# ABSTRACT

LEE, LASHANDA ELAINE. Assessing Interactive System Effectiveness with Usability Design Heuristics and Markov Models of User Behavior. (Under the direction of David Kaber.)

The purpose of this research was to create a new measure of usability to aid researchers in determining whether the intent of system designers is realized and to objectively assess user behavior as a basis for interface design recommendations. Contemporary human-computer interaction (HCI) research has focused on methods to promote usability in interactive systems. Unfortunately, there are few quantitative, objective measures of usability, among many subjective measures. A measure that can combine both objective and subjective data may be valuable in determining user perceptions of system designs and whether performance goals are achieved.

The present research utilized an existing HCI framework developed by Abowd and Beale and a mathematical model of the average number of user actions at an interface (e.g., mouse clicks) required for task performance in order to determine an overall system usability effectiveness score. Part of the score involves usability ratings by users, while considering designer interests in interface development. The components of the Abowd and Beale framework include the user, input, the system and output, which are interconnected by links that affect system effectiveness. Some links represent computational load, while others represent the complexity of articulation and system state observation for the user; that is, cognitive resources expended in performance.

Four designers were asked to rank the importance of each link in the framework, based on their design intentions for the two online ordering interfaces. Pair-wise comparisons were made of the various dimensions of system effectiveness and usability represented in the Abowd

and Beale framework, and the rankings were averaged across designers. Designers were expected to consider user articulation and observation as the most important aspects of the interface design because they represent cognitive load.

Twenty users, divided into two equal groups, were asked to complete the task of buying a certain type of computer (ThinkPad R60) using an existing and new prototype web interface. The new prototype was designed to increase usability with consolidated pages, more pronounced buttons and a multi-level menu structure. The users then considered the links in the Abowd and Beale framework and rated the specific design alternatives to provide a subjective assessment of whether the designer's intent was achieved. It was expected that the new interface would receive higher average ratings than the old interface, based on the design changes.

Because subjective analysis of an interface is typically insufficient for identifying all serious usability problems, and for justifying design changes from a management perspective, it was necessary to include an objective component in the system usability effectiveness score. In web applications, the average number of clicks can be used as a measure of system performance efficiency. Markov Chain models can be used to predict human motor behaviors at an interface, such as the average number of mouse clicks in a task, based on small samples of actual performance data. In the present study, on-line interface state transitions elected by users were recorded by a java script and used to establish probabilities for each system state. The probabilities were included in a Markov model to predict the average number of clicks in task performance.

Once the average number of clicks was predicted, the subjective usability ratings were divided by the Markov model output to determine a ratio of 'usability per interface action' for the system (i.e., the overall effectiveness score). The interface alternative resulting in the higher

score (ratio of usability per interface action) can be said to have higher usability and represent a better match to the intent of the designer. The actual number of user clicks at the two web interfaces was also recorded during the study. It was expected that the new interface would reduce the number of clicks necessary to optimally reach the computer purchase goal; therefore, increasing the overall system effectiveness score. It was also expected that using Markov models would accurately predict the average number of clicks.

It was found that the overall effectiveness score was, in fact, higher for the new interface, suggesting the intent of the designers was achieved and the design revision provided better usability for ordering a computer. The Markov model was also proven to accurately predict the average number of clicks. The current research developed a usability evaluation method that incorporates both subjective and objective data, which may be more effective than prior measures focusing on subjective usability ratings only. The new measure may also reduce the need for user testing experimentation, in comparison to objective usability measurement approaches.

# Assessing Interactive System Effectiveness with Usability Design Heuristic and Markov Models of User Behavior

by
Lashanda Elaine Lee

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Industrial and System Engineering

Raleigh, North Carolina

2007

APPROVED BY:

| | |
|---|---|
| _____ | _____ |
| Dr. Gary Mirka | Dr. Robert St. Amant |
| Committee Member | Committee Member |

_____

Dr. David Kaber
Chair of Advisory Committee

## DEDICATION

To my beloved, Ronald H. Hodge II and to my family, who has supported me through everything.

## BIOGRAPHY

Lashanda Lee was born on June 11, 1983, in Raleigh, North Carolina, were she also grew up. When she was a child, her parent's often encouraged here to be involved in many activities and that academics were the most important focus. She was heavily involved in volunteer work at a young age. She participated in girl scouts and often worked at the Food Bank of North Carolina. She was especially interested in science and math and her parent's encouraged those interests, driving her to achieve important goals. She graduated from high school knowing she wanted to be an engineer, largely based on her involvement in many programs such as the MSN pre-college program, DECA, a marketing club, and national honor society. She earned her Bachelor of Science degree in Industrial Engineering at North Carolina Agricultural and Technical State University, where she also participated in many activities. Lashanda was the secretary and vice president of Institute of Industrial Engineers and the vice president of Alpha Pi Mu, the industrial engineering honor society. She coordinated a mentoring program for new students in the Industrial Engineering department and established a tutoring relationship with the Welfare Reform Liaison and Black Child Development. Lashanda was also the recipient of many scholarships, such as the Medtronic scholarship, Dean's scholarship and the Tom Joyner scholarship. Her interest grew in the field of human-computer interaction after taking several required classes in the area and obtaining a Human Factors Internship at the SAS Institute for two summers. Currently, she is working to complete her Masters of Science degree in Industrial Engineering at North Carolina State University, with the support of a NIOSH training grant and hopes to continue towards a Doctor of Philosophy degree in Industrial Engineering.

## ACKNOWLEDGEMENTS

I would like to acknowledge my family and friends who have supported me. I would also like to thank the NC OSHERC program for their financial support through my Master of Science degree. I would especially like to thank my committee and professors, in particular, Dr. Kaber, my advisor, for his great guidance and help during my career as a graduate student.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## 1.0 MOTIVATION

Humans and computers are complex biological and mechanical systems that differ in terms of methods of communication and the ways in which they analyze tasks. In order for human computer interaction (HCI) to be successful, interfaces must be designed to effectively translate the intentions, actions and inputs of the operator for the computer, and translate machine outputs effectively for human comprehension. In HCI applications, data translations like these often fail because of interface usability problems. For this reason, contemporary HCI research has focused on methods to promote usability in interactive systems. Unfortunately, from a usability engineering perspective, there are few quantitative measures of usability as basis for design, among many subjective measures.

