertise models can be built using features that can aid in differentiating experts and non-experts. A three-level user expertise model for a speech-based e-mail system was built with a feature set to distinguish between experts and non-experts [27]. The feature set included occurrence of timeouts, and occurrences of help requests among other features.

Here, the expertise level of each user was updated by considering the user's way of interacting with the system. This might help users by maintaining a database about each user's expertise level. However, this might result in the system needing to maintain many databases as the number of users increases.

Another user model [24] was built for providing intelligent assistance to users by adapting the user interface based on their expertise level. User expertise level was classified either as expert or novice based on mouse acceleration, dwell time or features related to the interaction technique such as number of opened submenus and menu item visits. In particular, the features used were quantitative. A user study was conducted to collect data related to these features. Data collected for each task were labelled with the user expertise level for that particular task. Each task was repeated seven times. The seventh instance of the task was labelled as "Expert" whereas the first instance was labelled as "Novice". The instances between 1 and 7 were not labelled, as their expertise levels were between expert and novice. Each task was considered as an instance of the collected dataset and a decision tree was built to classify expert or novice performance with an accuracy of 91%, and was independent of a task model. As the authors did not consider user level of expertise in between expert and novice, there could be instances where the user may not have knowledge about the task as an expert user does, but more knowledge than a novice user. Thus there are chances of classifying an average user as novice or classifying an advanced beginner as expert. This indicates the need for determining expertise levels between expert and novice.

2.5.2 Other ways of building Expertise Models

Another way of building an expertise model is to observe the individual's behaviour in a usual environment over a long period of time and build an expert model where pooling the knowledge of numerous individuals is done [40]. This would be helpful if the comparison of the knowledge of individual to the pooled knowledge of her peers is done in order to provide help.

Finally, work done by Ramachandran [53] refers to an application called "SmartAidè" which provides help suggestions by observing the preused menu items (or used features of an application) and by guessing the goal of the user. SmartAidè can identify the application being used and generates separate help suggestions based on it. Whenever the user requests help, intelligent help suggestions would be given without asking the current task name. However, help suggestions were provided only when the user requested help. As we previously mentioned, there are disadvantages associated with help systems that will activate only when the user requests help. For example, users may not remember the available help system, or choose not to activate it for various reasons.

In summary, we examined related work for building expertise models and found that there are different ways to build expertise models. One way to build an expertise model is to pool user knowledge and update it as they use the application. This method needs to maintain a database for each user. Another way to assist users is to develop an application that keeps track of commands used and provides help if a user clicks on it. Another way to provide help suggestions is by building an expertise user model with features related to the interaction technique such as number of opened submenus, menu item visits, dwell time between clicks, and mouse acceleration. We found this way of building the expertise model useful because we wanted to build an expertise model that detects user expertise levels, interrupts users with help only if they are not detected as experts, and varies the attentional draw of the help notification with their level of expertise.

CHAPTER 3

EXPERIMENT ONE: USER PERCEPTION OF HELP INTERRUPTIONS

In this chapter we explain the first experiment we conducted to determine users' perceptions about being interrupted with help suggestions. While the majority of this thesis is focused on augmenting system-suggested help systems, such research is only relevant if it is possible to know users' perceptions about being interrupted with help suggestions vary with their expertise level. The result of this experiment was the motivation for building the user model to determine user expertise level that is explained in Chapter 4.

3.1 Motivation and Background

Software applications have become part of our daily lives and we use them during most of the day; however, today's software applications are complex and often users may not know how to complete a task or how to complete a task efficiently.

If a user is not familiar with their current task, then the user might need additional time to figure out the steps required to complete the task. Thus the time required to complete a familiar task might be less compared to an unfamiliar task. As the user uses a new application, she tends to learn a set of features required to complete the task she is currently working on. Users' expertise level may vary from one application to another and their expertise level may vary from one task to another within an application.

If help is provided in the right moment by suggesting task completion steps, then the user might feel grateful. If the help interruption is provided during a familiar task then the user might feel frustrated with being interrupted.

System-suggested help has not been a favourable option because in the past, many users found it annoying and interrupting [42, 11]. An example of system-suggested help is "Clippy" that was provided in previous versions of Microsoft Word. This system-suggested help was accepted by novice users but was hated by expert users. This indicates that prior to providing help suggestions, it is required to find out whether users' perceptions of being interrupted with the help suggestions vary with their expertise level. To achieve this, we need to dynamically determine an individual user’s level of expertise. This was done by asking participants to self report their expertise level for each given task.

Our intention was to find out whether expert users differ from novice users in their perceptions of being interrupted with help suggestions, or whether a single user varies her opinion about help interruption depending on her expertise level for a given task.

The ultimate goal of our work is to build an expertise model. Previous work done in this area by Amy Hurst [24] classifies user expertise level as either expert or novice. As there exist other levels of expertise between expert and novice, there is a need to find these expertise levels between expert and novice.

For this experiment, we considered four self-rated expertise levels. They are: expert, average, poor and novice. Four levels of expertise were chosen because we did not want to classify the user as novice if the user is familiar with the task but does not know the exact steps of completing the task. Also, we did not want to classify the user as expert if the user knows a bit about the task. For example, knowledge about a similar application may help to figure out the task completion steps [4]. In this case users may spend some time figuring out the task completion steps but may figure them out sooner than a novice user.

In particular, we have conducted an experiment to determine whether users’ perceptions of help interruptions vary with their expertise levels. Also we wanted to found out whether users’ expertise levels vary with task and whether users with different levels of expertise have different perceptions of help interruptions (i.e. as the user level of expertise increases, users feel frustrated and interrupted with the help suggestions).

We were also interested in knowing whether users who rated their expertise higher on average would differ in their perceptions than users who rated their expertise as lower on average- that is, whether the relationship between the frustration of being interrupted is at the task level or at the user level.

3.2 Experimental Design and Procedure

An experiment was conducted with Microsoft Word 2003, which is a commonly used application. A set of tasks with varying familiarity were chosen and assigned to participants and participants rated their expertise level with each task after completing it. A help suggestion window was made to pop up for each task and users' ratings regarding frustration and interruption felt with help suggestions were collected. In the following subsections, we describe the experimental tasks, participants and the experimental procedure.

3.2.1 Experiment Tasks

For our first experiment, 12 tasks were chosen from Microsoft Word 2003. These tasks were chosen in such a way that they include both frequently-used and rarely-used tasks. Some of the chosen tasks could have been completed using keyboard shortcuts. Table 1 indicates the tasks chosen for the first experiment. These tasks were completed in the same order by each participant.

Table 1.List of tasks chosen in First Experiment

Task Number	Task Name
1	Cut, paste and replace
2	Insert hyperlink to a file
3	Justify all the paragraphs in center alignment
4	Increase the line spacing of the whole document to 1.5
5	Change the background color of first page to green
6	Add line numbers to a whole page
7	Translate the first paragraph into French
8	Insert a picture from a file after the 1st paragraph
9	Insert a comment to any word in the whole document
10	Increase the font size of all characters to 16
11	Change the font of all the characters to "Kartika"
12	Increase the number of columns to 2

A description of each task was given to ensure that the participants understood the tasks but the description did not contain the specific steps required to complete the task. The description of the next task was provided once they finished the previous task. Unlimited time was given to each participant to understand each task before attempting the task. Participants were allowed to use both keyboard shortcuts and menu selections to complete the tasks. Below are descriptions of each task:

Task 1: Cut, paste and replace

Description: Please cut the 2nd paragraph and insert that paragraph at the end of the document. Replace every occurrence of the word "student" with the word "scholar" in the whole document.

Task 2: Insert hyperlink for a file

Description: Please provide the hyperlink for the file. File name is "Ex.txt" and is on the desktop. The hyperlink provided to a file looks like this: Ex.txt

Task 3: Justify all the paragraphs in center alignment

Task 4: Increase the line space of the whole document to 1.5

Description: For example:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented.

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented.

Task 5: Change background colour of Microsoft Word to green.

Description: The background colour of Microsoft Word is by default white. Change the background colour of Microsoft Word to green.

Task 6: Add line numbers to the whole page.

Description: Add line numbers to whole page including the blank lines. Example is as follows:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented. The word "Convocation" arises from the Latin "con" meaning "together" and "vocare" meaning "to call." The University of Saskatchewan's Convocation ceremony is a calling together of new graduates.

During the period between which you have completed the requirements to graduate and you are awarded your degree, you are called a "graduand." The University encourages all graduands to attend the Convocation ceremony. However, if you are not able to attend, you are still eligible to graduate and your parchment will be mailed to you. You must apply to graduate regardless of whether you plan to attend the Convocation ceremony.
All undergraduate and graduate students who expect to graduate at either the spring or Fall Convocation must complete an Application to Graduate form. This form must be submitted by March 31 for Spring Convocation or by August 31 for Fall Convocation.

Students who have submitted an Application to Graduate will receive a Convocation package approximately one month prior to the appropriate Convocation ceremony regardless of whether or not they have met the requirements to graduate. It is each student's responsibility to ensure that they have met the requirements to graduate. College offices can confirm that students have met the requirements for their degrees.

Task 7: Translate the first paragraph into French.

Task 8: Insert a picture (from a file) after the first paragraph.

Description: Insert the picture from the file called convocation.jpg: the file is on the desktop.

Task 9: Insert a comment to any one word in the document and write the comment "This is an example comment".

Description: For example:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented.

Comment [K1]: This is an example comment.

Task 10: Increase the font size of the all letters to 16

Task 11: Change the font of all the letters to “Kartika”

Task 12: Increase the number of columns to two

Description: For example:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented. The word "Convocation" arises from the Latin "con" meaning "together" and "vocare" meaning "to call."

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented. The word “Convocation” arises from the Latin “con” meaning together and “vocare” meaning “to call”.

3.2.2 Participants

Microsoft Word is frequently used by Computer Science students. In order to better represent the general population, participants were recruited from outside of Computer Science. There were 15 participants (9 female, 6 male) with an average age of 26. All participants were right-handed. Participants were paid $5. The experiment took approximately 20 minutes for each participant to complete.

3.2.3 Procedure

Tasks were completed in the same order by each participant. Help windows were designed using Java language. For each task, regardless of the user's expertise, a help dialogue (Figure 1) was popped up after 45 seconds and after 3 minutes. Pilot studies showed that the 45-second delay gave users with high expertise ample time to complete the task, while not causing users who struggled with the task to wait for extensive periods of time. If the given task was completed before 45 seconds then participants were asked to wait until the help dialogue is popped up. If the participants did not make use of first help option or could not figure out how to complete the task even after the first help dialogue, they made use of the second help option.

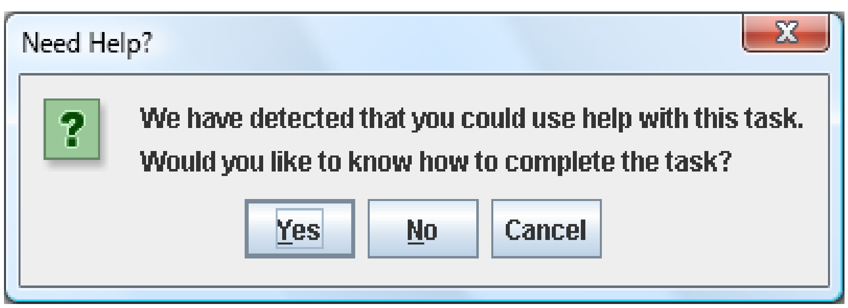

Figure 1: Help dialogue.

With the offer of help "We have detected that you could use help with this task. Would you like to know how to complete the task?" Users could select 'Yes' or 'No' or 'Cancel'.

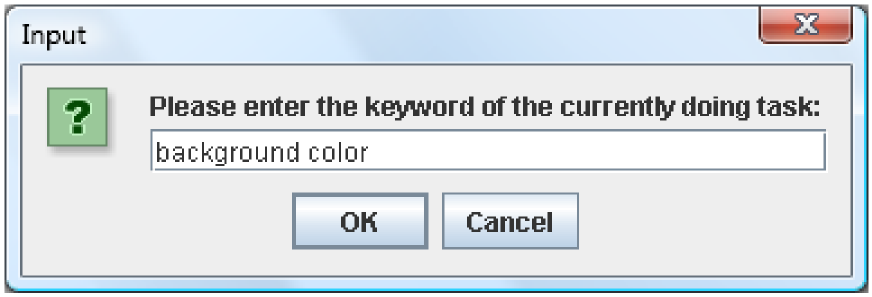

Figure 2: Input dialogue for task information

If users selected ‘Yes’, another dialogue asked them to enter their task information (Figure 2), from which completion instructions were provided (Figure 3). Task information includes the main keywords of the current task name. Users were then directed to a questionnaire form.

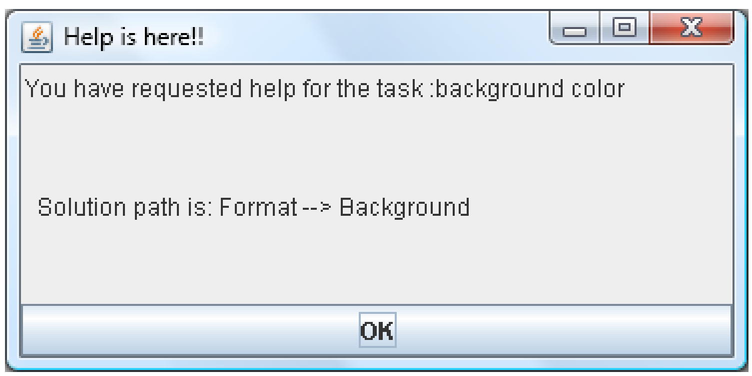

Figure 3: Dialogue box with instructions for a given task information.

Even if users had not completed the tasks and selected ‘No’ from the original offer of help dialogue, they were directed straight to the questionnaire form (Figure 4). The questionnaire was designed to obtain participants’ perceptions on being interrupted for help. Users were asked to rate their agreement with the following statements on a seven-point scale (Strongly Agree, Agree, Slightly Agree, Neutral, Slightly Disagree, Disagree, and Strongly Disagree):

- This task is easy
- I am familiar with this task and have done it before
- I did not need to be interrupted

- I am frustrated by the “help” interruption

In addition, participants rated their expertise with the current task on four levels (expert, average, poor, and novice). This process was repeated for each task. At the end of the experiment, users were given the opportunity to comment on the system, on the experiment, and on the help interruptions. The experiment was conducted on a PC running Windows XP and MS Word 2003. Participants did not receive help from the experimenter if they could not complete the task as help was provided by the system.

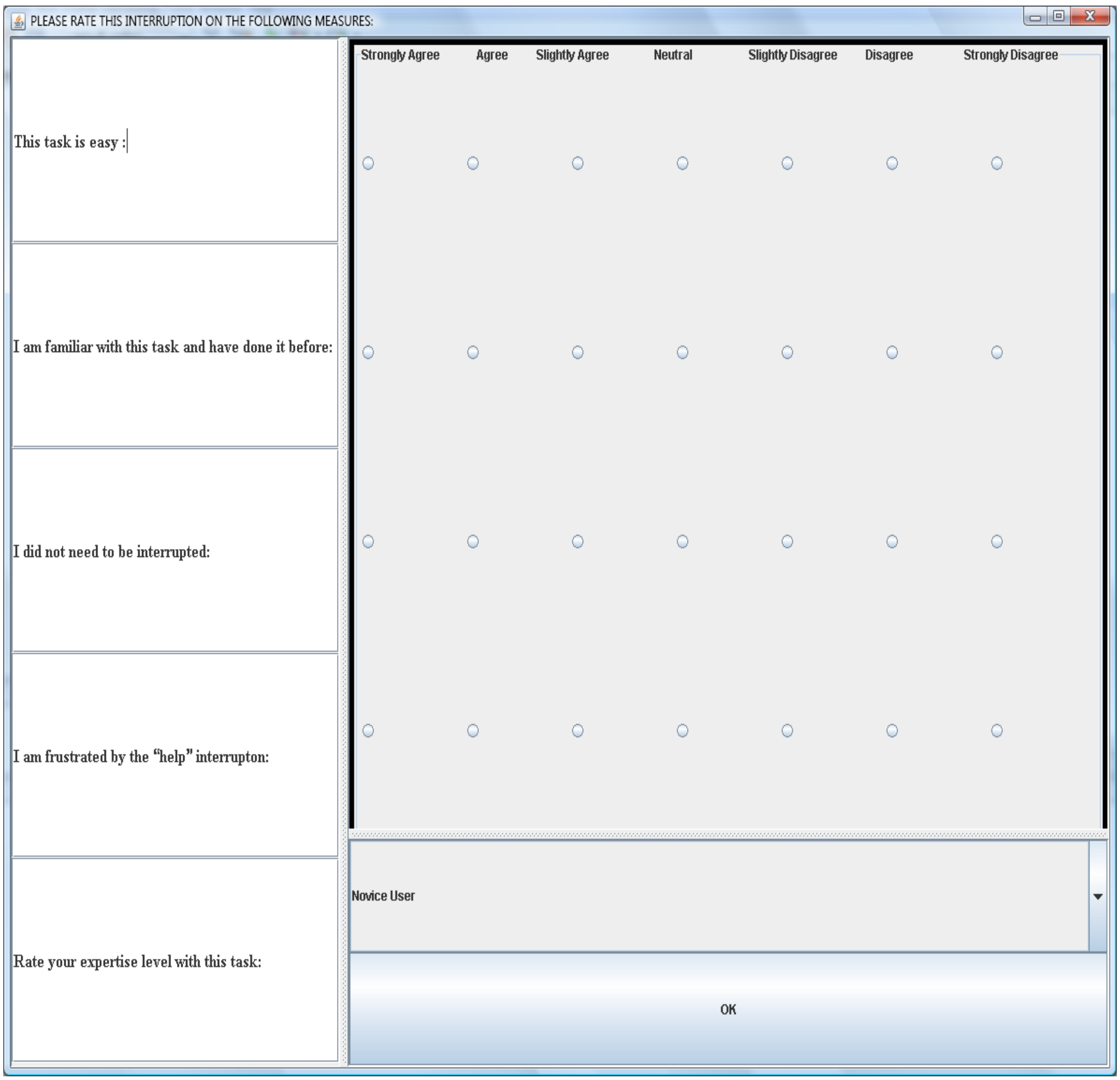

Figure 4: Questionnaire form which popped up following every help window

3.3 Hypotheses

Our assumption was that users' perceptions of help interruptions would vary with their expertise with a given task. As such, we had the following experimental hypotheses:

- On tasks where users accept the help option, they will be less frustrated with the interruption and will agree less strongly with the statement that they did not need to be interrupted.
- Users' frustration with being interrupted will increase with their familiarity with the task.
- Users' frustration with being interrupted will increase with increasing task expertise.
- Users will agree more strongly that they did not need to be interrupted with increasing task expertise or with increasing their familiarity with the steps to complete the task.