To promote effective HCI, usability needs to be considered throughout the product or system design process. Currently, many interface development processes do not incorporate usability principles creating issues for users, which are often addressed through costly retrofitting of systems after production. Usability may not be considered in design because companies do not understand how it can lead to profits and software engineers may be unaware of low cost methods that can be used to ensure usability. Even if a company knows of usability methods, one common perception is that there is a high degree of complexity in creating objective tests to guarantee usability (Gould, 1988). For this reason, in many software development companies, usability analysts may simply be asked to review interface designs for usability when a product or system is near the end of production. When it is determined that there are usability problems, which need to be addressed for performance or customer preference, it can be difficult for a programmer to reconstruct system logic to incorporate new design changes on the basis of

subjective analyst input. However, with objective usability analysis methods, specific design changes may be identified that can lead to compatibility of user input and system states. Such measures can also decrease the problems programmers have in addressing changes recommended by usability analysts.

In recent years, several comprehensive interactive system design frameworks have been developed, such as Abowd and Beale (1991) and Norman (1988), as bases for guiding HCI application development from a usability perspective. Such frameworks can be useful for classifying usability problems and identifying aspects of system designs that are considered critical to usability and effectiveness. Norman's model describes cycles of execution and evaluation in HCI and focuses on the objectives and actions of the user, as well as system responses. Based on Norman's model, Abowd and Beale created an analogous interactive framework that includes four major components, which focuses more on the role of the interface and machine in the overall interaction framework. Studying this research can help researchers develop approaches to classify, assess and direct re-design of interactive systems.

In order to facilitate robust and accepted usability analyses, a simple framework like Abowd and Beale, can be used to organize and structure a design problem. It may also allow for criteria to be established providing the usability analyst with direction in design. Usability analysts may also apply Operations Research techniques to provide quantitative information as a basis for design. With such a combined approach, the usability analyst may be able to qualitatively and quantitatively compare various design alternatives along several dimensions of usability (e.g. perceived importance of usability issues for designers and number of interface actions) and create meaningful outputs for a system's design process.

## 2.0 LITERATURE REVIEW

### *2.1 HCI Frameworks*

HCI frameworks describe interaction in complex systems between the human and computer. Newman et al. (1995) described the manner in which humans process information and communicate in a HCI application. The authors suggested human information processing involves multiple processors (cognitive, perceptual, and motor). Each processor is like a subsystem, which must be available for perception of auditory and visual information and vocal communication, etc. Newman et al. stated each processor is associated with separate memory stores. Similarly, the computer possesses different components that are metaphorically named after the components of the human brain, such as a processor and memory, and these processors function in a conceptually similar manner. In effective HCI, the processor and memory capabilities of the human and the computer should complement each other. The methods of communication, including verbal and visual, should also be made compatible through the system interface.

There are several models, like Newman et al. that attempt to describe the methods of communication between the human and the computer. The purpose of modeling the interaction between the human and the computer is to provide a basis for explaining how and why interactions between the two may fail, leading to errors. The purpose of this section is to describe in detail the most common interaction models and to identify a model that may be robust and concise in terms of describing the communication between the human and computer in interactive system applications. The models reviewed include Norman's model of HCI,

Abowd and Beale's interaction framework (an extension of Norman's model) and the Pipe-Line information processing model (Mulhem and Nigay, 1996).

Norman's model is based on a cycle of interaction that contains two major phases. The first phase, execution, consists of four separate steps. The user must first establish a goal. Next, the user forms the intention, followed by deciding on an action sequence to reach the goal. Lastly, the user executes the action sequence. The second phase, evaluation, consists of three steps. After the user has executed the action to reach the goal, the user then perceives the system state. Next, the user interprets the state of the system and, finally, he or she evaluates the system state based on the original goals and intentions (see Figure 1).

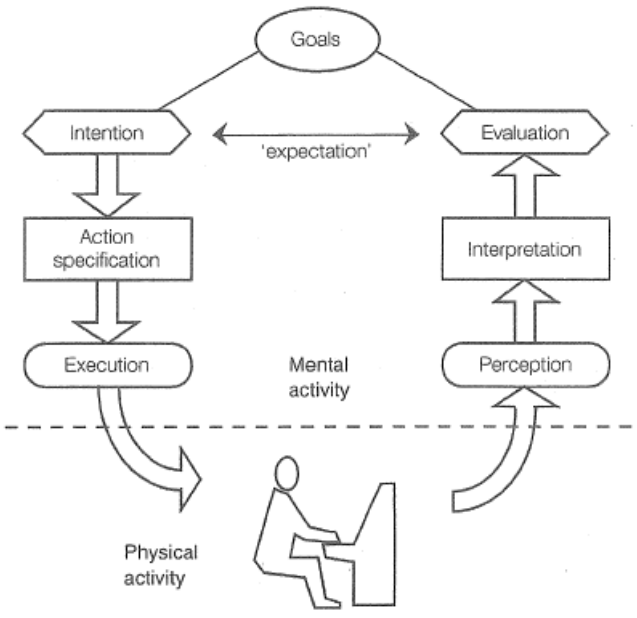



Figure 1: Norman's (1998) model of human-computer interaction (from Newman and Lemming (1995)).

In order to evaluate the effectiveness of the communication between the human and the computer, Norman proposed that there are gulfs that separate the user's actions and goals. The gulf of execution is considered the difference in the intention of the user's action to reach the goal and the allowable actions of the system. For effective interaction (usability), the interface design for the system should try to reduce this gulf. The gulf of evaluation is the figurative distance between the display of the system state and what the user expects the interface to display. Therefore, to reduce this gulf, the output should be presented in terms of the expectations of the user. Norman's model describes HCI primarily in terms of the objectives and actions of the user; however, it only considers the system as an interface and does not focus on the continual (parallel streams of) communication between the human and the computer.

In Mulhem and Nigay's (1996) Pipe-Line model of human information processing in complex systems, human goals and mental processes represent one level of abstraction (encompassing the stages in Norman's model). The Pipe-Line model (see Figure 2) shows two pipes that represent the output of the system interface and the human input to the interface. They operate in parallel. At the beginning of the pipes, there are physical actions that can be connected to states of the system through a link from system input to the output. This captures the parallel nature of HCI, unlike Norman's model. The second stage in the model is the information input unit. Third, the system formulates an action in response to the input and it generates output. These events are also connected to the output interface through small pipes, representative of immediate feedback. After the system action formulation, there is an adapter to translate the user action to the system language. The command is sent to a matching node where

it can be related to an existing database of relevant commands for execution. The system output then passes through another adapter for translation back to the user's language.

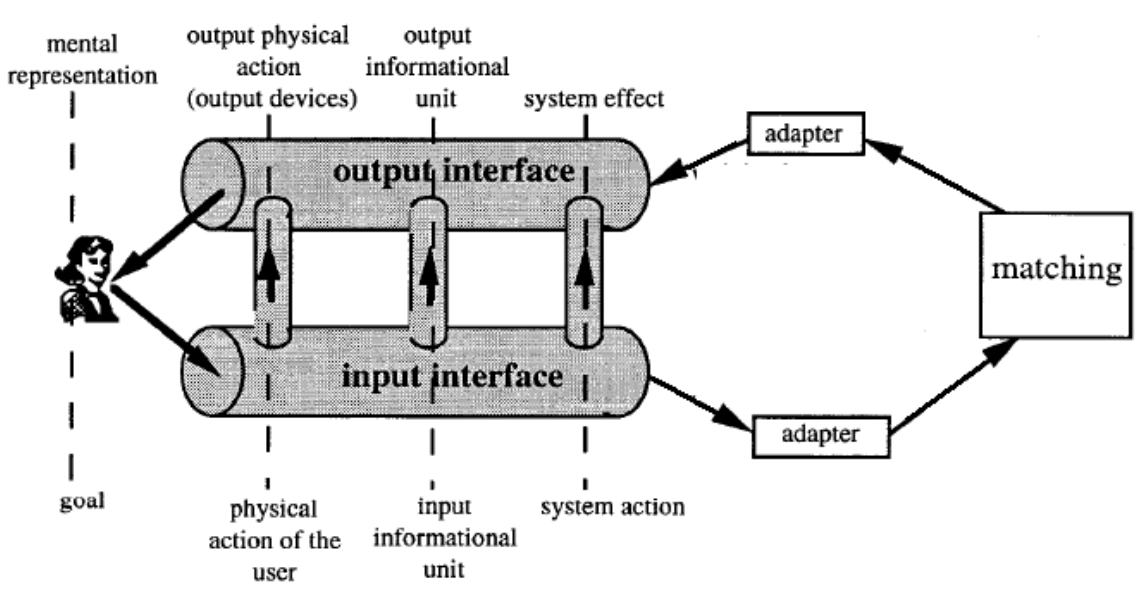



Figure 2: Pipe-Line model

Because the Pipe-Line model is relatively complicated it is best to describe it using an example. In research done by Mulhem and Nigay (1996), a query system was used as an example of the model. First, the user types a command, which is considered the physical action. Then, the system identifies the information input units (the information typed) entered by the user. Next, the user hits the enter key and the system determines an action to be preformed. The system then takes the information and adapts it into computer language, matches it to a command database, and restores it to a language the user can understand. Next, the query results are ranked for the user and presented based on relevance. This is considered the output information unit. The user can then pick the result that best fits their needs. The Pipe-Line model is somewhat similar to Norman's model of interaction but it presents greater detail of how the

system works to translate user commands and to retrieve data for the user. It also represents the highly parallel nature of input and output in HCI.

Abowd and Beale's model of interaction is an extension of Norman's model. Because the interface lies between the user and the system, an interactive cycle was defined to reflect this physical arrangement. The interaction cycle contains four distinct steps, as opposed to the two represented in Norman's model, and it is less complex than the intricate pattern of interaction presented through the Pipe-Line model. According to Abowd and Beale (1991), the first step begins with the user's formulation of goals, where the goal is articulated into input language (see Figure 3). Next, the input is translated to the system language. This stage is not usually considered in usability evaluation because is represents the time and effort by the system designers and programmers for accommodating user psychological intentions. The step, however, should be considered in any model of HCI, because when trying to create usable designs the capabilities of the programmers may be limited, resulting in design ideas that cannot be implemented or that do not account for user needs. Next, the system presents the output on the interface. Lastly, the user observes and evaluates the display.

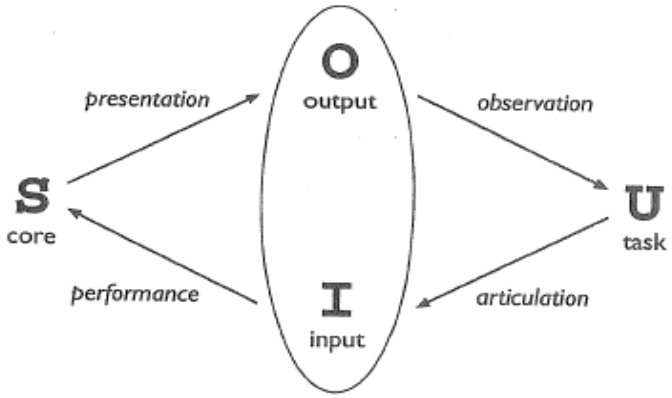



Figure 3: Abowd and Beale et al (1991) model of interaction

The Abowd and Beale interaction framework is robust in representing the stages of HCI yet more parsimonious than Norman's model. It is also not as complex to understand as the Pipe-Line interaction model. Abowd and Beale's model of interaction represents the entire cycle of communication between the human and the computer. It is a simple model that can be used to describe the actions that take place during communication. It details the continual cycle of communication between the human and the computer, which is important in describing how people use computers. The representation of the system interface also captures the connection of input and output in a parallel way. The Pipe-Line model is very accurate in describing the interaction cycle; however, it is complex and focuses mainly on how the system processes information sent by the user. The Abowd and Beale model, allows for easier interpretation in terms of identifying communication failures. Based on the attributes of Abowd and Beale's model, the framework was used in this research as a basis for defining a new measure of system usability and effectiveness to improve design and prevent various types of user errors. This measure was combined with another objective usability evaluation.