3.4 Data Analysis

A Java program logged the users' response to the offered help, and the users' answers to the questionnaires for each task. As the majority of our data was gathered from user ratings, and this data did not conform to the requirements of parametric statistical techniques (e.g., ANOVA), we used non-parametric tests. Correlations were tested using Spearman's rho for non-parametric correlation; differences were tested with Friedman's test for the comparison of multiple-related samples or the Wilcoxon signed-ranks test for two-related samples [25].

3.5 Results

3.5.1 Task Familiarity and Ease

We used tasks of varying familiarity to create variable expertise in the users across the different tasks. As Figure 5 shows, the ratings of familiarity increased with increasing task expertise, and the ratings of ease of the task increased with increasing expertise. These results show that our experimental tasks were of variable familiarity to the users and were creating variable expertise levels in the users over the course of the experiment.

We compared the ratings of frustration with being interrupted and the familiarity ratings of the tasks and found that there was a significant correlation (rho=.28, p<.001). Users' agreement with the statement that they did not need to be interrupted was also correlated with the familiarity ratings of the tasks (rho=.65, p<.001). User comments supported these results; for example, "[the pop-up help] was a little annoying with really easy tasks that are commonly performed, but very useful for unfamiliar tasks"; "the popup helps sometime if you don't know the function of Microsoft Office. However if you know the task of the function, it's a little bit frustrating"; and "but if the task is easy, the pop-up "help" is very frustrating".

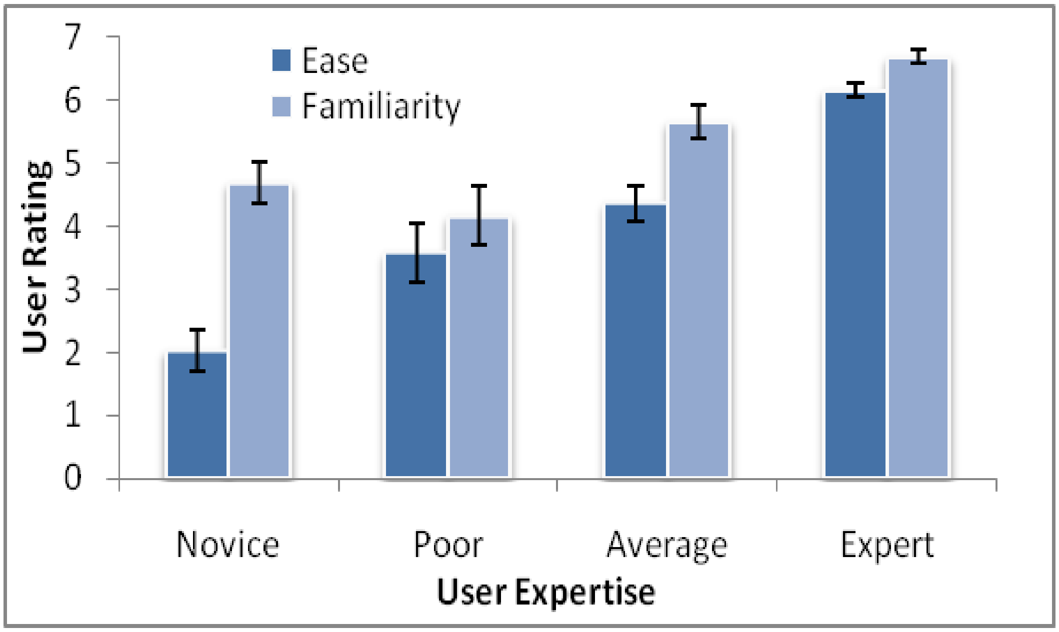

Figure 5: Means (±SD) for agreement of ease and familiarity of task by user expertise on task (higher is more agreement)

3.5.2 Use of and Response to Help Dialogue

Help was always offered to the user after 45 seconds and again after 3 minutes. Users selected help in 29 of 180 trials (16.1%). When users did not know how to complete the task, they were often grateful for the pop-up help option. For example, one user commented that, "for one task I was glad to see the window as I didn't know how to perform the task and I realized that it was helpful."

Table 2: Means and SD of user ratings depending on whether or not help was selected. Expertise was rated on a 4-point scale (higher is better), while the remaining factors were rated on a 7-point scale (higher is more agreement).

	Mean (SD) no help	Mean (SD) help	Z	p
Expertise	3.8 (0.3)	2.1 (0.9)	3.41	.001
Ease	6.6 (0.4)	4.6 (1.8)	3.30	.001
Familiarity	5.8 (0.9)	3.3 (1.8)	3.30	.001
Frustration	4.6 (1.4)	2.8 (1.4)	2.73	.006
Interruption	6.2 (0.9)	3.1 (1.4)	3.35	.001

Users accepted help when the tasks were unfamiliar or hard; during these tasks they rated their expertise lower than in tasks where they did not accept help. Wilcoxon signed ranks tests for two-related samples show that these differences were significant. Table 2 shows the mean differences and statistical test results.

We were interested in whether users' perceptions of interrupting help were different on tasks where they accepted the system help. Wilcoxon signed ranks tests [25] for two-related samples showed that on trials where users accepted help, their frustration with being interrupted was significantly lower, and their agreement that they did not need to be interrupted was significantly lower. Table 2 shows the mean differences and statistical test results.

3.5.3 Effects of Expertise

Our goal was to see how users' perceptions of help interruptions varied with their expertise. Users rated themselves as expert in 71.7% of the trials, as average in 14.4% of the trials, as poor in 4.4% of the trials, and as novice in 9.4% of the trials. Although all ratings of expertise were selected, they were not all selected by every user – thus our data does not enable us to individually test for differences in the ratings of frustration for the different levels of expertise.

We compared the ratings of frustration with the expertise ratings of the tasks and found that there was a significant correlation ($\rho=.50$, $p<.001$). Users' agreement with the statement that they did not need to be interrupted was also correlated with expertise (rho=.72, p<.001). As Figure 6 shows, both of these factors increased with an increase in expertise.

We anticipated that users' perceptions of help interruptions would vary with their expertise on a given task (i.e., a single user would change their perception depending on the task), but we were also interested in whether users with higher overall average expertise would respond differently than users with lower expertise on average. As such, we used k-means clustering to divide the users into two, three and four groups based on their mean expertise ratings across all tasks. We found no significant correlation between frustration and average user expertise after clustering users into two (rho=.28, p=.288), three (rho=.18, p=.521), or four (rho=.31, p=.263) groups based on their expertise.

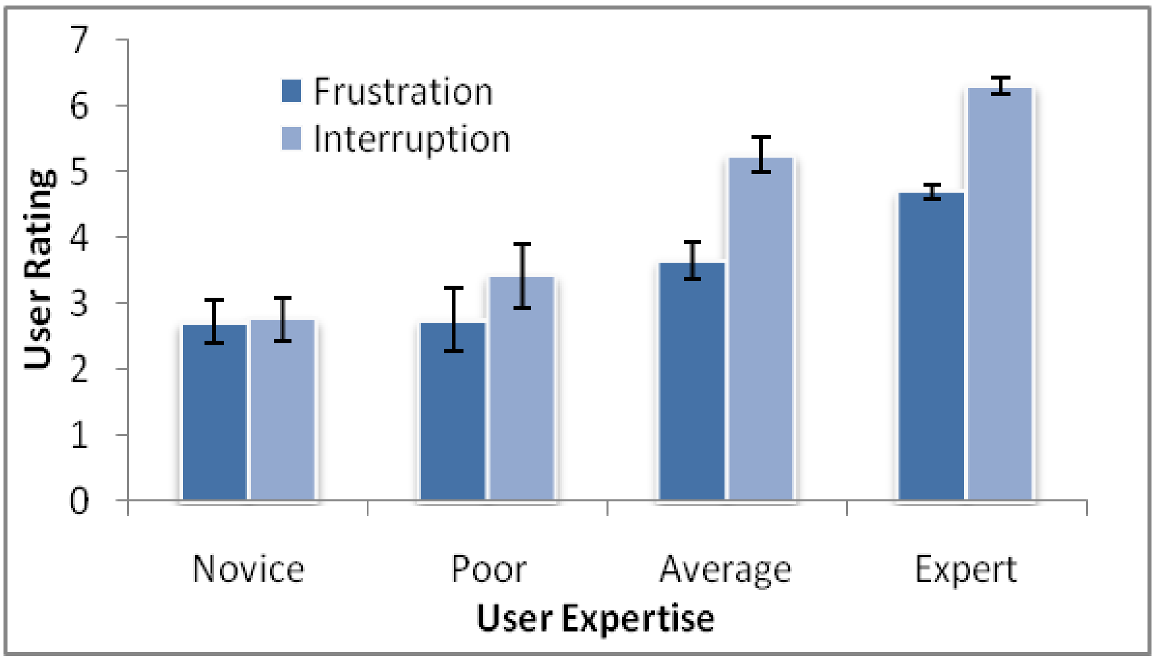

Figure 6: Means (±SD) for agreement of frustration with the interruption and agreement that user did not need to be interrupted by user expertise on task (higher rating is more agreement).

Together, these results show that the relationship between frustration with being interrupted and expertise is on task level, rather than on user level- that is to say that a single user will change their perceptions of being interrupted with help depending on their expertise with a given task, rather than basing their opinion on their expertise in general.

3.6 Summary

We conducted an experiment with 12 tasks in Microsoft Word. Collected data from questionnaires was analysed and the results indicated that our entire hypotheses were met. Results indicated that:

1. Users' perceptions of help interruptions varied with their expertise with a given task.
   - Users' frustration with being interrupted increased with their familiarity of the task.
   - Users' frustration with being interrupted increased with increasing task expertise.
2. On tasks where users accept the help option, they were less frustrated with the interruption and agreed less strongly with the statement that they did not need to be interrupted.
3. Users agreed more strongly that they did not need to be interrupted with increasing task expertise.

3.7 Discussion

In this experiment, we tested users' perceptions of help interruptions. As the familiarity and ease of the tasks increased, users' expertise also increased and they were more frustrated by suggested help and agreed more strongly with the statement that they did not need to be interrupted.

Although both frustration with the interruption and the lack of necessity for the interruption increased with user expertise, users generally rated their level of frustration lower than their lack of need (see Figure 6). These results suggest that users who are interrupted unnecessarily may be somewhat tolerant before becoming frustrated or annoyed.

Our results show that the multi-level approach to defining user expertise is valuable. We show that user response to the help interruptions varied with the four levels of task expertise, suggesting that multiple levels of interrupting help suggestions may be more appropriate than an all-or-nothing approach. Although a two-level approach would benefit users – by suggesting help when they are not expert at their current task – our results clearly show that there is a different user response when they have moderate levels of expertise (poor, average). As such, there is value in exploring interrupting notifications that vary in their assertiveness and subtlety.

Our results suggest that system-suggested help should be provided to users who are not experts with their current task and that this help should be provided at multiple levels of interruption. To integrate our results into a system, we must be able to dynamically detect multiple levels of expertise for a single user in a real application.

In our next experiment, we present an experiment designed to test whether or not we can model users' expertise dynamically at four levels: expert, average, novice and poor.

CHAPTER 4

EXPERIMENT TWO: BUILDING THE EXPERTISE MODEL

4.1 Motivation and Background

Our first experiment showed that non-expert users felt grateful for the help interruptions and expert users felt frustrated for the help interruptions. Also, the frustration with helpful interruptions increased for each increase in expertise level. These results imply that help interruptions should be given by considering the user's level of expertise. In particular, help interruptions should be given overtly to novice users, subtly to poor and average users and not at all to expert users.

In the first experiment, participants used all four expertise levels during self-evaluation, supporting the use of higher levels of granularity for expertise ratings than simple expert/novice categories. Not all participants rated their expertise level as either expert or novice. There were participants who rated their expertise level as either average or poor too. This indicates the need for a finer granularity of expertise levels compared to only two levels of expertise level. To support the goal of providing different help to users based on their

expertise, we need to build a model that predicts expertise at a finer granularity than two levels. Also, in the first experiment, participants' expertise levels were varied from one task to another signifying the need for a model to detect the user's level of expertise dynamically at the level of a user's current task.

The goal of our second experiment is to demonstrate the feasibility of a multi-level expertise model based on real tasks within a familiar application, where a user's expertise level is expected to vary between tasks. As such, we conducted an experiment using multiple tasks of variable familiarity in Microsoft Word 2003. A model is built from features derived from mouse use, keyboard short cuts, menu data, and mouse click times.

4.2 Experimental Setup

4.2.1 Experimental Tasks

The tasks chosen for this experiment were the same as in Experiment One; however, we removed one task (12) and modified the first task to be “cut, copy, and replace”.

Table 3: List of tasks chosen in second experiment

Task Number	Task Name
1	Cut, copy and replace
2	Insert hyperlink for a file
3	Justify all the paragraphs in center alignment
4	Increase the line spacing of the whole document to 1.5
5	Change the background color of first page to green
6	Add the line numbers to whole page
7	Translate the first paragraph into French
8	Insert a picture from a file after the 1st paragraph
9	Insert a comment to any one word in the current document
10	Increase the font size of the all letters to 16
11	Increase the number of columns to 2

Table 3 contains the tasks chosen for the second experiment. These tasks were completed in the same order by each participant. If the users could not finish a task in 3 minutes, the experimenter provided them with help.

Participants were provided with a description of each task, but the descriptions did not contain information on how to complete the task. Descriptions for each task were presented after the completion of the previous task and before the start of a new task. Unlimited time was given for each participant to examine task descriptions before attempting the task. Participants were allowed to use both keyboard shortcuts and menu selections to complete the tasks.

The descriptions of each of the tasks are provided in Section 3.1 as the tasks are similar to those used in Experiment One.

After completing all 11 tasks in order, participants repeated the 11 tasks in the same order an additional two times. This was done to vary participants' levels of expertise with a single task as it became more familiar over the course of the experiment. Before repeating the same task again, participants completed the other tasks in order. After completing each task, participants rated their expertise on a 7-point scale, where "1" indicated "novice" and "7" indicated "expert". The 7-point scale was given in order to build a model which predicts expertise at a finer granularity than two levels and to provide the most freedom in our classification approach.

4.2.2 Participants

Seventeen participants (7 female, 10 male) with an average age of 25, and who had not participated in Experiment One volunteered to participate in this experiment. Most of them were recruited using an online student portal which can be accessed by students and staff at the University of Saskatchewan called "Bulletin board". Participants were paid $10. The entire experiment took approximately 45 minutes to complete.

After completing an informed consent form, participants filled out a background questionnaire, rating their computer expertise, their expertise with Microsoft Word, and their expertise in speaking English on a seven-point scale (higher is better). Their average expertise with computers was 4.53 (range: 3-6), with MS Word was 4.88 (range: 3-6) and in speaking English was 5.47 (range: 4-7).

4.3 Data Collection

The experiment was conducted on a PC running Windows Vista and Microsoft Word 2003, with a standard 2-button plus scroll wheel mouse as the input device. User actions were logged with AppMonitor [2]. AppMonitor makes use of Windows SDK libraries to record both low-level interactions such as keystrokes and high-level logical actions such as menu bar selection. We used AppMonitor because it is able to record a variety of user actions. An example of raw data logged from AppMonitor is shown in Figure 7.

```
25/12/2008 12:30:19.118: WM_LBUTTONDOWN 0x2023e Microsoft_Word_Document/client
25/12/2008 12:30:19.243: WM_LBUTTONUP 0x2023e Microsoft_Word_Document/client
05/01/2009 13:06:20.295: EVENT_SYSTEM_MENUPOPUPEND 0xa0372 View
05/01/2009 13:06:20.301: EVENT_SYSTEM_MENUPOPUPSTART 0xa0372 Insert
05/01/2009 13:06:20.308: WM_MOUSEMOVE 0xa0372 Insert/menu_item{126,36}
05/01/2009 13:06:20.314: WM_MOUSEMOVE 0xa0372 Insert/menu_item{136,41}
05/01/2009 13:06:20.319: WM_MOUSEMOVE 0xa0372 Page_Numbers.../menu_item{147,49}
05/01/2009 13:06:20.324: WM_MOUSEMOVE 0xa0372 Field.../menu_item{201,73}
```

Figure 7: Sample data logged by AppMonitor

4.4 Hypotheses

Our assumption was that users' expertise would vary between tasks and users' expertise levels would increase with their familiarity with the task. As such, we had the following experimental hypotheses:

- As users repeat the same task, their expertise level with that task would increase.
- Users' expertise would vary between tasks.

- A predictive model should effectively classify expertise at a finer granularity than two levels.

4.5 Analysis and Results

Data from AppMonitor was analyzed using custom Java software, which extracts features for use in the model. We also recorded the time taken by the user to complete each task, and the user's self-reported expertise on a scale from 1 to 7. We expected that a user's expertise would vary across the experimental tasks and also across repetitions of the same task.

As shown in Figure 8, the time taken to complete most tasks decreased with multiple repetitions, which suggests that users' task expertise increased with repetition. Wilcoxon signed ranks tests for two-related samples show that this average decrease in time was significant for each repetition of the task ($Z_{1-2}=3.6$, $p<.001$, $Z_{2-3}=2.2$, $p<.03$).

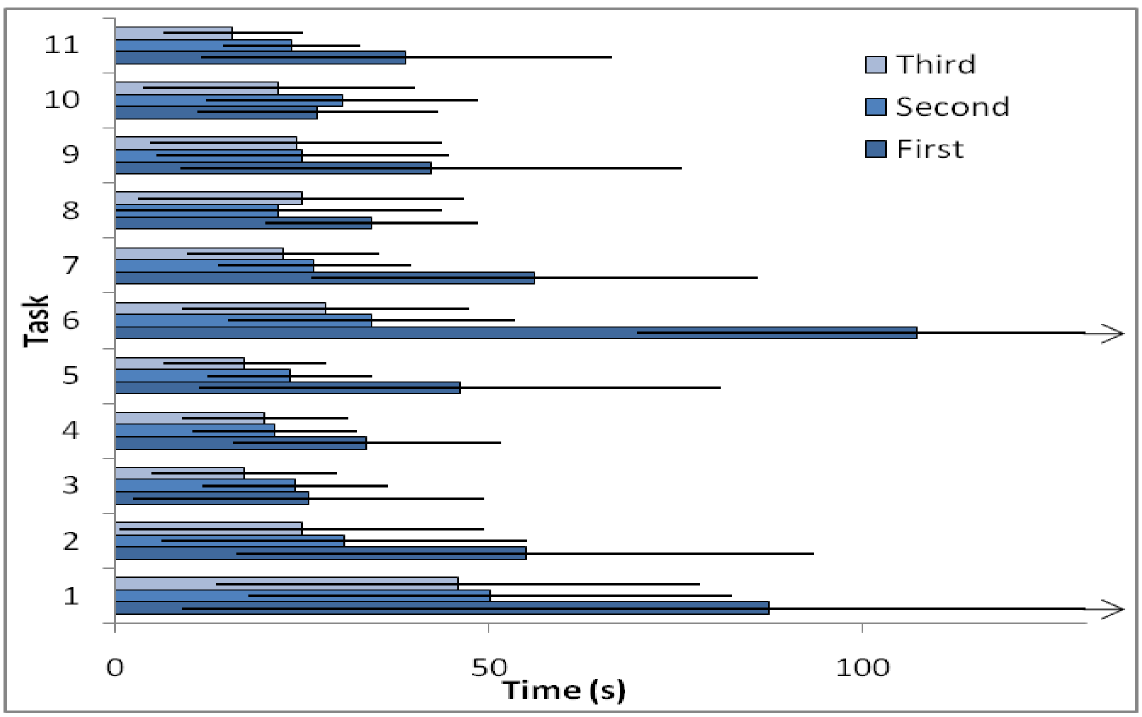

Figure 8: Average time (± SD) to complete each repetition of the experimental tasks.