### *2.2 Usability Paradigms and Principles*

There are several historical principles of usability that are evolving into new usability paradigms each day. Usability paradigms reveal how humans interact with computers in contemporary applications, while usability principles describe how these paradigms work. Understanding usability paradigms and principles allows for researchers to change and enhance existing technologies and make them better, leading to a better interaction and performance. Usability measures can be used to select among paradigms in a design process to promote system

effectiveness from a user needs perspective. Many historical usability paradigms are still used today but a plethora of new paradigms have begun to surface as technology has become more intricate. Some historical paradigms include command line interfaces and large mainframe computers that were shared by groups of users. Newer paradigms include collaborative systems, ubiquitous computing, intelligent systems, virtual reality and more.

The purpose of this section is to describe several paradigms and principles in use today, describe some usability tradeoffs associated with each, and show how these paradigms and principles can minimize confusion in the communication between the human and the computer. With each type of paradigm, different usability measures can be used to identify acceptable levels of performance and user satisfaction and to address different usability principles. Some usability principles include error prevention, flexibility, portability, consistency, robustness, parsimonious, recoverability, learnability and efficiency. Flexibility describes the many ways the computer and user exchange information. Learnability is defined by Dix et al. (1993) as how easy it is for a user to begin effective interaction and perform optimally. Robustness represents how easy it is to apply the interface to a range of applications. How closely an interface resembles other system interfaces and how closely the output and input match a user's expected input and output is considered consistency. Efficiency describes how effective the interface is in achieving the user's goals. One important research need in this area is to relate specific types of usability measures and outcomes to the use of certain paradigms for performance (i.e., what measures are best for evaluating the usability of a particular design approach).

### 2.2.1 Collaborative systems

According to Rosson and Carroll (2002), collaborative systems are used for several applications, where groups of people need to collaborate remotely or locally and in real time or at different times. Some examples of collaborative systems include video conferencing, email, list-serves, shared file systems and intranet, as well as digital whiteboards. Collaborative systems have several tradeoffs. In some systems, nonverbal communication cues are missing (e.g., user hand gestures and facial expressions); however, addressing such limitations through addition of interfaces can cause the display to become cluttered and distract users in comparison to other desktop applications. Other collaborative system designs can lead to reduced interpersonal communication, which may promote awkward or risky discussion, not normally present in face-to-face conversation. Collaborative systems, such as email, can store archived data, which can help manage group discussions, but the cost of archiving is usually paid by persons who do not benefit directly from the systems.

Collaborative systems are used to achieve certain usability principles and a specific set of usability measures can be used to evaluate this type of system. Rosson and Carroll (2002) suggested collaborative systems increase the flexibility of work and efficiency. These systems are considered flexible because information and resources are shared more effectively and each group member can create a customized work setting (e.g., a worker in another country can participate from home). Collaborative systems increase efficiency because they allow several users to create and enhance an idea while they are not all currently in the same location. To evaluate these types of systems, usability measures such as heuristic methods, guidelines, style and rule inspections and walkthroughs should be used versus empirical evaluations that consider,

for example, task completion time, time in mode and time until an event. In order to ensure that these systems provide efficiency and they are used to effectively transfer ideas and resources, the time that it takes a person to complete a task is important but may be difficult to test. However, issues such as inconsistency in design, etc. should be eliminated through inspection methods, which can be implemented easily. Collaborative systems are not successful in regards to robustness because this type of interface cannot be applied to certain types of displays, for example, an online ordering system for use by independent workers.

### 2.2.2 Ubiquitous computing

Ubiquitous computing is a term first used by Mark Weiser in 1991 to represent a future where computers are incorporated into our everyday tasks. Some other terms synonymous with ubiquitous computing are pervasive computing, calm technology and "everyware". Examples of ubiquitous computing include: PDAs, pocket PCs, and mobile phones, which enable mobile computing. With ubiquitous computing there are also usability tradeoffs. According to Rosson and Carroll (2002), when using these devices flexibility and portability are increased; however, complex tasks cannot be supported because of the limited display space and memory in small devices. Because ubiquitous computing devices are small and easy to carry, they are considered portable. Yet there is often a problem of creating menus in small ubiquitous computing devices that are easy to navigate.

Ubiquitous computing is used to achieve certain usability principles and certain usability measures can be used to evaluate these types of systems. Rosson and Carroll (2002) also considered ubiquitous computing devices to increase efficiency. Users can email and surf the Internet wherever they go, allowing for efficient work without being in an office, thereby

increasing overall efficiency. Usability measures such as verbal reports, guidelines, and walkthroughs should be used in advance of empirical evaluations of task completion time, average number of clicks, etc. to evaluate these types of systems. Measures such as guidelines should be used to eliminate initial problems with consistency and functionality. However, collection of empirical data should also be considered. The time required to complete tasks is important in gauging how efficiently a user can use the device. The number of clicks (interface actions) can help evaluate the structure of a menu system in a PDA, where fewer clicks to reach a desired option means better usability for the user. Ubiquitous computing is not effective in recoverability because it is often hard to show system states and paths out of menus with such a small interface display.

### 2.2.3 Intelligent systems

The goal of an intelligent system is to collect and organize enough information about a user, task or situation, to create an accurate prediction of what the user needs and wants to see or do. Intelligent systems learn during their existence. They receive information from the environment in order to adapt and make correct decisions for supporting users. Intelligent systems can be considered "stimuli-response" systems. According to Rosson and Carroll (2002), two types of intelligent systems are software agents and natural language interactions. Using natural language processing (NLP), a system should be able to learn and respond to a user's language. However, natural language can be ambiguous and is less precise in computing. Natural language simplifies the interpretation for the user but may be very hard for the system to understand because of its ambiguity. Consequently, high computational processing loads and system speed are the prices for usability in NLP systems. Software agents take data from

specific applications and process the data further for ease of use. Software agents can be compared to personal computing assistants (e.g., Microsoft "paperclip"). However, this technology may reduce the feeling of control for the user and may not exhibit the type of etiquette humans have come to expect in human-human interaction.

Intelligent systems are used to achieve usability principles and a specific set of usability measures can be used to evaluate this type of system. The usability principles addressed by intelligent systems are efficiency and system parsimony. NLP systems do not require any thought for the users, the user can simply ask a question in their own language and the computer should understand and present the desired information to the user in their language. These types of systems promote efficiency because they eliminate the time a user would spend to interpret system feedback and the time it takes to input information into the system in a manner that the machine can understand. As stated by Rosson and Carroll (2002), the user does not have to interpret information or system feedback. This allows a user to complete more work in a shorter period of time therefore promoting efficiency. Usability measures useful in evaluating these types of systems include inspection methods (heuristic methods and guidelines) applied before collecting empirical data, such as number of correct tasks completed and task completion time. Guidelines, etc. should be used to eliminate initial and obvious usability problems. Schneiderman (1998) recommended one guideline that should be considered with NLP systems; specifically designers should avoid overly personal dialogs to prevent novices from developing unrealistic levels of trust in a system. Cognitive walkthroughs may be less appropriate because there are several dialogs that may develop using NLP systems and CW can only cover a limited portion of the application, which is insufficient in order to create design recommendations.

Because these systems should promote efficiency, task completion time is helpful in evaluating usability. Because NLP and intelligent systems are considered parsimonious they should be simple to use in completing all tasks given to the user. By recording the number of tasks completed correctly, it is easy to determine how well this paradigm promotes parsimony. These systems, however, are not effective in portability because the programming and memory capabilities that are required to create the technology cannot be compressed into a small computing device.

### 2.2.4 Virtual reality

Virtual reality (VR) is a technology that allows a user to interact with a computer-simulated environment. An issue in designing virtual environments is creating the appropriate degree of veridicality, or achieving a high degree of correspondence with the real world. Some example uses of VR are skill training, behavioral therapy, and design. Through the use of standard input devices, such as a keyboard and mouse, or through multimodal devices, such as a data glove (e.g., the Cyberglove), users can interact with a virtual environment. According to Rosson and Carroll (2002), VR and simulation can help ease learning of complex systems and allow for forms of unnatural interaction, such as extending an arm longer than naturally possible to reach an object in an environment, to more effectively perform tasks. Immersive virtual VR environments can also enhance user situation awareness on displayed information.

However, virtual environments can also cause fatigue and motion sickness for users. In practice, it is currently very difficult to create a high-fidelity VR experience, due largely to technical limitations in computer processing power, image resolution and communication bandwidth. However, those limitations are expected to eventually be overcome as processor,

imaging and data communication technologies become more powerful and cost-effective. Even though virtual reality can be used for several applications, in order to provide veridicality several objects may need to be placed in the VR space. With this condition, the work area may become cluttered making it hard for users to focus, leading to a short attention span (Roussou, 2000).

Virtual reality systems are used to achieve usability principles and a certain set of usability measures can be used to evaluate such systems. The usability principles supported by VR systems are efficiency, flexibility and learnibility. When a person uses VR systems, they can perform tasks in several different environments that may occur in the real world. The capability of using such systems to achieve goals in several environments provides flexibility and efficiency. Because VR environments mimic real world environments, users can use mental models from past experience to learn how to use a system easily. Usability measures that are effective in evaluating these types of systems include guidelines, heuristic methods and walkthroughs. More importantly, quantitative measurement data, such as error rates and task completion time should be recorded because VR is typically used with specific tasks in mind. Task completion time is a good measure of efficiency and error rates can determine how fast a user can learn to use a system. VR systems, however, are not typically effective in recoverability because error correction procedures are difficult to model.

### 2.2.5 WIMP interfaces

WIMP (Windows, Icons, Menus and Pointing device) interfaces are often used in desktop computing applications and are easy to learn and implement. Point and click interfaces are easy to manipulate and are often combined with menu systems to create the WIMP interface. WIMP interfaces have several standard elements that include visual controls designed based on

metaphors to real world systems. Icons in such interfaces represent actions and feedback mechanisms. Menus and controls include dropdown menus, popup menus, radio buttons, checkboxes, selection lists and dialog boxes. Some examples of pointing devices are the mouse, which is the most common, the trackball, the trackpad and pens. Metaphors are design tools used to help the user associate the actions of the interface with something that is already commonly used in other real-world tasks. Each element in a WIMP interface is easy to learn; however, maneuvering through several of these can increase the complexity of the interface (Biström et al, 2005).

WIMP interfaces are used to achieve usability principles and a certain set of usability measures can be used to evaluate such systems. The usability principles followed by WIMP interfaces include efficiency, ease of learning and recoverability from errors, robustness, and customizability (Biström et al, 2005). WIMP interfaces can also be considered portable, flexible, and parsimonious. The WIMP interface may be easy to learn because they map what a user is doing in real space to the display space. Within WIMP interfaces, programs allow "undo" and "back" functions in order to provide for quick error recovery. These interfaces are robust because they can be used across several applications. A WIMP interface can exist within a ubiquitous computing device and collaborative systems. For this reason, these interfaces are also considered portable. In many programs delivered through a WIMP interface, it is possible to customize the types of options and the way they are presented. WIMP interfaces are also efficient in completing several tasks simultaneously by providing multiple windows.

Several usability measurement techniques can be used to evaluate WIMP interfaces. All of the usability inspection methods (heuristic methods, guidelines, etc.) can be used to assess

usability prior to collecting empirical data for quantitative usability analysis. Empirical data that is helpful in determining the effectiveness of the WIMP interface include task completion time, time in mode, and time until an event, input rate (e.g., words per minute), mental effort (assessed using tools like NASA's Task Load Index (Hart and Staveland, 1988)), number of correct tasks completed and average number of clicks and task performance accuracy (error rates, spatial accuracy and precision).