As shown in Figure 9, users' self-reported expertise increased on average with multiple repetitions of the task. In the case of familiar tasks, large changes in self-reported expertise were not expected. Wilcoxon signed ranks tests for two-related samples show that this increase in expertise was significant for each repetition of the task (Z1-2=3.6, p<.001, Z2-3=2.3, p<.04). Taken together, the timing and expertise data suggest that participants had variable expertise across the tasks and that expertise increased over the course of the experiment on many of the tasks.

Figure 8 and Figure 9 satisfy our first and second hypotheses (i.e. As users repeat the same task, their expertise level with that task will increase and a user's expertise will vary between tasks).

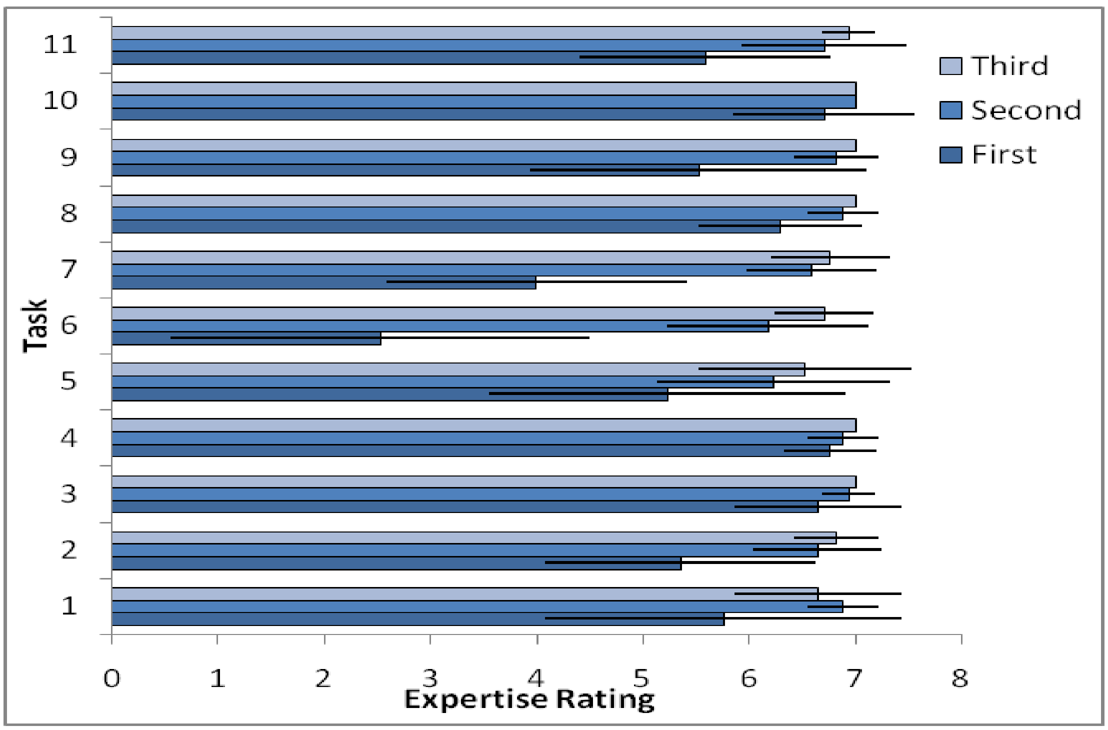

Figure 9: self-reported expertise levels (± SD) across multiple repetitions of the tasks.

4.5.1 Classification of Expertise Levels

Having established that the participants' expertise levels varied across tasks and across multiple repetitions of the same task, we needed to establish the expertise labels for our classifier.

In this experiment, participants rated their expertise level on a seven-point scale. This was done to provide the most freedom in our classification approach; however, we did not feel that it was necessary to classify seven levels of expertise. As such, we needed to cluster adjacent scale points to create fewer expertise levels. Although it was possible to cluster the values in myriad ways, the motivation for the work- to interrupt users with help solutions differently depending on their expertise- guided the clustering process.

Users with below-average expertise would benefit from suggested help more than users with above-average expertise, so the presentation of help should be more differentiated at the low end of predicted expertise, as we expect the desire for helpful interruptions to decrease non-linearly with increasing expertise. Using these principles and the subsequent performance of the classifier, we aggregated self-reported expertise according to Figure 10.

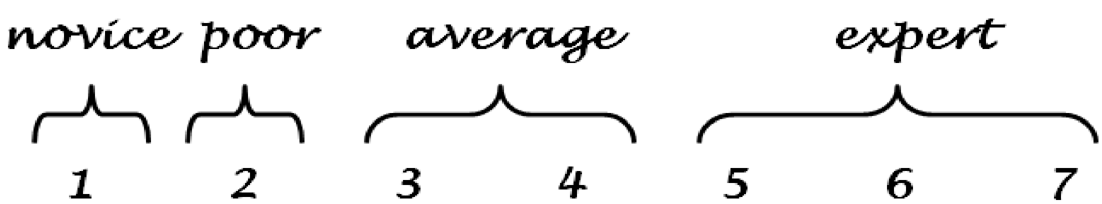

Figure 10: Mapping of the self-reported expertise levels (1-7) into appropriate categories.

4.5.2 Feature Selection

Before conducting the second experiment, we thought of some usage features which should help us to best differentiate the expert and non expert users. During the study, we noticed more interesting features to build a task-independent model which classifies the varying expertise levels of the users. We selected features derived from mouse use, keyboard short cuts, and menu data. We based many of our features on those used in [24], while also adding new ones. An overview of the most relevant features for our model follows:

Dwell time of a single click and dwell time between double click:

Novice computer users are most likely novice mouse users, which could be indicative of their expertise with different tasks. Also, expert computer users may reveal different mouse use characteristics when completing an unfamiliar task. Longer dwell times within a single click or double click gesture may indicate uncertainty of the user in making a decision while selecting items or opening sub-windows. The mean dwell time range of a single click count in this experiment was: 0.01-15.8 second(s). The mean dwell time range of a double click count was: 0.0 – 0.53 second(s).

Menu bar checking:

As described in [24], novice users search for unknown items in menus differently than expert users. Expert users might complete the task by remembering the menu item location or predict the menu item location by comparing with another menu bar of similar software. In addition, users with less expertise may hunt for the right menu items more than users who know what they are looking for and where to find it. As in [24], we used the number of menu items that were visited, but not selected in an interaction. The range of menu bar checking count in this experiment was: 1-25.

Cancel count:

When a user selects the wrong menu item without having sufficient knowledge about its functionality, they need to cancel that operation. This indicates a novice user or lack of expertise with a given task. The range of cancel count for a single task in this experiment was: 2-4.

Help count:

If a user is new to the application or a task, they may select the system help option. Although selecting help would negate the need to suggest help to a user, this feature (or its lack) could be used for predicting expertise.

Special key usage:

Many keyboard shortcuts have the same meaning across multiple applications. Use of keyboard shortcuts like ctrl+x or ctrl+v, reduce the time required to complete many tasks. Beginner users might choose to complete their tasks through the menu bar items for actions such as cut and paste, rather than through keyboard shortcuts. The range of special key usage count in this experiment was: 1-3.

4.6 Classification of the Data

A custom Java program extracted the aforementioned features from the AppMonitor data logs. Data was stored separately for each task at each repetition. Participants had completed the eleven tasks three times in the second experiment to make sure that at the end each participant became familiar or expert with each task. We included data logged from the initial and final completion of each task in building our model. This was done to maximize the variability in user expertise ratings.

DwellTime (real), DwellTimeBetweenDoubleClick (real), Help (numeric), Cancel (numeric), SpecialKeysUsage (numeric), MenubarChecking (numeric), ExpertLevel (nominal)

Figure 11: Instance format used in input file for WEKA

Each first and third repetition of each task for each participant was considered as an instance and these instances were used for constructing the classifier. Each instance contains an average of dwell time between single click and double click along with total count of help, cancel and special key usages. The instance format is shown in Figure 11.

We used the WEKA machine learning toolkit [57] to create and validate the classifier. This software tool contains several machine learning algorithms for solving data mining problems. It is built with tools for data preprocessing, classification, regression, clustering, and association rules. The application contains a simple GUI and uses the ARFF document format for input data.

Although we investigated many learning algorithms, we found that decision trees consistently gave us the highest classification accuracies. In addition, decision trees are easy to understand, easy to convert into rules, and allow multiple outputs, which was particularly important since we wanted to classify the data into four categories (i.e. expert user, average user, poor user and novice user).

Since our identified features are represented in both integer and real numbers, we discretized the data using unsupervised discretization before building the classifier, which can help to increase the accuracy when using numeric data.

Common decision tree algorithms include ID3 and J48 [57]. We used the J48 algorithm implemented in WEKA, which can deal with missing values, numeric values and nominal values. As our data contains missing values and is numeric, we used the J48 algorithm. The J48 algorithm uses a greedy technique to construct a decision tree in a top-down way. The information theoretic measure is used as a classification power of each feature and the data set would be split into two categories [57, 54]. The same process would be repeated until all or most of the instances belong to one category [57, 54].

To test the model, we used 10-fold cross validation, which divides the data into ten parts. Each part is held out in turn; the learning procedure is applied on the remaining 9/10 and the error rate is calculated on the held-out set [57]. This procedure is repeated ten times so that each set will get used exactly once for testing. The ten error rates obtained are averaged to obtain the overall error rate.

4.6.1 Pruning the Decision Tree

Weka provides many options for pruning the decision tree of the J48 algorithm. Pruning of decision trees is done in order to generalize the tree and also to improve the performance of the tree [66]. Most of the time, pruning the decision tree gives different performance or accuracy results compared to the unpruned decision tree. We pruned our decision model in order to improve its performance and to make it more generalizable. Weka provides the following important pruning parameters which can be adjusted:

reducedErrorPruning: If this option is set to true, a portion of the data would be kept as test data and the remaining data would be used as training data. The test data would be used to test the decision model built from the training data. This would obviously reduce the number of data sets in training data. Setting this option to true for limited data sets is not recommended [54, 66]. We have set this option to false as our training data set is limited.

subtreeRaising: If this option is set to true, then a process of moving the leaf node towards the upper node is done (if it is required). This is done until the root node (if it is required) [54, 66]. Error rates are used to decide on raising any part of the decision tree. While building our decision model, this option was to set to true to obtain a more generalized tree.

confidenceFactor: This option is also used for pruning and reducing this figure would give a more pruned decision tree. We have set this value to 0.25 which was a default value and also resulted in better accuracy and performance than other values.

minNumObj: This option decides the minimum number of attributes that can form leaves. We have set this value to 1 after trying multiple values.

binarySplits: If this option is set to true then it confirms the binary splits on nominal attributes while building the trees. We turned off this option as the performance of the tree was better compared to turning it on.

numFolds: This option allows to set the amount of data used for building the model and data for pruning the tree. Here always one fold is used for pruning the decision tree and other parts are used for building the decision tree. We have set this option to 3. Where 1 fold is used for pruning and 2 folds are used for building the model.

unpruned: If this option is set to true then pruning of the decision model would be done else it would not be done. We have set this option to true.

4.7 Results

4.7.1 Training Data

Initially, our model was built using the training data and the classification accuracy obtained for the training set was 92.2%. Run information obtained for the training set is as shown in Table 4 and the decision tree obtained is as shown in Figure 12. Training data contains all the given set of instances. From the obtained run information, the Kappa statistic was 0.63, which represents a fair to good level of agreement.

Table 4: Run information for training data set

Measures	Value
Correctly Classified Instances	92.2 %
Incorrectly Classified Instances	7.8 %
Kappa statistic	0.63
Mean absolute error	0.063
Root mean squared error	0.178
Relative absolute error	48.2%
Root relative squared error	70.1%

Table 5: Confusion matrix for training set. The rows represent the subjective response; the columns are the classifier result.

	Expert	Average	Poor	Novice
Expert	320	0	0	0
Average	22	13	0	2
Poor	2	1	4	1
Novice	0	1	0	8

Sixteen rules were obtained from the decision tree and each rule starts with menu-bar count which is a key feature. Among the six identified features, four features were used for building the expertise model.

The confusion matrix obtained for training data is as shown in the Table 5. 85.56% of total instances belonged to experts. This resulted in classifying most of the instances as expert instances and mis-classified most of the average users as experts. As the training data is limited, it is difficult to test the model to find out exact error rates. A statistical technique called cross validation was used [57].

The run information for classification using 10 fold cross validation is shown in Table 6. The classification accuracy using J48 and 10-fold cross validation was 89.8% and the Kappa statistics was 0.51, which represents a fair to good level of agreement.

Table 6: Run information for 10-fold cross validation

Measures	**Value**
Correctly Classified Instances	89.8%
Incorrectly Classified Instances	10.2%
Kappa statistic	0.51
Mean absolute error	0.078
Root mean squared error	0.214
Relative absolute error	59.1%
Root relative squared error	84.3%

The Confusion matrix obtained for 10-fold cross validation is as shown in Table 7. Here most of the users are classified as experts. Again, most of the average users are classified as expert users.

Table 7: Confusion matrix for 10-fold cross-validation; Rows-subjective response; Columns-classifier result.

	Expert	Average	Poor	Novice
Expert	319	1	0	0
Average	22	12	0	3
Poor	2	4	0	2
Novice	1	2	1	5

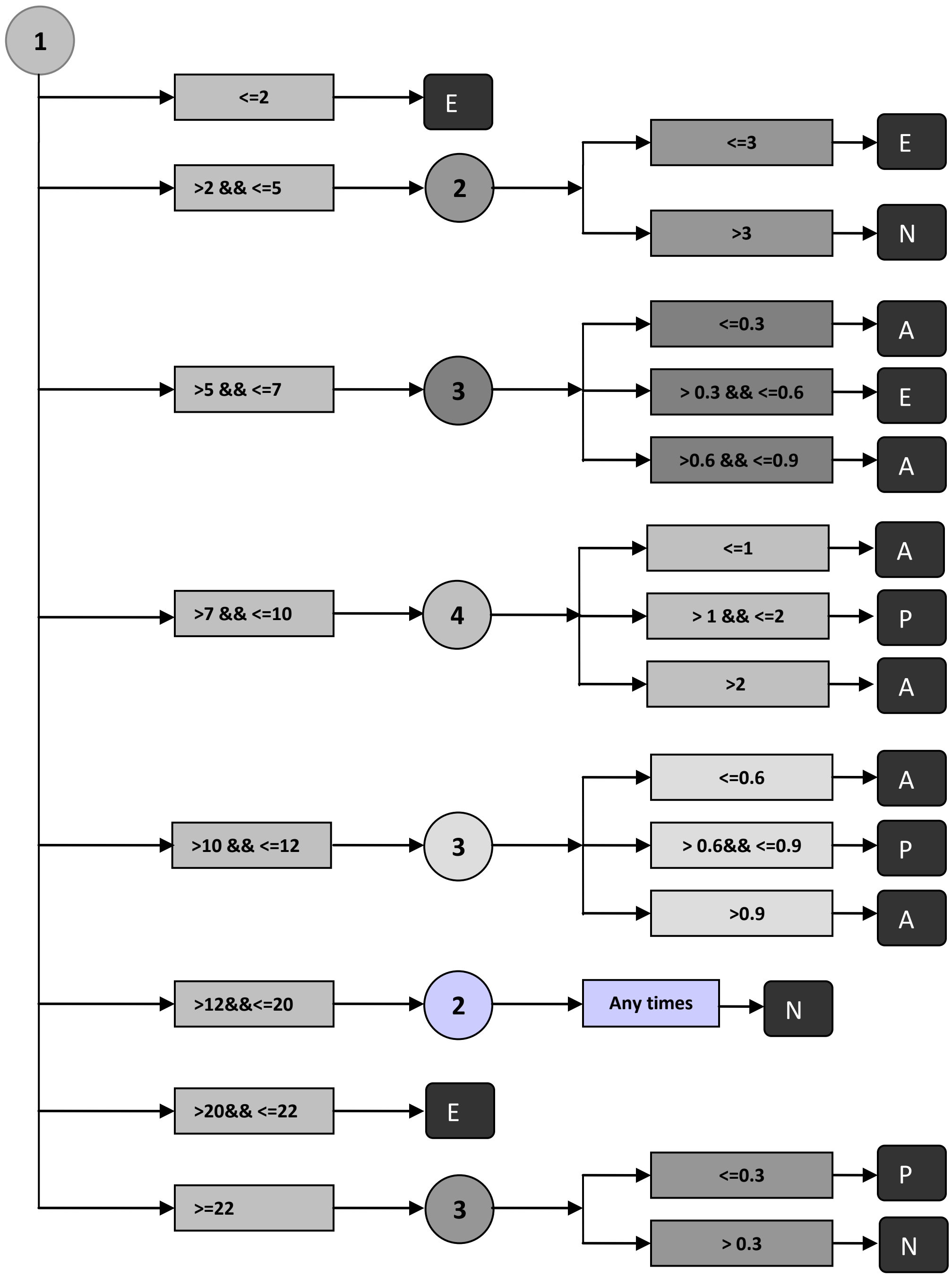

Figure 12: Obtained Decision Tree, where: E-Expert, A- Average, P-Poor, N-Novice. 1-Menu bar Count, 2-Cancel Count, 3- Dwell Time for Single Click, 4- Special Key Count.

Based on the obtained decision tree (see Figure 12), we observed that the menu check count was a key feature for classifying the user's expertise. As such, menu bar checking activates the classification of the expertise model; if a user does not check any menu options, it is assumed that they are an expert user. Among six features, the model considered four features (i.e. menu-bar count, cancel count, single-click dwell time, and special key count).