### 2.2.6 3D interfaces

3D interfaces are also used as part of WIMP interfaces, including buttons, scroll bars, and icons. These elements are made to appear as if they have three dimensions. In order to create a 3D look, shading is used, for example, to imitate an object crafted in stone. According to Dix et al. (1993), when designers use this feature sparingly, the 3D objects can be used to identify active objects and areas, which help with decision making and understanding information. 3D features can also be used with workspaces, which are much more complex. Size, light, and occlusion can create a visual allusion of distance in a workspace. When an object in the interface moves further away, it also appears much smaller and takes up less space in the 3D workspace. 3D interfaces allow users to do tasks in a quasi real-world simulation. Disadvantages of 3D interfaces include: (1) for novice users, it may be hard to tell where objects are placed in the 3D environment; and (2) designers often use 3D technology for every feature in the interface, diluting its use as an indication for activation. This type of interface addresses the same usability principles as the WIMP interface and is conducive to evaluation with the same usability measures used to evaluate the WIMP interface.

### 2.2.7 Summary

In the previous sections, several of the most common usability paradigms and principles were presented. All of these paradigms can be used to address the figurative distance of articulation in Abowd and Beale's model and to improve performance between humans and computing systems. Each paradigm can be evaluated by several usability measures and each is generally considered effective at addressing certain usability principles. For example, intelligent interfaces that use NLP may provide for the greatest reduction in articulation distance for users, but they represent alternatives furthest from the system language, in design. It takes a long time for a system to translate the information from a NLP interface into a command language. On the other hand, a command line interface represents a design alternative closest to the system language, without requiring users to use binary terms. This type of interface is also the farthest away from the user's language in the Abowd and Beale interaction framework. With command line interfaces the computer does not have to interpret the information it has received from the interface. However, the user has to recall the specific language and know which commands to use to generate certain system actions. The WIMP interface is a design alternative that is basically equal in distance between the user and the system in the Abowd and Beale interaction framework. It is easy for both the system and user to interpret, respond to, and choose the next action. In this research, new integrated usability evaluation measures were developed and applied to WIMP interface design alternatives for an online ordering system.

## *2.3 Measures of Usability*

There are several methods used to measure and evaluate usability of existing interfaces and to design interfaces from scratch. Measures of usability can be generally grouped into two

categories, quantitative and qualitative. These types of measures are used in certain instances in the design or evaluation phases of an interface and offer both pros and cons for assessing usability. Quantitative measures can be divided further into objective and subjective measures. Subjective measures are based on opinions (e.g., survey ratings) whereas objective measures are collected empirically (e.g., task completion time) and are based on observations. The purpose of this section is to describe quantitative and qualitative measures of usability and discuss how each measure is applied to HCI, as well as describe the pros and the cons of the measurement techniques.

### 2.3.1 Qualitative Measures

Qualitative measures make comparisons of designs based on interface qualities that cannot be quantified and statistical analysis typically cannot be used to validate the effectiveness of usability. Some qualitative measures in use today include inspection-based methods, verbal reports, and surveys and questionnaires. The last two measures are usually obtained from user interviews. An example of an inspection-based method is the use of guidelines. A group of experts will review an interface attempting to assess all aspects and determine if the interface meets a set of guidelines (e.g., all delete icons should be “blue” for consistency) created by the design team or a standards organization. Verbal reports are collected as a user completes a set of tasks by having the user “think aloud” during performance. This method is used to determine problems that a user may face as he or she completes real-world tasks and to consider the user’s thought process. Surveys and questionnaires are used after a user has completed a test with an interface to help associate feelings about the interface with empirical data collected, as the user

was completing tasks. They can also be used to gain subjective opinions and suggestions on how to make the interface more useful to its customers.

Each of these methods has several benefits, such as low cost and quick discovery of design problems, and is discussed in detail below. It has been suggested that inspection based methods are a good complement to user testing (Clamann and Kaber, 2004). They can help eliminate usability problems before testing begins and reduce analysis and reporting time. Inspection methods are also considered low cost, low skill methods. According to Cockton et al. (2002), there are five specific types of inspection methods that are used today: principle-based methods, heuristic methods, guidelines, style and rule inspections, and walkthroughs. Heuristic evaluations are completed by usability experts that determine strengths and weaknesses of a system by taking on the user's perspective as they analyze the system components and the interface. Research has shown heuristic evaluations methods to be thorough in correctly identifying problems; however, the problems that heuristic evaluations tend to miss are considered severe (Cockton et al., 2002; Molich and Nielsen, 1990). Previous research also suggests that because heuristic evaluations are typically performed by experts, they may not reflect the user's perspective (Mosqueira-Rey et al., 2004; Chang and Dillon, 2006). Cognitive walkthroughs methodically cover the application space and procedures and provide a good resource for candidate problem discovery. However, CW methods may not find all problems that affect the user as he or she completes a work related task because the analysis typically focuses on a single action sequence.

Even though inspection based methods may be low in cost and generate fast discovery times, they often fail to find many serious problems. Furthermore, many inspection methods do

not provide enough evidence to create design recommendations and invalid predictions often prevent formulation of firm design or redesign recommendations. According to Rosson and Carroll (2002), analytic summative evaluations are also considered as an inspection method. Summative evaluations are done at the end of the design process to measure the quality of a product and answer questions such as "does the system meet its specified goals?" Analytic summative evaluations produce interpretations and are only observations of the system to determine if usability guidelines were followed (Rosson and Carroll, 2002). They also suffer from user response bias and recall issues because they are completed after task performance.

According to John and Marks (1997), Usability Evaluation Methods (UEM) are not as effective as researchers predict. John and Marks created an effectiveness tree, which is a hierarchical structure, detailing how effective each usability evaluation method is based on its predictive power, persuasive power, and potential for design change effectiveness. The predictive power is considered the amount of problems a UEM can detect before a design is tested empirically. The persuasive power is whether a developer decides to change an aspect of the system based on the evaluation. The design change effectiveness determines if there are more, the same number, or fewer problems observed by the user as a result of actually changing parts of the system. In an experiment by John and Marks, 6 analysts used six different UEMS to analyze a Volume View Builder. The Volume View Builder is a multi-media authoring tool. The analysts were to use claims analysis, cognitive walkthroughs, GOMS, heuristic evaluations, user action notation, and reading of the specification repeatedly (the baseline condition) for assessing usability. Claims analysis is analyzing the relationship of design parameters to the usability of an interface. In claims analysis, artifacts of the user interface are listed along with

their design features and all relevant positive and negative implications of each design feature are listed. This approach can help to select among alternative designs, and clarifies the questions to be analyzed through user testing by stating how the design should work as a set of well-defined claims. Over the course of ten weeks, the analyst analyzed the system using their given UEM. The resulting lists of problems were given to a developer and he decided which problems were real usability problems that he wanted to change, which was considered the persuasive power metric. Given the decisions made by the developer, cognitive walkthroughs, the GOMS model, and reading produced the fewest non-usability problems. After the developer fixed the problems he felt were worth fixing, the researchers then conducted a usability test. The problems that were not fixed were monitored. After the test was completed the results were put into the effectiveness tree. Most of the problems from the walkthrough and heuristic evaluations were fixed. About half of those problems resulted in fewer problems when users used the new system. However, the other half either resulted in the same problems or more problems. Because of these results it is probably best to use UEM methods in conjunction with other methods.

Verbal reports are sometimes considered limited in utility because analysts may require specific background to conduct an evaluation, but often this is not the case (Karat, 1997). Sometimes evaluation procedures may also contain leading questions, which can generate false results. Beyond this, the information gained during a “think aloud” (verbal reporting) session may be problematic because it can often be hard to find an appropriate way to use the data. Many suggest that verbal reports need to be collected as the user is completing the task and can be used with the data on task performance for usability analysis (Karat, 1997; Lewis, 1992; Gould, 1988; Rosson and Carroll 2002). If a researcher waits to do a follow-up survey after

interactive system use, user responses are then based on afterthoughts, which may be influenced by the information they have learned while completing the rest of the task. On the positive side, verbal reports are a good way to gain insight into cognition and information from someone's short-term memory. Through research, it has been shown that verbal reports are good for iterative design changes but not formal evaluations (Karat, 1997).

Surveys and questionnaires can be implemented in many ways. Surveys can have free-form questions, rating questions, and even "Yes" or "No" questions. Questionnaires, like inspection-based methods, are fairly inexpensive and easy to distribute. Questionnaires should ask about specific user experiences, not hypothetical situations, and can assist in finding usability problem spots present in an interface (Karat, 1997; Gould, 1988). The use of rating scales, rather than short answer questions, can allow data to be quantified and put into charts (but typically not submitted to statistical analysis). Because the information received through surveys and questionnaires is gathered after the completion of a task, some information about the experience will be lost due to human memory limitations. Even though some questionnaires can be quantified, free-form surveys cannot. It has been recommended that questionnaires accompany verbal reports in usability analysis in order to serve as a means for validating verbal report responses (Karat, 1997; Gould, 1988). Karat (1997) said questionnaires should be used during iterative design in order to use the feedback to create better designs.

All of the methods of a usability evaluation detailed above are qualitative measures. Each method has its advantages and disadvantages, and it has been recommended that the majority of these methods should be used during iterative design. Despite this fact, often in industry, such methods are used for evaluation after the product design is complete. Some of

these methods can be low cost, such as the surveys and expert evaluations (heuristic-based evaluations), while others can be rather expensive, such as verbal reports developed through user testing. The setup cost, including creating a usability lab, for such analyses can be expensive, but it is a one-time cost. Much of the overall cost comes from subject recruitment and honorarium fees for participation in test runs. For these reasons, often companies outsource usability analysis work; however, this may be more expensive than doing a test in house.

The low cost method of usability inspection, as stated before, may only be complimentary to user testing. Surveys do not provide as much insight into problems as user testing; therefore, in order to obtain useful information, iterative user testing is still needed. An interface that was first inspected by an expert, then tested by users, resulting in verbal reports, and followed by surveys, is a good way to gain enough information to create a user centered design. Therefore, the money that it takes to do testing will still be applied later on, iteratively in the design cycle.

### 2.3.2 Quantitative Measures

Quantitative measures are used to make comparisons of designs based on quantities associated with certain interface features. Usually researchers use quantitative measures during iterative design processes. There are several quantitative measures that do a good job of predicting "good" and "bad" usability, and they are discussed below. Quantitative measures typically result from usability testing and/or user modeling. Using quantitative measures to assess usability is useful when it comes time to present information to the management of a company. For example, the researcher can show that a percentage of people could not complete a task based on the placement of a button in the interface. If a website design is directly related

to sales, the researcher can show how the percentage of persons that could not complete the task directly impacts business and profit.

According to Wixon and Wilson (1997), however, sometimes in usability testing, goals are too ambitious or there are too many goals. A process may also be too complex for testing or the design team is unwilling to cooperate and does not completely explain to the usability engineer how the product works. Beyond this, usability tests cannot cover entire systems and participants may not show up for testing. This is a sample of the problems that occur in usability testing in attempting to generate quantitative measures. Some of the most common quantitative measures include measures of satisfaction and fuzzy sets theory and objective measures such as correct tasks completed, task completion time, and time in a particular mode of system operation. These methods and their specific advantages and disadvantages are presented here.

#### 2.3.2.1 Quantitative Subjective Measures

Measures of satisfaction include preferences such as ranking or rating of interfaces, and ranking based on ease of use. These measures are typically collected through surveys after system use. Hornbaek (2005) researched over 180 articles that explained how to collect measures of satisfaction. He found that 93% of the studies used surveys to collect this data. Because surveys are collected after the fact and questions can be misunderstood by users, it may be hard to link specific parts of the survey to the correct parts of an interface. All the identified measures of satisfaction can be quantified and presented as graphs and charts; however, these measures are subjective and rely on the responses of users instead of observed data. They may be helpful in identifying the parts of an interface to change, but because they are opinion based, results may not correspond with the objective data from observational studies. For example,

when a researcher asks how easy was this task on a scale of 1 to 10, with 1 being very difficult and 10 being very easy, the user may say 9. However, during the task the user may not have been able to complete the task without prompting.