Class Skew Problem

We used Microsoft Word, a common application that is familiar to users to gather realistic usage data for creating the model. Although, we assigned tasks of varying familiarity within Microsoft Word and participants were not from Computer Science background, participants had knowledge about Microsoft Word and tasks in it. Among the collected instances 85.56% of the instances were expert. This led to a training data set with "Expert" class having highly skewed distribution compared to other classes. Eventually (as shown in Table 6) 91.9% of the instances were classified as expert. However, this problem may be improved by collecting a very large training data set or more work needs to be conducted using a greater proportion of unfamiliar tasks. We address more about this problem in Chapter 7.

4.7.2 Rules

From the decision tree, we developed a series of rules that are used in a prototype to detect the user's expertise level in real time.

A real-time system is used in our fourth experiment in order to provide users with different help depending on their expertise. Our real-time prototype system is task-independent as it is built from task-independent features.

Rules were extracted from the decision tree where the leaf node contains the conclusion of the rule and the path traced from the leaf node to the root node yields the conditions for the rule. Each leaf node obtained from the decision tree represents one of the four expertise levels. Rules obtained from the decision model are:

1. If (MenuBarCount < = 2)

   Then Expert User

2. If (MenuBarCount > 2 && MenuBarCount <=5)

   And If (CancelCount <=3)

   Then Expert User

3. If (MenuBarCount > 2 && MenuBarCount <=5)

   And If (CancelCount >3)

   Then Novice User

4. If (MenuBarCount > 5 && MenuBarCount <=7)

   And If (DwellTimeForSingleClick <=0.35)

   Then Average User

5. If (MenuBarCount > 5 && MenuBarCount <=7)

   And If (DwellTimeForSingleClick >0.35 && DwellTimeForSingleClick <=0.6)

   Then Expert User

6. If (MenuBarCount > 5 && MenuBarCount <=7)

   And If (DwellTimeForSingleClick > 0.6)

   Then Average User

7. If (MenuBarCount > 7 && MenuBarCount <=10)

   And If (SpecialKeyCount =1)

   Then Average User

8. If (MenuBarCount > 7 && MenuBarCount <=10)

And If (SpecialKeyCount =2)

Then Poor User

9. If (MenuBarCount > 7 && MenuBarCount <=10)

And If (SpecialKeyCount >2)

Then Average User

10. If (MenuBarCount > 10 && MenuBarCount <=12)

And If (DwellTimeForSingleClick<=0.6)

Then Average User

11. If (MenuBarCount > 10 && MenuBarCount <=12)

And If (DwellTimeForSingleClick > 0.6 && DwellTimeForSingleClick<=0.9)

Then Poor User

12. If (MenuBarCount > 10 && MenuBarCount <=12)

And If (DwellTimeForSingleClick > 0.9)

Then Average User

13. If (MenuBarCount > 12 && MenuBarCount <=20)

Then Novice User

14. If (MenuBarCount > 20 && MenuBarCount <=22)

Then Expert User

15. If (MenuBarCount >22)

And If (DwellTimeForSingleClick <=0.3)

Then Poor User

16. If (MenuBarCount >22)

And If (DwellTimeForSingleClick > 0.3)

Then Novice User

4.8 Discussion of Experiment Two

We created a model that classifies user expertise into four levels. This model was built with features which are extracted from users' interactions with Microsoft Word. Some of the features are similar to that of Hurst et al. [24], but our model differs in following ways:

- First, our model classifies expertise into four levels, rather than two levels. Based on the results of Experiment One, we feel this is an important advance toward creating systems that do not employ an all-or-nothing approach to suggested help, but rather allow for helpful suggestions that vary in their assertiveness and subtlety.

- Second, as we used a real application and real tasks, the proportion of labeled cases in each expertise category were not balanced. As expected, users rated themselves as expert most of the time (320 out of 372 trials). To improve the performance of an expertise classifier, more work needs to be conducted using a greater proportion of unfamiliar tasks. However, we enhanced the ecological validity of the data gathered for creating the model through the environment and tasks used. We used a common application that is familiar to users to gather realistic usage data. Along these lines, we had users complete tasks of varying familiarity within Microsoft Word, similar to what would be expected if they were actually using the application.

- Finally, although some tasks that the users performed were simple, others were high-level tasks comprised of multiple parts- similar to what one would experience in regular use of a familiar software application. Our goal is to build an effective expertise-based suggestive help system for use with common applications where a single user's expertise may vary dramatically from task to task; thus, we wanted the gathered data to be from an environment and task that were ecologically valid.

4.8.1 Usage of Built-In Help System

The goal of our work is to build a system that suggests help to users differently depending on their expertise thus perhaps avoiding the necessity of navigating through the help menu bars.

We explored whether users explicitly chose the built-in help option. We observed that most of the participants did not choose to ask the system for help and instead would search for the required menu item. Users only selected the help option 18 times out of 374 trials (4.8%), and the help selection feature was not included in the decision tree of the expertise model.

If users could not figure out how to complete the task, many users would wait three minutes for the experimenter to instruct them rather than choose the system's help. When we asked one user about the help option after the completion of the experiment, she responded that she had forgotten about the help option even though she knew that it existed. One participant told us that he hates the help option in any system because he never gets the exact answer for his problem – rather help systems give a set of solutions for one problem and we need to search for the required one. When system help was explicitly suggested in Experiment One, it was accepted 16.1% of the time – a substantial increase.

Users may have been reluctant to search for system help given that they only had to wait three minutes for the experimenter to assist them. Also, some users treated the experimental situation like a challenge and commented that they wanted to try to figure it out themselves. These behaviors do not transfer to real work environments, but the help-usage data from both experiments support the idea that users could benefit from system-suggested help when they are struggling with unfamiliar tasks.

4.9 Conclusion of Experiment Two

We conducted an experiment to build an expertise model that classifies user expertise into one of four levels. Our results show the following:

- User's expertise varies between tasks and user's expertise level increases with their familiarity with the task.
- As a user repeats the same task, his/her expertise level with that task increases.
- A model is built for determining the approximate user's level of expertise and the model predicts expertise at a finer granularity than two levels. This model is built from features derived from mouse use, keyboard short cuts, and menu data. This model classifies the user's expertise level as one of the four levels. The four levels are Expert, Average, Poor and Novice.
- We enhanced the ecological validity of the data gathered for creating the model through the environment and tasks used. We used a common application that is familiar to users to gather realistic usage data. However, the proportion of labeled cases in each expertise category were not balanced which may be solved by conducting an experiment using a greater proportion of unfamiliar tasks.

CHAPTER 5

EXPERIMENT THREE: CHOOSING NOTIFICATION SIGNALS

5.1 Motivation and Background

From our previous two studies, we established that:

- Users' perceptions about help interruptions vary with their expertise level;
- A single user's expertise level varies from one task to another;
- An expertise model that classifies user expertise as one of the four levels is valuable;
- An expertise model was built with four features derived from menu bar data, keyword data and mouse click events. This classifier predicts four levels of expertise (expert, average, poor, and novice) with an accuracy of 90%.

Our next step is to design expertise-sensitive help suggestions with different levels of attentional draw at the task level so that systems could recommend help overtly to novices, subtly to poor and average users, and not at all to experts. To do so, we need three visual signals that have varying levels of attentional draw so that different visual signals could be used to provide help suggestions to three levels of non-expert users (novice, poor and average). In this chapter, we explain our third experiment that we conducted to select three visual icons for providing help suggestions at three levels of subtlety (as it is not required to provide help to experts).

We wanted to explore three visual signals with varying attentional draw, with varying annoyance, with varying ignorability, and with varied intrusiveness in order to provide help suggestions to users with different expertise levels. The main intention was to find out one visual signal with high attentional draw to provide help suggestions to novice users, one visual signal with medium attentional draw to provide help suggestions to poor users and another visual signal with low attentional draw to provide help suggestions to average users. This approach is needed because our previous studies showed that help suggestions are required for and desired by novice users; help suggestions are less required for and desired by poor users who might have some knowledge about the task; and help suggestions are least required for and desired by average users as they have slightly more knowledge about the task.

In the following subsections we talk about the hypotheses of this experiment, the designed visual signals, the experimental setup and procedure, and the results.

5.2 Goals of this experiment

As a model was built for classifying a user expertise level as one of four expertise levels, three visual signals with different attentional draw were required for providing help to users with three levels of expertise. This was done because we wanted to provide help suggestions only to non-expert users (novice, poor and average) as expert users do not need help suggestions. As such our goal was to choose three visual signals within designed six visual signals. We wanted three different signals having following different characteristics:

- A visual signal with low attentional draw, lower intrusiveness and should be highly ignorable compared to other visual signals and this signal is to provide help suggestions to users with expertise level “average”;

- A visual signal with medium attentional draw, medium intrusiveness and should be slightly ignorable compared to other visual signals and this signal is to provide help suggestions to users with expertise level “poor” ;

- A visual signal with high attentional draw, high intrusiveness and should not be ignorable compared to other visual signals and this signal is to provide help suggestions to users with expertise level “Novice”. However, we did not want to design a visual signal which is too annoying and interrupting to novice users but compared to other visual signals, it should be highly annoying and less ignorable.

5.3 Design Options for Subtly-suggestive Help

There are many options for designing signals that are more subtle than standard pop-up signals. For example, this could take the form of haptic tunnels in the interface created with pseudo-haptic feedback [37]. Alternatively, we could design solutions that leverage the advantages of the human visual system for action [9], by directing users towards icons without their conscious perception. These subtle approaches can be applied with varying levels of intrusion, as opposed to previous all-or-nothing pop-up approaches.

The approach we chose to take for subtly suggesting help is to use notifications that reduce the attentional draw of the help interruption. Researchers have looked into the benefits of matching attentional draw with incoming message utility [16]; applying the same attentional draw design principles to expertise will create help notifications that vary in their subtlety. For example, the use of motion can lead to better detection and reduced irritation in notification icons, by leveraging visual pre-attentive processing [16]. We propose to create subtle help interruptions based on detectability and ignorability of visual notifications.

5.4 Designing Visual Signals

We based our visual signal designs on research by Gluck et al. [16] where three visual signals with varied attentional draw were used. One of these visual signals called "Follow" (which follows the mouse cursor around) was detected as the highest attentional draw signal and the most annoying. As we wanted to provide help suggestions that are optional and that are not too annoying, we did not use the "Follow" visual signal but we used the same approach for designing a basic set of visual signals as used by Gluck et al. [16].

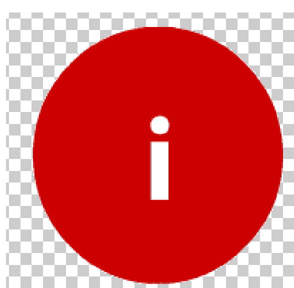

Figure 13: Basic shape of Visual Signal

Table 8: Initial set of Visual Signals

Category A: Continuous Slow State Change	
Colour (Figure 14)	Background colour of a visual signal is changed gradually. Entire background change loop is repeated after 1.6 seconds.
Grow (Figure 15)	Size of a visual signal is increased gradually. Entire growth loop repeated after 1.6 seconds.
Bounce (Figure 16)	Visual signal is made to bounce gradually. Entire bounce loop took 1.6 seconds to complete.
Category B: Continuous Fast State Change	
Colour Fast (Figure 17)	Background colour of a visual signal is changed quickly. Entire background change loop is repeated after every 0.4 seconds.
Grow Fast (Figure 18)	Size of a visual signal is increased quickly. Entire growth loop is repeated after every 0.4 seconds.
Bounce Fast (Figure 19)	Visual signal is made to bounce quickly. Each bounce loop is repeated after every 0.4 seconds.

The basic shape of all visual signals is as shown in Figure 13. Six visual signals were designed and classified into two categories. The two categories are: continuous slow state change and continuous fast state change. These two categories were formed based on work done by Gluck et al. [16]. We designed our motion-changing icons based on foundational research on the perception of moticons done by Bartram et al [8]. Table 8 contains the information about the basic set of visual signals designed for this experiment. These visual signals were animated images created with Adobe Flash and are as shown in Figure 8-13.

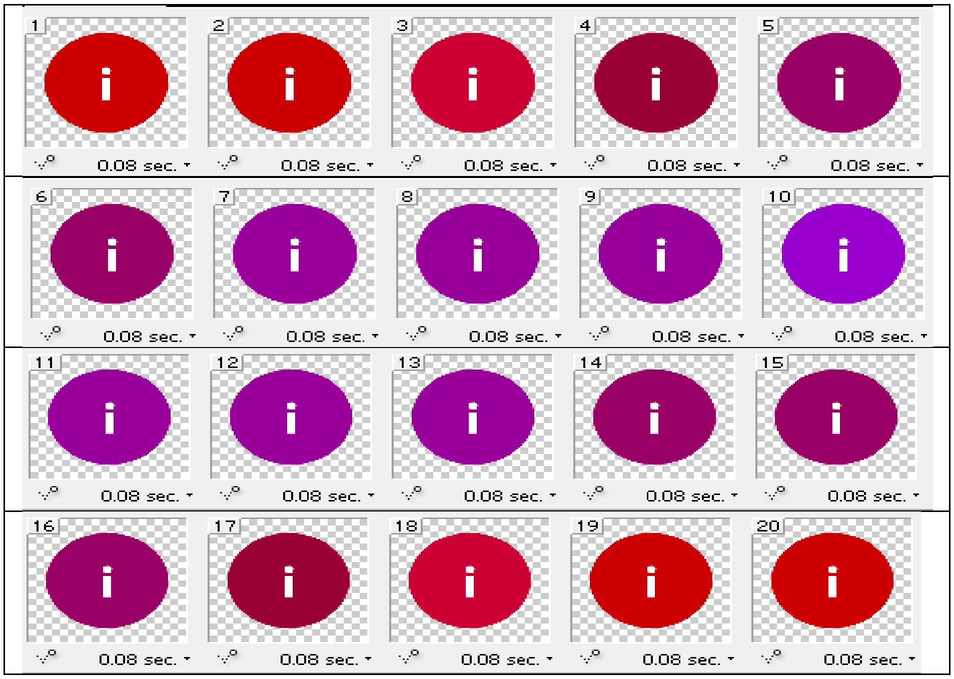

Figure 14: Frames of “Colour” visual signal along with time allocated for each frame.

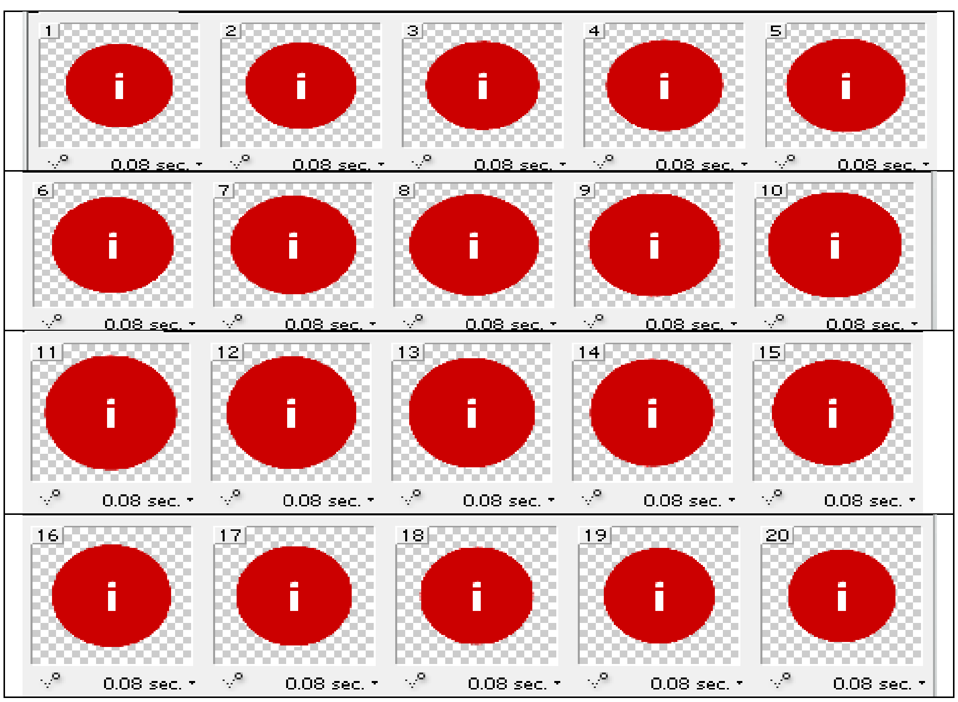

Figure 15: Frames of "Grow" visual signal along with time allocated for each frame.

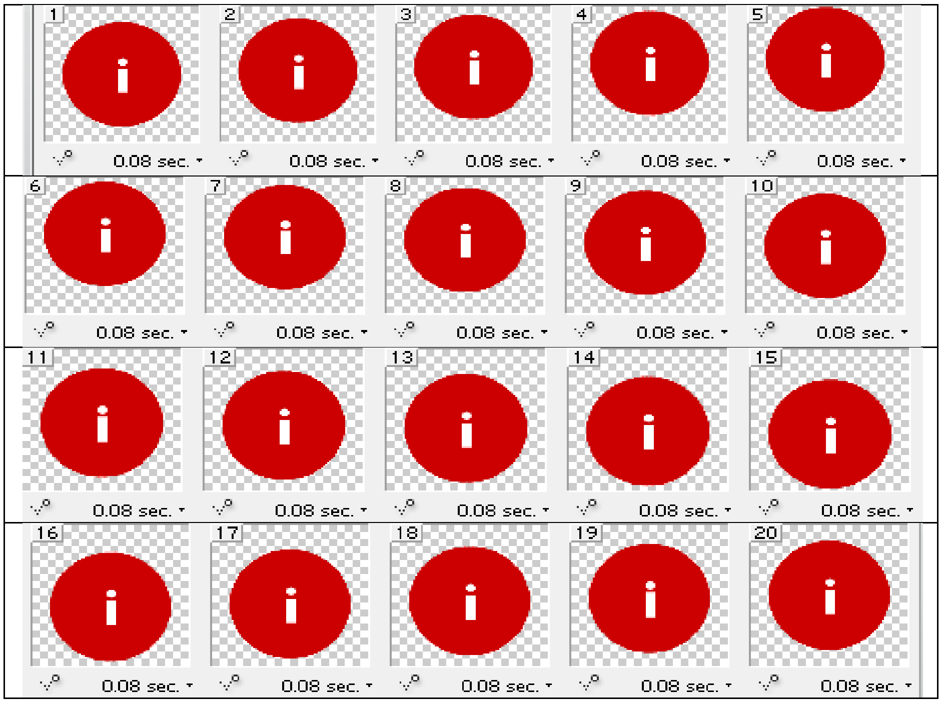

Figure 16: Frames of "Bounce" visual signal along with time allocated for each frame.

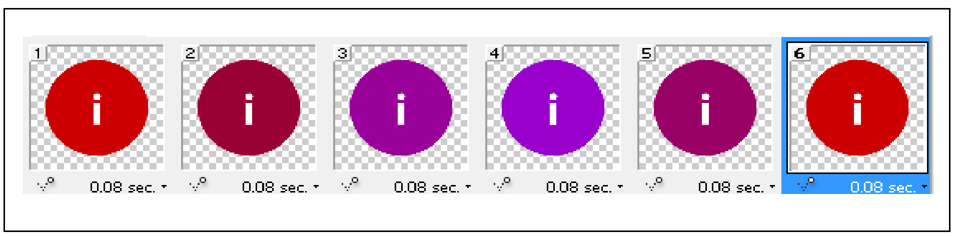

Figure 17: Frames of "Colour Fast" visual signal along with time allocated for each frame.

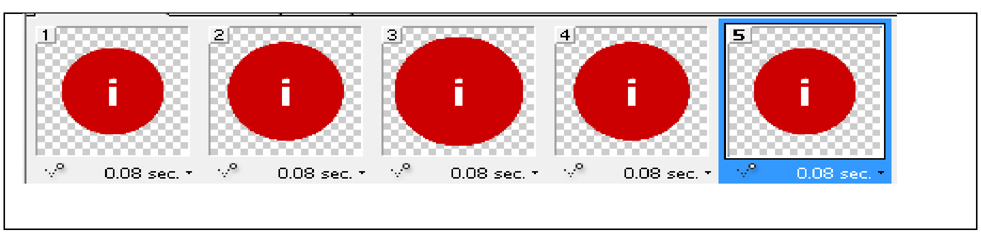

Figure 18: Frames of "Grow Fast" visual signal along with time allocated for each frame.

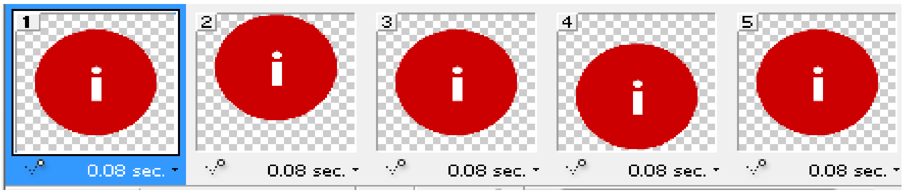

Figure 19: Frames of "Bounce Fast" visual signal along with time allocated for each frame.

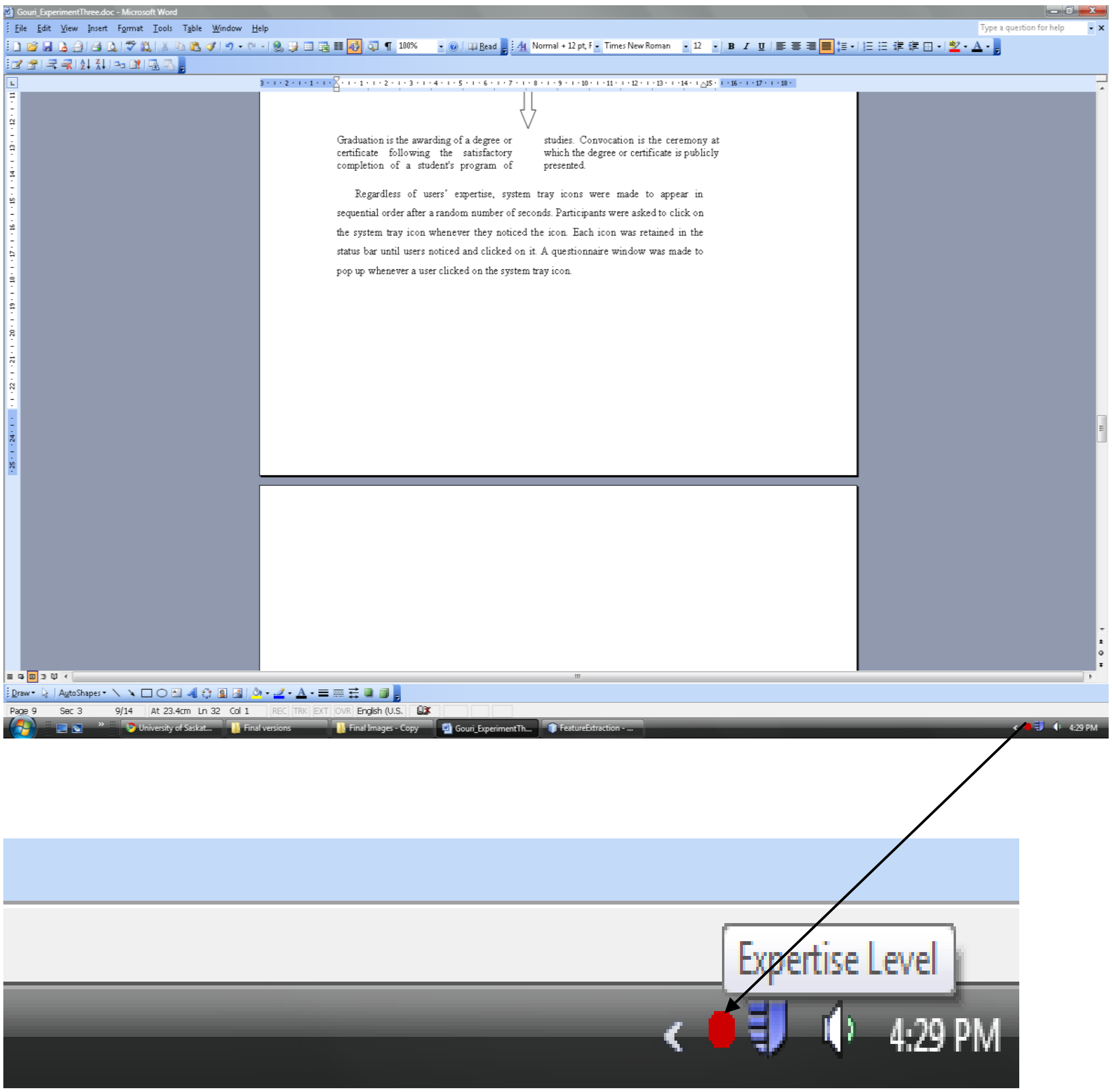

Figure 20: A visual signal displayed as a system tray icon.

In this experiment, each visual signal was used as a system tray icon i.e. a visual signal was made to appear at right bottom corner of the screen (see Figure 20).

5.5 Experimental Setup and Procedure

This experiment was conducted on a PC running Windows XP and MS Word 2003. There were 5 participants (male). For each user, the entire experiment took 10-12 minutes to complete. All participants had background knowledge in Computer Science.

Users were given a set of tasks (Table 9) and a short explanation for each of them. These descriptions were clear enough to ensure that the user understood the task without providing information on how to complete it. Participants did not receive help from the experimenter if they could not complete the task. Users could complete all of the tasks through menu selection and some of the tasks through the use of keyboard shortcuts; the choice of approach taken was left up to the user.

Table 9: List of tasks chosen in Microsoft Word for Experiment Three

Task No.	Task
1	Insert hyperlink to a file.
2	Increase the line spacing of the whole document to 1.5
3	Change the background colour of first page to green.
4	Translate the first paragraph into French.
5	Insert a picture (from a file) after the first paragraph
6	Insert a comment to any word in the document.
7	Add line numbers to a whole page.
8	Increase the number of columns to 2.

The tasks chosen were a subset of those used in Experiment One and Two. We chose the least familiar tasks as we wanted to study the attentional draw of our visual signals when users were not experts with the given task. The descriptions of each of the tasks are provided in Section 3.1.

Regardless of users' expertise, system tray icons appeared in order listed in Table 8. Each system tray icon appeared after a random number of seconds between 30 to 40 seconds. Participants were asked to click on the system tray icon whenever they noticed the icon. Each icon was retained in the status bar until users noticed and clicked on it. A questionnaire window was made to pop up whenever a user clicked on the system tray icon.

The questionnaire was designed to elicit participants' perceptions on being interrupted and to determine the noticeability ranking of the system tray icon. Users were asked to rate their agreement with the following statements on a five-point scale (Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree):

- This system tray icon is noticeable.
- This system tray icon is annoying.
- This system tray icon is intrusive.
- This system tray icon is ignorable.

Participants were asked to stop doing tasks when they noticed the last icon. At the end of this experiment, participants ranked the noticeability of all system tray icons. In addition, users were given an opportunity to comment on the experiment and on the system tray icon interruptions.

5.6 Analysis and Results

Subjective quantitative data collected during the experiment was analyzed to rank visual signals based on their noticeability, intrusiveness, ignorability, and annoyance. Also an average of the rankings of all visual signals collected from participants was taken.

Figure 21 indicates the mean ratings of attentional draw for all the icons. "Grow Fast" and "Bounce Fast" icons are more noticeable compared to all other icons and the least noticeable icons are "Grow" and "Colour". The order of noticeability ratings of the six visual signals (from less noticeable to more noticeable) was: Colour, Grow, Bounce, Colour Fast, Grow Fast and Bounce Fast.

Figure 22 indicates the mean ratings of annoyance, ignorability, intrusiveness and noticeability of each signal. Considering Figure 21 and Figure 22, “Grow Fast” seemed to be very noticeable, barely annoying, and more intrusive and less ignorable compared to other visual signals. Also, “Colour” seemed to be much less annoying, less intrusive and likely to be more ignorable.

Considering these results, “Grow Fast” is chosen as a visual signal with more attentional draw, more annoying, more intrusive, and less ignorable compared to other visual signals. “Colour” is considered as a visual signal with very less attentional draw, less annoyance, less intrusion and less ignorability. Considering the results in Figure 22, “Bounce Fast” had nearly same results as “Grow Fast” and “Bounce” had nearly same results as “Colour”.

As we needed a visual signal with medium attentional draw, “Bounce Medium” icon was chosen for representing a visual signal with medium attention draw. To make the bounce rate visually slower than “Bounce Fast”, one bounce loop time for “Bounce Medium” is 1.2 seconds. “Bounce Medium” visual signal’s frames are as shown in Figure 23.

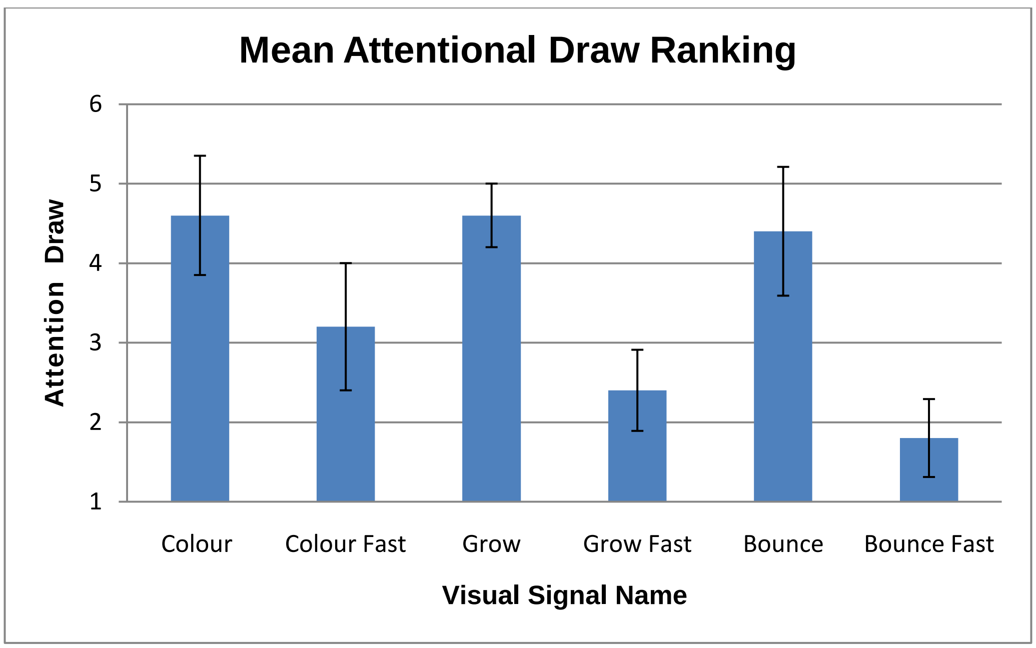

Figure 21: Mean of self-reported attentional draw ranking (±SD) of six visual signals (lower rating means higher attentional draw)

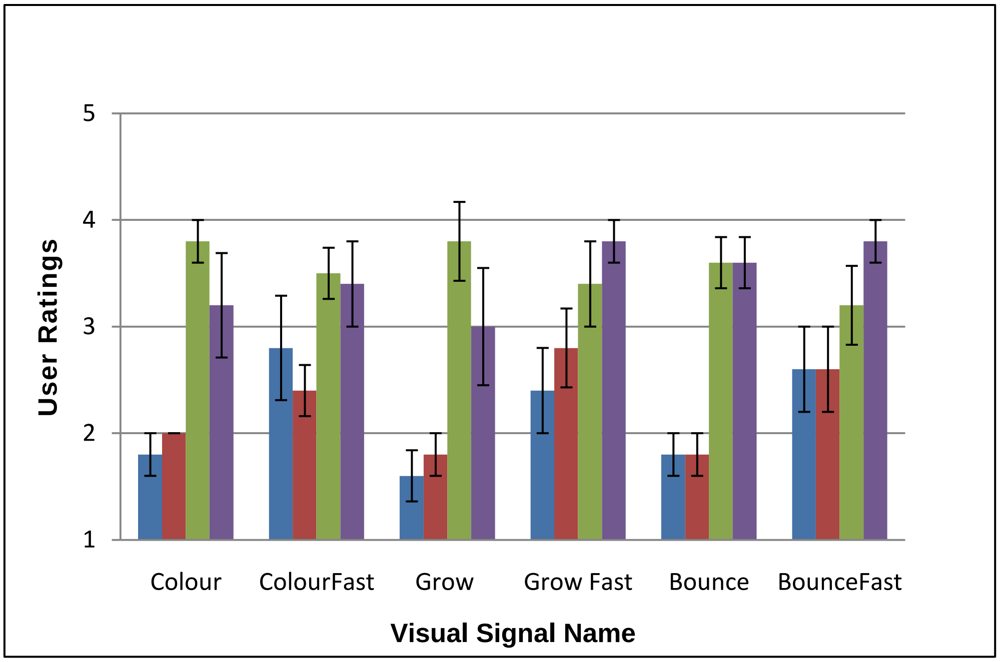

Figure 22: Means of agreement of annoyance, intrusiveness, noticeability and ignorability (±SD) with six visual signals (higher rating stands for more agreement).

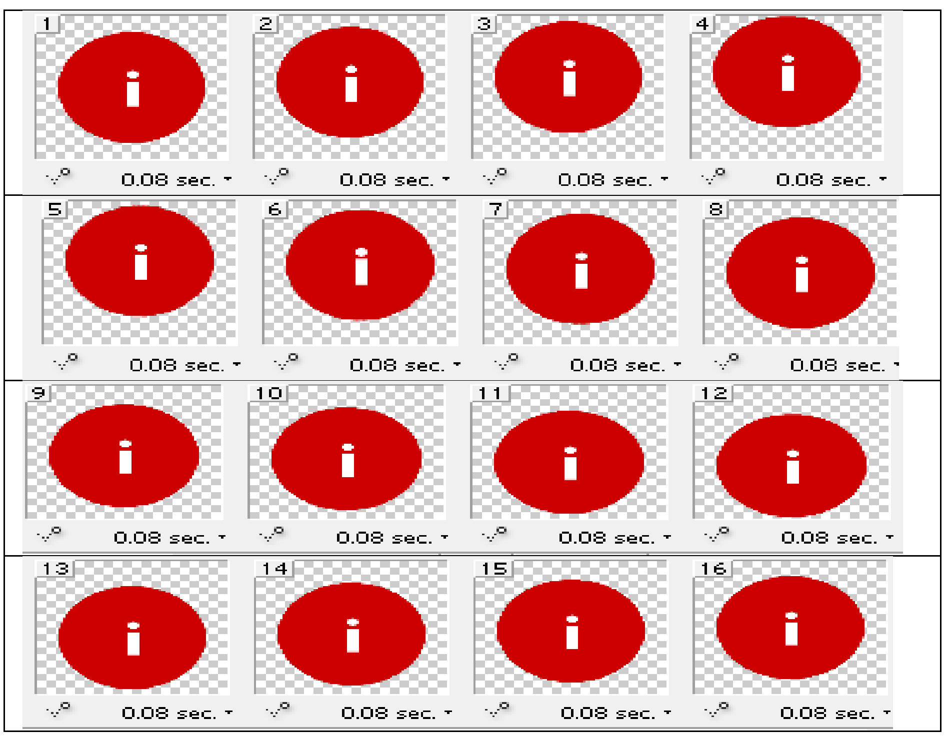

Figure 23: Frames of "Bounce Medium" along with time allocated for each frame.

We established three visual signals with different levels of attentional draw to use these visual signals for providing help suggestions to users with three expertise levels. We designed a visual signal with high attentional draw ("Grow Fast") to provide help suggestions to novice users, a visual signal with medium attentional draw ("Bounce Medium") to provide help suggestions to poor users, and a visual signal with low attentional draw ("Colour") to provide help suggestions to average users.

5.7 Summary

This experiment was conducted to elicit three visual signals with varied attentional draw, with varied annoyance, with varied ignorability and with varied intrusiveness. Initially a set of visual signals were designed and among them, "Colour", "Bounce Medium" and "Grow Fast" were chosen as a set of signals with different levels of attentional draw. These visual signals were used in our fourth experiment to provide help suggestions to non-expert users. In Chapter 6, we explain how we integrated these signals into our final experiment which provides overt help to novice users, subtle help to poor and average users, and no help to expert users.

CHAPTER 6

EXPERIMENT FOUR: EXAMINING EXPERTISE-SENSITIVE HELP SUGGESTIONS

We have previously discussed the disadvantages of built-in help systems and system-suggested help systems that do not take into account the expertise of the user. Our main goal of this thesis was to provide expertise-sensitive help suggestions that could solve problems associated with system-suggested help and built-in help systems. In previous experiments we: 1. demonstrated the need for expertise-sensitive help systems; 2. built a real-time dynamic model of expertise; and 3. designed three visual signals with varying attentional draw for use in expertise-sensitive help systems. This final experiment puts all of the results together and investigates whether expertise-sensitive help suggestions are really helpful.

We conducted this experiment by combining the contributions from experiment one, two and three to explore the efficacy of the intelligent help suggestions. In the first experiment, we found that expert users dislike help suggestions and non-expert users like help suggestions and this study also proved that an expertise model that classifies the user expertise level as one of the four levels (i.e. novice, poor, average and expert) is valuable. This encouraged us to build an expertise model in the second experiment that can classify user expertise level into one of the four levels (i.e. novice, poor, average and expert). In the third experiment, we found three visual signals with different levels of attentional draw to provide help suggestions to non-expert users (novice, poor and average). In the final experiment, we combine the results from these three experiments to provide an expertise-sensitive help system. Here, the help option was not provided as a pop up window; instead three visual signals with varied attentional draw, established in our third study, were used for providing help to non-expert users (novice, poor and average). This final experiment provides preliminary results on user response to expertise-sensitive help systems.

In the following subsections, we explain about the experimental setup, experimental procedure, and hypotheses of this experiment, analysis of the collected data, the results and a summary of this experiment.

6.1 Experimental Set up and Procedure

The experiment was conducted on a PC running Windows Vista and Microsoft Word 2003, with a standard 2-button plus scroll wheel mouse as the input device.

There were 5 participants. The experiment took approximately 10-12 minutes to complete. Participants completed multiple tasks of variable familiarity in Microsoft Word 2003. Users completed 10 tasks (Table 10) during the experiment.

Table 10: List of tasks chosen for Experiment four (in Microsoft Word-2003)

Task No	Task
1	Please insert a foot note for any word
2	Insert a hyperlink to a file
3	Justify all paragraphs in centre alignment
4	Change the background colour of first page to green
5	Use track change option in MS word to track the changes you make in the current word document
6	Translate the first paragraph into French
7	Insert a comment to any word in the given document
8	Include a table of contents to this document
9	Add line numbers to the whole page
10	Increase the number of columns to two

Ten tasks were completed in the same order by each participant. As in the previous study, the task descriptions were clear enough to allow the user to comprehend the task without providing information on how to complete it, and they were given unlimited time to understand the task before beginning.

Descriptions of some of the tasks (i.e. 2, 4, 6, 7, 9, and 10) were similar to the descriptions of those tasks provided in Section 3.1. Descriptions of other tasks (1,3,5,8) are the following:

Task 1: Please insert a foot note for any word

Description: For example a foot note has been inserted for a word "any" in the below sentence: Please insert footnote to any word in the document.

Task 3: Justify all the paragraphs in center alignment.

Task 5: Please rewrite this current sentence and your changes to this document should be tracked i.e. the changes you make should be tracked.

Description: For example: this is a track change usage.

Task 8: Please include a table of contents to this document

The expertise-sensitive help suggestion system and the system tray icons were explained to the users. User actions were logged with AppMonitor [2]. A custom Java program was written for analyzing the logged data from AppMonitor in real time and to decide the user expertise level based on the model built in the second experiment. The help tray icon appeared depending on the detected expertise level based on the data analyzed after every 15 seconds. Only one visual signal appeared at a time. "Grow Fast" appeared for novice users, "Bounce Medium" appeared for poor users and "Colour" appeared for average users. Each visual signal was displayed as a system tray icon for 15 seconds.

Users needed to complete all tasks in the given order one after another. They were asked to perform the tasks as if they were working on a document in their daily work. If they needed help and if they noticed any system tray icon and if they wanted to take help from it then they could click on that icon. A help window (see Figure 18) appeared whenever a user clicked on system tray icon. The provided help window asked the user to enter the task name and provided the required menu item path to complete the task (see Figure 19). A questionnaire window popped up after the help window.

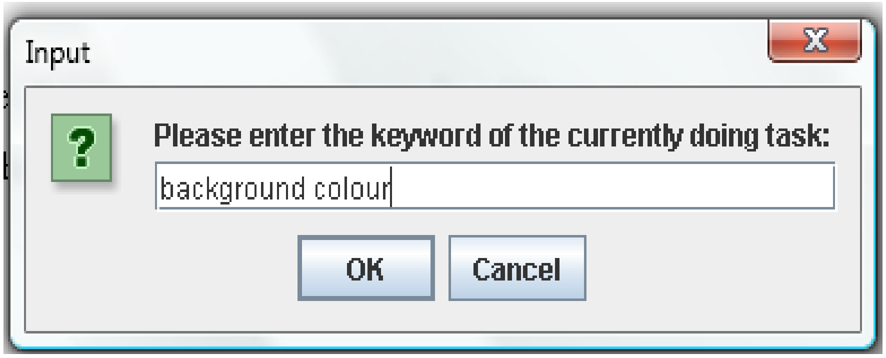

Figure 24: Input dialogue box that appeared after clicking the visual signal to get the current task information.

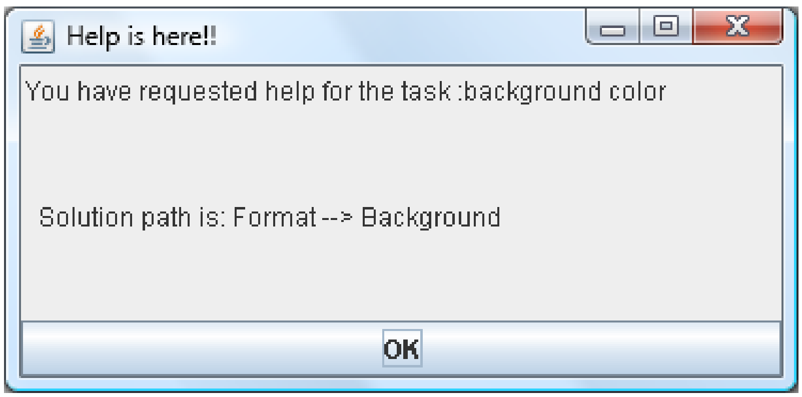

Figure 25: output dialogue box with solution path for a given task.

The questionnaire was designed to elicit participants' perceptions on being interrupted for help, to know whether help was useful for participants and whether it appeared at the right moment. Users were asked to rate their agreement with the following statements on a five-point scale (Strongly Disagree to Strongly Agree):

- The help provided was useful
- The help icon appeared at the right time
- The help icon was noticeable
- I did not need to be interrupted
- I am frustrated by the "help" interruption

In addition, participants were asked to write their opinion about the help option at the end of the experiment.

6.2 Hypotheses

Our assumption was that expertise-sensitive help suggestions would be beneficial and would not be annoying. As such, we had the following experimental hypotheses:

- Help suggestions will be useful.
- Help suggestions will appear at the right time.
- Help suggestions will not negatively interrupt the current task (even for novice users).
- Help suggestions will not be annoying.
- These hypotheses apply to all users with varying expertise levels.

6.3 Data Analysis and Results

Quantitative subjective data was collected from people who took the help by clicking on the system tray icon. Among the five users, 2 users did not use the help option. As shown in Figure 20, the mean ratings of all questionnaires were taken. From Figure 26, it is clear that most of the participants who used the help agreed that help was useful.

Our first hypothesis was that help suggestions would be useful. The mean (±SD) agreement obtained for the usefulness of help is 4.833. This rating indicates that our first hypothesis was supported and participants agreed that expertise-sensitive help suggestions were useful.

Our second hypothesis was that help suggestions would appear at the right time. The mean (±SD) rating obtained for the appearance of the help suggestion at the right time was 3.66. This rating suggests that on average participants agreed that system tray icon appeared at the right moment.

Our third and fourth hypotheses were that help suggestions will not negatively interrupt the current task and will not be annoying. The mean rating obtained for frustration with help suggestions was 1.66, the mean rating obtained for interruption felt because of help suggestions was 1.5 and the mean rating obtained for noticeability of help-suggesting visual icon was 2.833. These results satisfied our third and fourth hypotheses and participants agreed that expertise-sensitive help suggestions were not frustrating and not negatively interrupting to the current task.

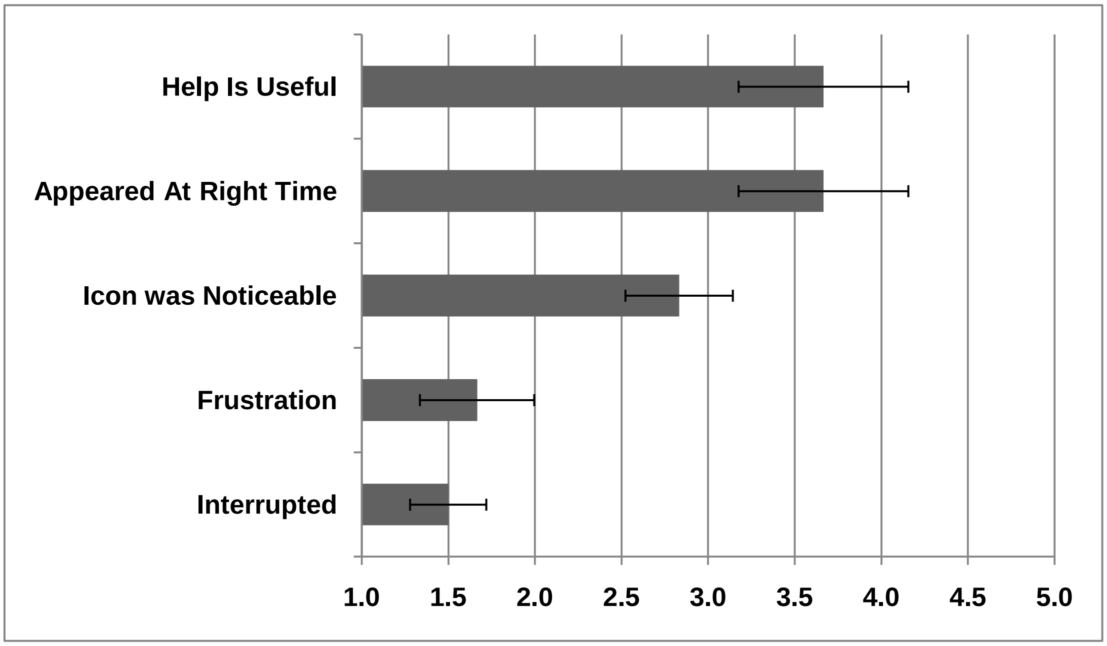

Figure 26: Mean self-reported ratings (±SD) regarding the provided help (higher rating is more agreement).

Participants' comments provide some explanation of our results. Participants felt that the system tray icons were not highly noticeable. This might be because of system tray icons with different levels of attentional draw. Another user reported that he usually searched for a while longer and then consulted the Word Help, which he found to be extremely painful and he felt that the helper tool was useful.

One user reported that help icons were not distracting enough to draw away his attention from a task that he was able to complete on his own. One user reported that he was looking for it before it appeared, but it usually appeared shortly after he began thinking he needed it. One user was protanopic colour blind and he had difficulty in recognizing red colour on a black background.

6.4 Summary of Experiment Four

We have established subtly-suggestive help through a controlled user study. We deployed the full system in a preliminary experiment to gather data on the efficacy and acceptability of expertise-sensitive subtly-suggestive help in real work environments. We found that most users found our intelligent help to be a useful tool and that our help appeared approximately at the right time.

6.5 Discussion

This experiment answers the key question of our thesis goal- whether users' response to intelligent expertise-sensitive help suggestions would be favourable. As we found that both frustration with the interruption and the lack of necessity for the help interruption increased with user expertise, we designed three visual signals with varied attentional draw. This was done because as the expertise level increases, users need less attention towards help suggestions. From the results of this experiment, we found that users' response to expertise-sensitive help suggestions is favourable. Results showed that multiple levels of interrupting help suggestions are helpful, help suggestions appeared at right time, help suggestions did not negatively interrupt the current task and help suggestions were not interrupting and frustrating.

One of the participants was protanopic colour blind and had a difficulty in identifying the visual signal as a system tray icon with black background. We based our visual signals' design on Gluck's experiment, however there might be other design options for the visual signals which we will be accessible to all users.

This final experiment provided preliminary evidence that expertise sensitive system-suggested help is valuable. In the next section we discuss the strengths and weaknesses of our implementation and how we can improve upon our current system.

CHAPTER 7

DISCUSSION

In this chapter, we elaborate on the findings of our studies on building an expertise model and examining expertise sensitive help systems. We begin by discussing the implications of our findings and then discuss the most important design lessons we learned through the course of our research.

7.1 Summary of Findings

The primary aim of our research was to determine how users perceive helpful interruptions, and to demonstrate that we can identify users' task expertise in multiple levels from novice to expert. Finally, our aim was to integrate this knowledge to build and examine an expertise-sensitive help system. Results from four studies (1. a study to determine users' perceptions of being interrupted with help interruptions; 2. an experiment to build an expertise model that classifies user expertise level into one of four levels; 3. a study to determine three visual signals with varied attentional draw to provide help suggestions; and 4. a study for examining an expertise-sensitive help system) yielded a number of important results. Here we summarize our most important findings.

1. User perception of help interruptions varied with their expertise with a given task.
    - User frustration with being interrupted increased with task familiarity.
    - User frustration with being interrupted increased with task expertise.
2. On tasks where users accept the help option, they were less frustrated with the interruption and agreed less strongly with the statement that they did not need to be interrupted.
3. Users agreed more strongly that they did not need to be interrupted with increasing task expertise.
4. Users' expertise varies between tasks and users' expertise levels increase with their familiarity with the task.

5. If the users repeat the same task more than once, their expertise level with that task increases.
6. We built a model for determining the user's approximate level of expertise and the model predicts expertise at a finer granularity than two levels. This model was built from features derived from mouse use, keyboard short cuts, and menu data. This model classifies the user's expertise level as one of the four levels –Expert, Average, Poor and Novice.
7. An experiment was conducted to inform the design of three visual signals with varied attentional draw, with varied annoyance, with varied ignorability and with varied intrusiveness. Based on the results, a set of visual signals were designed and among them, "Colour", "Bounce Medium" and "Grow Fast" were chosen as a set of signals with different levels of attentional draw.
8. These three visual signals were used in our final experiment to provide subtle help suggestions to users based on their expertise level. We deployed the full system in a laboratory experiment to gather data on the efficacy and acceptability of expertise-sensitive subtly-suggestive help in real work environments. We found that most users found our system to be a useful tool.

Through the four experiments, we have explored an effective solution for expertise-sensitive system-suggested help. Our results show that users respond differently to helpful interruptions depending on their expertise at that particular moment, not their overall expertise as a computer user. Also, results from a final experiment integrating an expertise model with a system for suggesting help with different levels of attentional draw are promising.

7.1.1 Our expertise model differs from others

Our user study in Chapter Four was modeled after a similar study performed to assist novice users by adapting the interface [24]. The results of Hurst et al's research agree with the results of our study, suggesting that it is possible to classify user expertise levels and assist users according to their expertise level. Both studies have built expertise models and found that users spend less time to complete the task as they repeat the task. Our model uses some similar features (to build a user model) to that of Hurst et al. [24], but differs in two key ways.

First, our model classifies four levels of expertise rather than two. Based on the results of Experiment One, we feel this is an important advance toward creating systems that do not employ an all-or-nothing approach to suggested help, but rather allow for helpful suggestions that vary in their assertiveness and subtlety.

Second, we enhanced the ecological validity of the data gathered for creating the model through the environment and tasks used. We used a common application that is familiar to users to gather realistic data. Along these lines, we had users complete tasks of varying familiarity within Microsoft Word, similar to what would be expected if they were actually using the application. Although some tasks that the users performed were simple, others were high-level tasks comprised of multiple parts, similar to what one would experience in regular use of a familiar software application. We have built an effective expertise-based suggestive help system for use with common applications where a single user's expertise may vary dramatically from task to task; thus, we wanted the gathered data to be from an environment and task that were ecologically valid.

7.2 Lessons Learned

Several of our findings were particularly surprising or noteworthy. Here we summarize the most important lessons we learned from our experiments.

We demonstrated that a multi-level expertise classifier can be built which uses low-level features and does not rely on knowledge of a user's task, but detects a user's expertise dynamically and regularly. Our system accurately classifies users into four levels of expertise based on features extracted from their use of MS Word. We aim to provide a solution for applications where users have a high level of expertise with most of the tasks, in order to help users improve their use of common productivity software as opposed to helping them learn unfamiliar applications. Our model was developed with data from a familiar application, which resulted in high levels of expertise for the majority of our labelled samples in Experiment One (for finding whether users' perceptions of being interrupted with help suggestions vary with their expertise of currently doing task) and Experiment Two (for building an expertise model). Although this imbalance may have decreased the performance of our classifier, it represents a realistic scenario. As such, we expect our results to generalize better than if we had collected our data in a more controlled and less realistic scenario.

7.2.1 Limitations of the Studies

Although our studies showed consistent patterns for user perceptions of interrupting help, there are some limitations to our experimental design and results. We explain them in this subsection.

Reasons for Frustration

Although we know that users' frustration with suggested help increases with their task expertise, our results do not differentiate between annoyance with the interruption itself and annoyance with a system that erroneously suggests that the user needed help with the task. In a realistic deployment of our system, users are interrupted less often and only when it is necessary. Although our final experiment moves closer to this ideal, data from field trials where help is only suggested to users when their expertise falls below average is needed to determine the source of our users' frustration.

Generalization to Real Computer Use

In our experiments, users were completing realistic tasks; however, the environment for completing the tasks was unrealistic. In real work tasks, users want to complete their work efficiently and may feel more strongly against being interrupted, even if they require assistance. On the other hand, our users were regularly interrupted during every task in the first three experiments; a system that suggests help only when a user's task expertise is detected as below average may not be seen as being as interrupting in regular computer use. In our final experiment, we chose tasks that we thought would be unfamiliar to users, so users were likely interrupted more than they would be during regular computer use.

Interruption Frustration versus Study Frustration

In our initial experiments, the help option and questionnaire window popped up on every task. When users did not require help, they may have been frustrated or irritated with having to answer the questions, which might have affected their perceptions of the help interruption. Users who accepted and appreciated system help might have had their opinions of the help interruption biased by having to repeatedly complete the questionnaire. In addition, as there was no lengthy break between tasks, frustration from being interrupted when they did not require help might have carried over into tasks where they did require system intervention. Finally, users who completed the task quickly and needed to wait to be interrupted may have felt frustration with the wait which could have carried over to their response to the system.

Types of Novice Users

In the First Experiment (for determining whether users' perceptions of being interrupted with help suggestions vary with their expertise) that is explained in Chapter Three, if users did not know how to complete the experimental task, they either embraced the challenge or tried to figure it out through the interface, or they simply waited 45 seconds for the help option to appear. In real work environments, we might expect this latter group of users to activate system help explicitly, or to try to figure out the solution through the interface, documentation, or colleagues. Regardless of the user's natural response to a task with an unknown solution, system-suggested help is beneficial.

Imbalance of Expertise Cases

We demonstrated the feasibility of creating a four-level classifier with real tasks in a familiar application. This ecologically-valid approach was important to determine whether an expertise classifier would perform well in general computer use. Because we used a real application and real tasks, the proportion of labelled cases in each expertise category were not balanced. As expected, users rated themselves as expert most of the time (320 out of 372 trials). Had our classifier simply selected expert all of the time, it would have achieved good accuracy. Although we could have used a contrived task that would have better balanced the labelled cases (e.g.,

[24]), we wanted to demonstrate the feasibility of a classifier that works on natural behaviour in a realistic environment. To better the performance of an expertise classifier, more studies need to be conducted using a greater proportion of unfamiliar tasks. We discuss this further in Appendix B.

Incorrect help- how often will that happen?

In our final experiment (Chapter 4) help suggestions varied in their assertiveness based on users' expertise. We found that expertise-sensitive help suggestions are helpful and appear at almost right time. But our results did not indicate the likelihood of pop-up help being correct. This must be investigated in future work to avoid users' frustrations with incorrect help suggestions as our results all assume that system-suggested help will provide instructions or tutorials on exactly what users are looking for.

7.2.2 Targeted Help

Users found the provided help suggestions in the first experiment useful and easy to follow. Although Word provides help, the built-in system requires searching through several solutions to find the desired one. Users commented that our targeted help was clearer and more useful. For example, "The help feature being tested was useful, especially because the commands it suggested were streamlined, rather than having dozens of options like in MS Word.", and "I think that the pop-up thing could be very useful, and is certainly clearer than the regular 'help' from Microsoft Word."

Our help solutions were hard-coded according to the study tasks, but our users' comments suggest that they would appreciate pop-up help if it was targeted at their current task. The comments also suggest that they were unfamiliar with the targeted help options provided in previous versions of MS Word. Although our expertise classifier is task-independent, there is prior work on task detection [20] that could be combined with our approach to realize this system. With an expertise-sensitive approach, targeted help could be well received by a broad range of users.

7.3 Other Design Options for Subtly-suggestive Help

For users with moderate expertise, system-suggested help was very useful; however, we did not want to annoy or frustrate users with overt interruptions. We worked on developing and evaluating subtly-suggestive help. We made use of an approach for subtly suggesting help notifications that reduced the attentional draw of the help interruption by matching attentional draw with incoming message utility [16]. In particular, we proposed help interruptions by matching detectability and ignorability with user level of expertise. These subtle approaches applied with varying levels of intrusion, as opposed to previous all-or-nothing pop-up [24] approaches.

As we explained in Chapter Six, non-expert participants felt grateful for the help suggestions and expert participants did not interrupted with help suggestions. One of the participants was protanopic colour blind and he had difficulty in recognizing red colour on a black background. This problem can be solved by considering other design options for subtly-suggested help. For example, this may take the form of haptic tunnels in the interface created with pseudo-haptic feedback [37]. Alternatively, design solutions can be done that leverage the advantages of the human visual system for action [9] by directing users towards icons without their conscious perception.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the solution to the problem addressed in this thesis and discusses the scope for future work. We begin by giving a brief summary of the solution. Then an outline of the main contributions of this research is presented. We conclude by discussing avenues for future work.

8.1 Summary

The problem addressed in this thesis was that system-suggested help systems are not favoured by all users because expert users hate system-suggested help suggestions and non-expert users feel grateful for help suggestions. There was no system-suggested help system that provides intelligent help suggestions that vary in their assertiveness. The problem had three main parts: first, finding whether users' perceptions of being interrupted with help interruptions vary with their expertise levels; second, building the expertise model to detect user level of expertise; and third, evaluating the expertise-sensitive help system.

We designed a series of experiments to build and validate the expertise model. First, we found out that users' perceptions of being interrupted with help suggestions vary with their expertise levels (novice, poor, average and expert); second, we built an expertise model that classifies user expertise into one of the four levels in Microsoft Word based on their mouse and keyboard events; thirdly, we established three visual signals to provide help suggestions with varying attentional draw to provide help suggestions; finally we evaluated a complete expertise-sensitive help system. The experimental results show that our model can be used effectively to predict user level of expertise and provide help suggestions according to it. The evaluation of the expertise-sensitive system-suggested help system shows that a program that provides help overtly to novice users, subtly to average users and not at all to expert users is beneficial and effective.

8.2 Contributions

The major contributions of this research is providing empirical evidence that expertise-sensitive help suggesting system is a valuable concept for improving available assistance to the users in interfaces.

In particular, our contributions are:

- A better understanding of users' perceptions of being interrupted with help interruptions in interfaces. Additional evidence that help interruptions are perceived differently by users depending upon their knowledge about a task. Research indicated that experts dislike help interruptions and non-experts feel grateful for the same.

- Established that a four-level classifier of user expertise is valuable.

- Established a four-level classifier of user expertise requiring no prior knowledge about the user or their task, which was created and validated with real tasks of variable familiarity in a familiar software application.

- A better understanding of users' expertise levels. Users' expertise levels vary from one application to other and also vary from one task to other task within an application.

- Additional evidence that user's expertise level increases as he/she repeats the task. Eventually, a time taken to complete the task will be decreased as he/she repeats the task.

- Designed intelligent help interruptions that vary in their assertiveness and explored the efficacy of the suggested design options for subtly-suggestive help

- A better understanding of the limits of expertise-sensitive help system. This thesis discusses issues that ought to be considered.

We present an effective solution for expertise-sensitive help, and our results also show that a global, user-based setting is not an appropriate solution as user expertise varies with task within a single software application. Our work shows that system-suggested help will be desired and accepted by users when these systems are designed to respond appropriately to user expertise.

8.3 Future Work

We have demonstrated the appropriateness, feasibility and efficacy of an expertise-sensitive help-suggestion system; however, our work has two main limitations: first, the generalization of our user perception results to real work environments; second, the accuracy and generality of the model. We consider these limitations for our future work.

First, we plan to test the generality of our approach across applications other than Microsoft Word. Second, we will try other approaches to provide subtle help suggestions based on user expertise-level, rather than only relying on system tray icons. Finally, we plan to deploy the full system in field trials to gather data on the efficacy and acceptability of expertise-sensitive subtly-suggestive help in real work environments.

In addition, accuracy could be increased by considering more features. For example, we did not use the low-level motion characteristics from targeting movements; including these in the classifier could improve classification accuracy. We are furthering work on our classifier by integrating additional features derived from low-level motion of the pointing device as well as general aspects of a user model as we explain in Appendix B. Our classification results were limited by an imbalance in the number of cases labeled as expert. We used Microsoft Word which is a commonly-used application that is familiar to users to gather realistic usage data in order to enhance the ecological validity of our solution. However, we had a main problem in our expertise model i.e. most of the instances were classified as experts. This is because, Microsoft Word is a commonly-used application and the majority of the collected instances were labeled as expert. We will continue to improve the performance of our classifier by designing experiments that provide a better balance of expertise cases, by identifying and integrating more features into our model, and by considering the use of alternate algorithms for modeling user expertise.

We could improve our classifier for real computer use through field trials and an experience-sampling approach, where users would be asked to submit their expertise at random intervals throughout the day. We chose not to use this approach in this first contribution, as it would be unclear from field-trial data where one task ended and the next task began.

8.4 Conclusions

System-suggested help has not been well received by users with a high level of expertise who do not like to be interrupted. To improve system-suggested help, we identified a series of problems to be solved. First, we needed to understand users' perception about helpful interruptions and how these perceptions changed with a user's general level of expertise with an application and how they changed dynamically with a user's changing expertise level as they complete different tasks within an application. Second, we needed to design an expertise model that could identify users' expertise dynamically within an application. Third, we needed to design a series of helpful interruption notifications that varied in their assertiveness and subtlety. Finally, we needed to combine all of these results together into an expertise-sensitive help suggestion system.

Through a series of four experiments, we addressed these problems. We gained an understanding of users' perceptions of helpful interruptions; we classified user expertise into four levels (expert, average, poor, and novice) based on low-level computer usage features; and we designed three visual signals that vary in their attentional draw. Finally, we combined these efforts into a single system that identifies user expertise at four levels dynamically, and suggests help to average, poor, and novice users with interruptions whose visual signal matches the user's level of expertise. Together, our results make excellent progress toward expertise-sensitive system-suggested help.

REFERENCES

[1] Adamczyk P. D. & Bailey B. P., If not now, when?: The effects of interruption at different moments within task execution, in: Human Factors in Computing Systems: Proceedings of CHI, 271-278, 2004.

[2] Alexander, J., Cockburn, A., & Lobb, R., AppMonitor: A Tool for Recording User Actions in Unmodified Windows Applications. Behaviour Research Methods, in press, 2008.

[3] Altmann E. M. & Trafton J. G., Task interruption: Resumption lag and the role of cues, Proceedings of the 26th annual conference of the Cognitive Science Society, 42-47, 2004.

[4] Anderson, J.R., Farrel, R., & Sauers, R., Learning to program in LISP. Cognitive Science, 87-129, 1984.

[5] Atkinson, R.K.. Optimizing learning from examples using animated pedagogical agents. Journal of Educational Psychology, 416-427, 2002.

[6] Avrahami, D., & Hudson, S. E., Augmenting an instant messaging client to balance user responsiveness and performance. Proceedings of CSCW, 515–518, 2004.

[7] Bailey B. P., Konstan J. A. & Carlis J. V., Measuring the effects of interruptions on task performance in the user interface, in IEEE International Conference on Systems, Man, and Cybernetics, 757-762, 2000.

[8] Bartram L., Ware, C., & Calvert, T., Moticons: detection, distraction, and task, International Journal of Human-Computer Studies, 515-545, 2003.

[9] Binsted, G., Visuomotor system uses target features unavailable to conscious awareness. Proceedings of NAS, 69-72, 2007.

[10] Bunt, A., Conati, C., & McGrenere, J., What role can adaptive support play in an adaptable system? Proceedings of IUI, 117-124, 2004.

[11] Catrambone, R., Stasko J., & Xiao, J., Anthropomorphic Agents as a User Interface Paradigm: Experimental Findings and a Framework for Research, Proceedings of CogSci, 166-171, 2002.

[12] Covi, L., Such Easy-to-Use Systems! How Organisations Shape the Design and Use of Online Help Systems. Proceedings of COOCS, 280-288, 1995.

[13] Gajos, K. Z., Everitt,,K., Tan,S.D, Czerwinski, M., & Weld,D.S., Predictebility and Accuracy in Adaptive User Interfaces. Proceedings of CHI, 1271-1274, 2008.

[14] Gentner, D., & Stevens, A., Mental Models. Lawrence Erlbaum, Hilldale, New Jersey, 1983.

[15] Gillie T. & Broadbent D., What makes interruptions disruptive? A study of length, similarity and complexity, Psychological Research, 243-250, 1989.

[16] Gluck, J., Bunt, A., & McGrenere, J., Matching attentional draw with utility in interruption, Proceedings of CHI, 41-50, 2007.

[17] Goldin, E.E., & Hayes-Roth, B., Individual differences in planning processes, Office of Naval Research, 1980.

[18] Greenberg, S., & Witten, I. H., "Adaptive personalized interfaces -A question of viability," Behavior and Information Technology 31-45, 1985.

[19] Hook, K.., Steps to take before intelligent user interfaces become real. Interacting with Computers, 409–426, 2000.

[20] Horvitz, E., Principles of mixed-initiative user interfaces, Proceedings of CHI, 159-166, 1999.

[21] Hubert L., Dreyfus , Dreyfus, S. E., & Athanasiou,T., Mind over machine: the power of human intuition and expertise in the era of the computer, The Free Press, 1986.

[22] Hui, B., Partridge, G., & Boutilier, C., A probabilistic mental model for estimating disruption. Proceedings of IUI , 287-296, 2009.

[23] Hui,B., & Boutilier, C., Who's asking for help?: a Bayesian approach to intelligent assistance. Proceedings of IUI, 186-193, 2006.

[24] Hurst, A., Hudson,S., & Mankoff, J., Dynamic detection of novice vs. skilled use without a task model, Proceedings of CHI, 271-280, 2007.

[25] Huck, S.W., Reading statistics and research , 5th Edition, Allyn and Bacon, 2007.

[26] Jackson, T., Dawson, R., & Wilson, D., The cost of email interruption. Journal of Systems and Information Technology, 81–92, 2001.

[27] Jokinen, K. & Kanto, K.., "User Expertise Modelling and Adaptivity in a Speech-based E-mail System". Proceedings of ACL, 88-95, 2004.

[28] Jameson, A.., Adaptive interfaces and agents. In Human-Computer Interaction Handbook, 305–330, 2003.

[29] Kaptelinin, V., Item recognition in menu selection: the effect of practice, Human factors in computing systems, 183-184, 1993.

[30] Kearsley,g., Campbell, R. L., Elkerton, J., Judd,W., & Walker,J., Online Help System: Design and Implementation Issues, Proceedings of CHI, 287-288, 1988.

[31] Kelleher, C. & Pausch, R., Stencils-based Tutorials: Design and Evaluation, Proceedings of CHI, 541-550, 2005.

[32] Kolodner, J.L., Towards an understanding of the role of experience in the evolution from novice to expert. International Journal of Man-Machine Studies, 497-518, 1983.

[33] Kreifeldt, J. G., & McCarthy, M. E., Interruption as a test of the user-computer interface. Proceedings of the 17th Annual Conference on Manual Control, 655-667, 1981.

[34] Kyle B. Murray, & Häubl, G., A human capital perspective of skill acquisition and interface loyalty, Communications of the ACM, 272-278, 2003.

[35] LaFrance, M., The quality of expertise: implications of expert-novice differences for knowledge acquisition, ACM SIGART Bulletin, 6-14, 1989.

[36] Landauer, T., & Nachbar, D,. Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and the width. Proceedings of CHI, 73–78, 1985.

[37] Lécuyer, A, Coquillart, S, & Kheddar, A., Pseudo-Haptic Feedback: Can isometric input devices simulate force feedback? Proceedings of IEEE Virtual Reality Conference, 18-22, 2000.

[38] Lee, E., & MacGregor, J., Minimizing user search time in menu retrieval systems. Human Factors,157–162, 1985.

[39] Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., & Bhogal, R.S.. The persona effect: Affective impact of animated pedagogical agents. Proceedings of CHI, 359-366, 1997.

[40] Linton, F., & Schaefer, H.P., Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use. User Modeling and User-Adapted Interaction, 181 – 207, 2000.

[41] Logan, G. D., Shapes of reaction-time distributions and shapes of learning curves: A test of the instance theory of automaticity. Journal of Experimental Psychology, 883–914, 1992

[42] Luening, E., Microsoft tool “clippy” gets pink slip. CNET New.com, 2001. (http://news.com.com/2100-1001-255671.html?legacy=cnet)

[43] Mackay, W., Triggers and barriers to customizing software. Proceedings of CHI, 153–160, 1991.

[44] McGrenere, J. & G. Moore., Are we all in the same "Bloat"? Graphics Interface. 187-196, 2000.

[45] Miller, S. L., Window of opportunity: Using the interruption lag to manage disruption in complex tasks, Proceedings of HFES, 245-249, 2002.

[46] Moreno, R., Mayer, R. E., Spires, H., & Lester, J., The case for social agency in computer-based teaching: Do students learn more deeply whttn they interact with animated pedagogical agents? Cognition and Instruction, 177-214, 2001.

[47] Murray, K. B., & Häubl, G., The fiction of no friction: A user skills approach to cognitive lock-in. S. M. Broniarczyk and K. Nakamoto, eds., Advances in Consumer Research, Volume XXIX, Association for Consumer Research, 11-18, 2002.

[48] Norico, A.F., & Stanley, J., Adaptive Human- Computer Interfaces: A Literature Survey and Perspective. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, 399-408, 1989.

[49] Norman, D.A., The trouble with Unix: The user interface is Horrid. Datamation, 139-150, 1981.

[50] Norman D.A., Draper, D.W., & Hillsdale, NJ.,Communication paradigms for a window system, in User Entered System Design: New Perspectives in Human-Computer Interaction,: Lawrence Elbraum Associates, 285-313, 1986.

[51] Norman, K., The Psychology of Menu Selection. Ablex Publishing Corporation, Norwood, New Jersey, 1991.

[52] Obermayer, R. W, & Nugent, W. A., Human-computer interaction for alert warning and attention allocation systems of the multi-modal watchstation. Proceedings of SPIE, 14-22, 2000.

[53] Ramachandran, A., & Young, M. R., Providing intelligent help across applications in dynamic user and environment contexts, Proceedings of IUI, 269-271, 2005.

[54] Soman, T. & Bobbie, P.O., "Classification of Arrhythmia Using Machine Learning Techniques," WSEAS Transactions on Computers, Vol.4, 548-552, 2005

[55] Sutch, A.L., “You know more than you think you do” –helping participants transfer knowledge, Proceedings of SIGUCC, 332-334, 2007.

[56] Wahlster, W., and Kobsa,A., “Dialogue-based user models,” Proc. IEEE, vol. 74, no. 7, 948-960, 1986.

[57] Witten, I.H., & Frank, E. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, 2005.

[58] Yallow, E., Individual differences in learning from verbal and figural materials, School of Educ. Stanford Univ., Aptitudes Research Project Tech. Rep. No. 13, 1980.

[59] Zhou, X., & Conati, C., Inferring user goals from personality and behaviour in a casual model of user affect. Proceedings of IUI, 211-218, 2003.

[60]http://209.85.173.132/search?q=cache:http://itec.sfsu.edu/wp/865wp/S05_865_dalby_ped_agent.pdf (July, 2009)

[61]http://www.ellisonconsulting.com/downloads/Seven_Golden_Rules_of_Online_Help_Design.pdf (July, 2009)

[62] http://library.thinkquest.org/C0110291/basic/brain/semanticVepisodic.php

[63] http://en.wikipedia.org/wiki/Fitts's_law (29th June, 2009)

[64] http://en.wikipedia.org/wiki/Steering_law (29th June, 2009)

[65] http://en.wikipedia.org/wiki/Keystroke-Level_Model (November 2009)

[66]http://gautam.lis.illinois.edu/monkmiddleware/public/analytics/decisiontree.html (July 2009)

APPENDIX A- EVALUATION MATERIALS

A.1 Experiment 1- Informed Consent Form

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project: **Task-dependent detection of user expertise**

Investigators: Dr. Regan Mandryk, Department of Computer Science (966-4888)

Mangalagouri Masarakal, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with automatically detecting the expertise of users with a specific software application. We will be using Microsoft Word for this study.

The goal of the research is to determine whether we can design an accurate model of expertise for automatic and dynamic detection.

The session will require 15 minutes, during which you will be asked to carry out a number of tasks in Microsoft Word.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within one month).

This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or videorecordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Anonymity will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

1. Office of Research Services, University of Saskatchewan, (306) 966-4053

Participant's signature:____

Date:_____________________

Investigator's signature:____

Date:_____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Office of Research Services at the University of Saskatchewan.

A.1 Experiment 1 – Demographic Survey

Task 1:

Cut, paste and replace.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 2:

Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 3:

Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 4:

Increasing the line space of the whole document and the required line space is 1.5.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 5:

Background change

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 6:

Put the line numbers to whole page.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 7:

Translate the given paragraph into French language.

Please encircle or put the check mark on one of the given options to rate your expertise with this

Novice						Expert
1	2	3	4	5	6	7

task:

Task 8:

Insert a picture (from a file) after the first paragraph.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 9:

Insert a comment to any one word in the document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 10:

Increase the font size of the letters to 16.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 11:

Change the font of all the characters to "Kartika"

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 12:

Increase the number of columns to two.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

A.1 Experiment 1 – Instructions to Participants

1. First, I would like you to read and sign this consent form, and ask me if you have any questions about it. The consent form assures you that your data will be stored anonymously and securely, and that you can quit the experiment at any time if you are at all uncomfortable.

2. You need to complete 12 tasks in Microsoft Word 2003. You need to complete this set of tasks. As you complete each task, I will give you a description of next task. During performing each task, a help window will pop up to assist you to complete the task. Please click on “ok” button, if you wish otherwise click on “cancel” button. If you click “cancel” button, a questionnaire window will pop up. If you wish to take help then you need to enter the key word or name of currently doing task, in response you will see a help window with solution path. Soon after you click “ok” button of this window, a questionnaire window will pop up. A questionnaire window contains questions related to help window.

3. A help window will pop up after 1 minute and again after 3 minutes.

A.1 Experiment 1 – Post-study Questionnaire

1. Please write any comments about this experiment, especially what you felt about help suggestions.

2. Thank you for participating! Print and sign your name, fill in the date, and here's your $5.

A.1 Experiment 2 – Informed Consent Form

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF SASKATCHEWAN INFORMED CONSENT FORM

Research Project: **Task-dependent detection of user expertise**

Investigators: Dr. Regan Mandryk, Department of Computer Science (966-4888)

Mangalagouri Masarakal, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with automatically detecting the expertise of users with a specific software application. We will be using Microsoft Word for this study.

The goal of the research is to determine whether we can design an accurate model of expertise for automatic and dynamic detection.

The session will require 60 minutes, during which you will be asked to carry out a number of tasks in Microsoft Word. Tasks will each be repeated 3 times over the course of the experiment.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months).

This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Anonymity will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Office of Research Services, University of Saskatchewan, (306) 966-4053

Participant's signature:____

Date:_____________________

Investigator’s signature:____

Date: ______________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Office of Research Services at the University of Saskatchewan.

A.1 Experiment 2 – Demographic Survey

1 Rate your English knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice 1	2	3	4	5	6	Expert 7

2. Rate your Computer knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice 1	2	3	4	5	6	Expert 7

3. Rate your Microsoft Word knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice 1	2	3	4	5	6	Expert 7

After finishing the task for the first time:

Task 1: Cut, paste and replace

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 2: Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 3: Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 4: Increasing the line space of the whole document and the required line space is 1.5.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 5: Background change

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 6: Put the line numbers to whole page.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 7: Translate the given paragraph into French language.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 8: Insert a picture (from a file) after the first paragraph.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 9: Insert a comment to any one word in the document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 10: Increase the font size of the letters to 16.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 11: Increase the number of columns to two.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

After finishing the tasks for the second time:

Task 1: Cut, paste and replace

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 2: Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 3: Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 4: Increasing the line space of the whole document and the required line space is 1.5.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 5: Background change

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 6: Put the line numbers to whole page.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 7: Translate the given paragraph into French language.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 8: Insert a picture (from a file) after the first paragraph.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 9: Insert a comment to any one word in the document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 10: Increase the font size of the letters to 16.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 11: Increase the number of columns to two.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

After finishing the tasks for the third time:

Task 1: Cut, paste and replace

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	4	5	6	Expert 7

Task 2: Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 3: Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 4: Increasing the line space of the whole document and the required line space is 1.5.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	 4	 5	 6	Expert 7

Task 5: Background change

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 6: Put the line numbers to whole page.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 7: Translate the given paragraph into French language.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 8: Insert a picture (from a file) after the first paragraph.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 9: Insert a comment to any one word in the document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 10: Increase the font size of the letters to 16.

Please put the check mark in one of the bottom columns to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

Task 11: Increase the number of columns to two.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice						Expert
1	2	3	4	5	6	7

A.1 Experiment 2 – Instructions to Participants

1 First, I would like you to read and sign this consent form, and ask me if you have any questions about it. The consent form assures you that your data will be stored anonymously and securely, and that you can quit the experiment at any time if you are at all uncomfortable.

2 Now, please fill out this short demographic questionnaire. Ask if you have any questions. This questionnaire will ask your background knowledge about English, Computer usage, and Microsoft Word usage.

3 You need to complete 11 tasks in Microsoft Word 2003. You need to complete this set of tasks thrice but not like repeating a task for three times at one stretch. Instead, you need to complete all 11 tasks once then repeat this set of tasks in again two more times. After you complete each task, I will copy the logged data from AppMonitor and will save in a file. I will not help you to complete the task until after about 3 minutes.

4 Thank you for participating! Print and sign your name, fill in the date, and here's your $10.

A.1 Experiment 3 – Instructions to Participants

1 First, I would like to explain you the purpose of this task. This is a small experiment to rank animated visual icons which will be displayed as system tray icons in the right bottom corner of the screen.

2 You need to complete 8 tasks in Microsoft Word 2003. You need to complete this set of tasks for once. I will not help you to complete these tasks. While performing these tasks, if you notice any visual icon then please click on that and answer the questions related to it.

3 Your data will be stored anonymously and securely. You can quit the experiment at any time if you are at all uncomfortable.

4 Thank you for participating!

A.1 Experiment 3 – Post-study Questionnaire

1. Please rank the attentional draw of following visual signals which you came across while performing the tasks: Colour, Colour Fast, Grow, Grow Fast, Bounce, Bounce Fast.

Note: Lower rating means higher attentional draw

1.________________________

2. ________________________

3. ________________________

4. ________________________

5. ________________________

6. ________________________

A.1 Experiment 4- Instructions to Participants

1 First, I would like to tell you about this research project. Here, we provide help in Microsoft Word automatically by analyzing your performance with the currently doing task.

2 You need to complete 10 tasks in Microsoft Word 2003. As you complete each task, I will give you a description of next task. I will not help you to complete any of the given tasks. Your data will be stored anonymously and securely. You can quit the experiment at any time if you are at all uncomfortable.

3 A system tray icon appears at the right bottom corner of the screen. The icon will be different based on your knowledge with the current task. If you need help to complete the currently doing task and if you notice any visual icon in the right bottom screen then please click on that icon and you will get help. If you wish to take help then you need to enter the key word or name of currently doing task, in response you will see a help window with solution path.

4 Thank you for participating!!

A.1 Experiment 4 – Post-study Questionnaire

1 Please write your opinion about help suggestions or your opinion about this experiment.

APPENDIX B

BUILDING AN EXPERTISE-MODEL WITH MORE FEATURES

1 Informed Consent Form

2 Demographic Survey

3 Instructions to Participants

4 Expertise Model with More Features

5 List of Tasks

Appendix B – Informed Consent Form

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF SASKATCHEWAN INFORMED CONSENT FORM

Research Project: **Task-dependent detection of user expertise**

Investigators: Dr. Regan Mandryk, Department of Computer Science (966-4888)

Mangalagouri Masarakal, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with automatically detecting the expertise of users with a specific software application. We will be using Microsoft Word for this study.

The goal of the research is to determine whether we can design an accurate model of expertise for automatic and dynamic detection.

The session will require 60 minutes, during which you will be asked to carry out a number of tasks in Microsoft Word. Tasks will each be repeated 3 times over the course of the experiment.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Anonymity will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Office of Research Services, University of Saskatchewan, (306) 966-4053

Participant's signature:____

Date:____________________

Investigator's signature:____

Date: ______________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Office of Research Services at the University of Saskatchewan.

Appendix B – Demographic Survey

1 Rate your English knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice			Expert
1	2	3	4

2. Rate your Computer knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice			Expert
1	2	3	4

3. Rate your Microsoft Word knowledge

Please encircle or put the check mark on one of the given options to rate your expertise:

Novice			Expert
1	2	3	4

After finishing the task for the first time:

Task 1:

Please insert a foot note for any word.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 2:

Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 3:

Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 4:

Please change the background.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	Expert 4

Task 5:

Please rewrite this current sentence and your changes to this document should be tracked i.e. the changes you make should be tracked.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	Expert 4

Task 6:

Translate the first paragraph into French.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	2	3	Expert 4

Task 7:

Insert a comment to any one word in the document and write any comment.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 8:

Please include a table of index to this document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 9:

Add line numbers to the whole page.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 10:

Increase the number of columns to two.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	Expert 4

After finishing the tasks for the second time:

Task 1:

Please insert a foot note for any word.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 2:

Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 3:

Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 4:

Please change the background.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 5:

Please rewrite this current sentence and your changes to this document should be tracked i.e. the changes you make should be tracked.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 6:

Translate the first paragraph into French.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	Expert 4

Task 7:

Insert a comment to any one word in the document and write any comment.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 8:

Please include a table of index to this document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 9:

Add line numbers to the whole page.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 10:

Increase the number of columns to two.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	Expert 4

After finishing the tasks for the third time:

Task 1:

Please insert a foot note for any word.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	Expert 4

Task 2:

Insert hyperlink for a file.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	Expert 4

Task 3:

Justify all the paragraphs in center.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	2	3	Expert 4

Task 4:

Please change the background.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 5:

Please rewrite this current sentence and your changes to this document should be tracked i.e. the changes you make should be tracked.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 6:

Translate the first paragraph into French.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	Expert 4

Task 7:

Insert a comment to any one word in the document and write any comment.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 8:

Please include a table of index to this document.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 9:

Add line numbers to the whole page.

Please encircle or put the check mark on one of the given options to rate your expertise with this task:

Novice 1	 2	 3	Expert 4

Task 10:

Increase the number of columns to two.

Please put the check mark in one of the bottom columns to rate your expertise with this task.

Novice 1	 2	 3	Expert 4

Appendix B - Instructions to Participants

1. First, I would like you to read and sign this consent form, and ask me if you have any questions about it. The consent form assures you that your data will be stored anonymously and securely, and that you can quit the experiment at any time if you are at all uncomfortable.

2. Now, please fill out this short demographic questionnaire. Ask if you have any questions. This questionnaire will ask your background knowledge about English, Computer usage, and Microsoft Word usage.

3. You need to complete 10 tasks in Microsoft Word 2003. You need to complete this set of tasks thrice but not like repeating a task for three times at one stretch. Instead, you need to complete all 10 tasks once then repeat this set of tasks in again two more times. After you complete each task, I will copy the logged data from AppMonitor and will save in a file.

4. Thank you for participating! Print and sign your name, fill in the date, and here's your $10.

Appendix B - Expertise Model with More Features

Motivation

Although our expertise model could classify user expertise level as one of the four levels with an accuracy of 90% and most of the participants agreed that help suggestions appeared whenever they needed (from experiment four), it suffers from one main flaw. That is, in the model most of the instances are classified as expert. To overcome this problem, we conducted another experiment with a similar procedure as experiment two explained in Chapter Four. But, this time we extracted more features than in our second experiment. In the remainder of this Appendix we explain about experiment and results.

This model procedure is different from our previous model's procedure. Participants rated their expertise level as one of four levels instead of seven levels (as in our second experiment for building the model). In the previous model, we grouped seven levels into four while building an expertise model. To avoid this grouping in order to collect expertise levels directly from participants, we conducted this experiment with four levels.

Prior to building the previous model, six features were extracted from the collected data. Six features were as following:

1. Menu bar Count
2. Cancel Count
3. Dwell Time for Single Click
4. Dwell Time for Double Click
5. Help Count
6. Special Key Count.

Although six features were identified for building an expertise model, four features (Menu bar Count, Cancel Count, Dwell Time for Single Click, Special Key Count) were used in building an expertise model by WEKA-a machine learning tool.

While building an expertise model for second time, 11 features were extracted. They were as following:

1. SpecialKeysUsage
2. Help
3. Cancel
4. DwellTime
5. DwellTimeBetweenDoubleClick
6. MenubarChecking
7. Menubartime
8. SingleClickDeviation
9. DoubleClickDeviation
10. SelectionDepth
11. MeanSelectionDepth

Participants repeated 10 tasks for three times. Tasks were as listed in "Appendix B – List of Tasks". Expertise models were built using J48 algorithm in WEKA with many combinations of collected data (i.e. data collected while participants performed tasks for the first time, data collected while participants performed tasks for the second time, data collected while participants performed tasks for the third time, data collected while participants performed tasks for first and second time, data collected while participants performed tasks for second and third time, data collected while participants performed tasks for first and third time, data collected while participants performed tasks for first, second and third time).

<table>
<tr><th>Data Set</th><th>Algorithm Used</th><th>Accuracy</th><th>Confusion Matrix</th></tr>
<tr><td>First Data Set
(200 instances)</td><td>J48</td><td>30.5%</td><td>a b c d <-- classified as
0 0 0 54 | a = Novice
0 0 0 41 | b = Poor
0 0 0 44 | c = Average
0 0 0 61 | d = Expert</td></tr>
<tr><td>Second Data Set
(200 instances)</td><td>J48</td><td>53%</td><td>a b c d <-- classified as
0 0 0 16 | a = Novice
0 0 0 23 | b = Poor
0 0 0 55 | c = Average
0 0 0 106 | d = Expert</td></tr>
<tr><td>Third Data Set
(200 instances)</td><td>J48</td><td>75.5%</td><td>a b c d <-- classified as
0 0 0 5 | a = Novice
0 0 0 10 | b = Poor
0 0 0 34 | c = Average
0 0 0 151 | d = Expert</td></tr>
<tr><td>First and Second Data Set
(400 instances)</td><td>J48</td><td>41.75%</td><td>a b c d <-- classified as
0 0 0 70 | a = Novice
0 0 0 64 | b = Poor
0 0 0 99 | c = Average
0 0 0 167 | d = Expert</td></tr>
</table>

First and Third Data Set (400 instances)	J48	53%	a b c d <-- classified as 0 0 0 59 | a = Novice 0 0 0 51 | b = Poor 0 0 0 78 | c = Average 0 0 0 212 | d = Expert
Second and Third Data Set (400 instances)	J48	64.75%	a b c d <-- classified as 0 1 3 17 | a = Novice 0 1 1 31 | b = Poor 0 2 2 85 | c = Average 0 0 1 256 | d = Expert
First, Second and Third Data Set (600 instances)	J48	53.16%	a b c d <-- classified as 12 0 6 57 | a = Novice 9 0 5 60 | b = Poor 5 0 4 124 | c = Average 6 0 9 303 | d = Expert
Expertise Model with non-experts			
First Data Set (137 instances)	J48	41.61 %	a b c <-- classified as 32 13 8 | a = Novice 18 6 17 | b = Poor 17 7 19 | c = Average

Second Data Set (94 instances)	J48	59.57 %	a b c <-- classified as 2 0 14 | a = Novice 4 0 18 | b = Poor 0 2 54 | c = Average
Third Data Set (49 instances)	J48	65.31 %	a b c <-- classified as 0 1 4 | a = Novice 0 0 10 | b = Poor 1 1 32 | c = Average
First and Second Data Set (231instances)	J48	41.13%	a b c <-- classified as 21 7 41 | a = Novice 22 5 36 | b = Poor 22 8 69 | c = Average
First and Third Data Set (186 instances)	J48	46.24 %	a b c <-- classified as 24 8 26 | a = Novice 16 8 27 | b = Poor 10 13 54 | c = Average
Second and Third Data Set (143 instances)	J48	58.74%	a b c <-- classified as 0 0 21 | a = Novice 0 0 32 | b = Poor

			4 2 84 | c = Average
First, Second and Third (280 instances)	J48	50.36%	a b c <-- classified as 37 11 26 | a = Novice 24 6 43 | b = Poor 26 9 98 | c = Average

Appendix B – List of Tasks

Task 1:

Please insert a foot note for any word

Description: For example a foot note has been inserted for a word "any" in the below sentence: Please insert footnote to any word in the document.

Task 2:

Insert hyperlink for a file

Description: Please provide the hyperlink for the file: "Ex.txt"; this file is on the desktop. The hyperlink provided to a file looks like this: Ex.txt

Task 3:

Justify all the paragraphs in center alignment.

Task 4:

Background change

Description: Now, the background colour of Microsoft Word is white. Change the background colour of Microsoft Word to green.

Task 5:

Please rewrite this current sentence and your changes to this document should be tracked i.e. the changes you make should be tracked.

For example:this is a track change usage.

Task 6:

Translate the first paragraph into French.

Task 8:

Please include a table of index to this document

Task 9:

Add line numbers to the whole page.

Description: Add line numbers to whole page including the blank lines.

For example:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented. The word "Convocation" arises from the Latin "con" meaning "together" and "vocare" meaning "to call." The University of Saskatchewan's Convocation ceremony is a calling together of new graduates.

During the period between which you have completed the requirements to graduate and you are awarded your degree, you are called a "graduand." The University encourages all graduands to attend the Convocation ceremony. However, if you are not able to attend, you are still eligible to graduate and your parchment will be mailed to you. You must apply to graduate regardless of whether you plan to attend the Convocation ceremony. All undergraduate and graduate students who expect to graduate at either the spring or Fall Convocation must complete an Application to Graduate form. This form must be submitted by March 31 for Spring Convocation or by August 31 for Fall Convocation.

Students who have submitted an Application to Graduate will receive a Convocation package approximately one month prior to the appropriate Convocation ceremony regardless of whether or not they have met the requirements to graduate. It is each student's responsibility to ensure that they have met the requirements to graduate. College offices can confirm that students have met the requirements for their degrees.

Only students who have met the requirements to graduate may participate in the Convocation ceremony. The Convocation ceremonies are held on three days at TCU Place (formerly the Centennial Auditorium). Graduands should carefully note the date and time of the ceremony at which they will receive their degree. Graduands and guests will be admitted to the Convocation ceremony by ticket only.

The order of events for the ceremony will be outlined in the programs that will be placed on the seats in the Auditorium. Graduands will be led in a procession and guided to their seats. They should remain standing while the platform party enters, during the singing of

Task 10:

Increase the number of columns to two.

Description: For example:

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies. Convocation is the ceremony at which the degree or certificate is publicly presented.

Graduation is the awarding of a degree or certificate following the satisfactory completion of a student's program of studies.

Convocation is the ceremony at which the degree or certificate is publicly presented.