Fuzzy sets apply set theory to fuzzy logic. In set theory, an item either does or does not belong to a set; however, with fuzzy sets, functions are defined to determine a grade of membership of an observation into a set using values between 0 and 1 (probabilities). If an element maps to the value of 0, it is not included in a set, but if its value is 1, it is a full member of the target set. With respect to usability evaluation, fuzzy sets theory has been used to assess which interface design alternatives are superior, based on the opinions of users (Nagasawa, 1999). In logic, if a person says that alternative A is better than B and B is better than C, then A must be better than C. However, in sensory evaluations this is not always true. Seeing that sensory evaluation cannot always be evaluated with logic, Nagasawa (1999) applied Fuzzy structural modeling to usability evaluation in order to deal with frequent inconsistencies in user sensory evaluations. He took 8 VCRs and recruited 12 panelists to evaluate them in terms of usability dimensions, such as ease of operation and understandable sequences of operations. The panelists gave responses like “fairly good” and “fifty-fifty” in evaluating pairs of VCRs against each other. Nagasawa next obtained means for the various evaluation grades. He then used fuzzy sets theory to find a range for comparison of each pair of VCRs in each category between 0 and 1 in order to create a fuzzy reachability matrix. The matrix showed the relationship between each pair of VCRs. The numbers were used to create hierarchical structures to determine which VCRs were the best in regard to a certain aspect of product usability. Once the evaluation was completed, the highest rated VCRs in their respective areas could be studied to

determine the design features needed to create a highly usable VCR. This method was based on user testing and did not attempt to reduce the effects of learning and trial order effects in subject use of the evaluation methodology.

#### 2.3.2.2 Quantitative Objective Measures

Objective measures are measures based on data, such as the number of correct tasks completed or task completion time. Many objective measures, according to Hornbaek (2005), are measures of effectiveness and efficiency. Measures of effectiveness include: binary task completion, number of correct tasks completed, and task performance accuracy (error rates, spatial accuracy, and precision) as mentioned earlier. Precision is considered the ratio between correct information retrieved from a system for performance and the total amount of information retrieved. Measures of efficiency include: task completion time, time in mode, time until an event, input rate (e.g., words per minute), mental effort, usage patterns (how users make use of an interface to solve tasks), use frequency, information accessed (the amount of information users access or employ to complete a task), and degree of variation from an optimal solution (the ratio between actual behavior and an optimal method of solution). All of these measures take a count of concrete occurrences during interface testing and are not based on the opinions of users.

After doing an extensive review of the HCI literature, Hornbaek (2005) observed that a number of studies combined usability measures of effectiveness into a single measure, reported measurement values after they were combined, and then performed statistical tests on the combined values. Hornbaek (2005) argued that a combination of measures like this may hide underlying usability problems in specific areas.

Fuzzy sets have also been used to assist in an objective evaluation of interfaces. Chang and Dillon (2006) used fuzzy sets to evaluate if an interface was "good", "average", or "poor" (as a measure similar to the measure of effectiveness). First, heuristic evaluations were completed to identify "good", "average" or "poor" system interfaces (for several web applications) in terms of usability. Fuzzy sets logic was then applied to determine the number of interface inadequacies associated with evaluator judgments of "good", "average" or "poor". Next, the researchers used test subjects to complete a set of work-related tasks with the same system and each instance of inadequacy was recorded across six dimensions (system feedback, consistency, error prevention, performance/efficiency, user dislike and error recovery). The fuzzy sets logic determined that between 0-3 instances of inadequacies on any dimension was considered "good", 4-7 "average" and 8-10 "poor". For each task interface, using the six dimensions and the number of inadequacies, an aggregate score was produced. All of the dimensions were equally weighted in the score. Again fuzzy set logic was applied to determine which aggregate score was considered "good", "average" or "poor". From 0-3 was considered "good", 4-7 "average" and 8-10 "poor". All interface alternatives required user testing in order to determine which had the higher score. This method by Chang and Dillon (2006) used both subjective and objective data to evaluate an interface; however, this method is very complicated, using several analyses and data points to produce a score, and requires substantial user testing, potentially becoming very expensive. Furthermore, equally weighting of each rating dimension may not be adequate for interface evaluation because a certain dimension could cause more perceived problems than another. The method developed in this research is less complicated

and reduces the need for iterative user testing. It also applies variable weighting factors on the dimensions of system effectiveness.

#### 2.3.2.3 User Modeling

With user modeling, a researcher can collect quantitative and empirical data. User modeling is a fairly new development and has been used to predict interface action sequences based on prior use data. It can sometimes be used to determine if a user does not perform a correct action based on what a program expects a user to do. According to Fischer (2001), an early example of user modeling was the WEST system. The WEST system was a coaching system for "How the West Was Won", a game similar to "Chutes and Ladders", where the players had to form mathematical equations from spinning a wheel to determine the amount of spaces they could move. The users did not always select the best strategy and the WEST system coach pointed out the optimal move on a board. The WEST coach is an example of where user modeling is used to determine optimal interface action needs. This system also showed that humans do not always act as a programmed computer would, seemingly a flaw in user modeling. User models often do not take into account cognitive processes such as learning, and memory retention; therefore, representation of actual human performance may be limited in some cases.

User modeling is also used to observe human behavior. Fischer's research (2001) also referred to help systems that employ user modeling. An example of a help system is *Activist*. *Activist* advises users about the functionality of a system, as it is relevant to a user's task. It determines how to help a user by noting if suboptimal commands are used to reach a goal or if the user does not use a minimal key sequence to carry out the command. *Activist* can be classified as an intelligent agent, with respect to usability paradigms identified earlier. In order

for Activist to help, it recognizes what the user is attempting to do, evaluates the user's method in achieving a goal, constructs a model (from information over long usage periods) of the user, and it decides when and how to interrupt performance with useful information for completing the task at hand.

A common user modeling technique that has been researched over the last few years is GOMS modeling. It is used to predict and explain real-world task performance with interactive systems. A GOMS model shows how a user interacts with a system and shows how he or she performs tasks based on the hierarchical structure of Goals, Operators, Methods and Selection Rules (Card et al., 1983). According to Mosqueira-Rey et al. (2004), the goal is what the user is attempting to accomplish with the software or application. These goals may be subdivided and put into a hierarchical structure. Operators are atomic actions performed to complete the goals. Methods are the sequences used to achieve the goals that may also be performed simultaneously. Selection rules are rules used to decide which method is optimal in performing a specific goal, chosen from a group of feasible goals. According to Rosson and Carroll (2002), a GOMS model is first created by using human performance data to estimate the time required to complete each task. An advantage of using GOMS modeling is that it can produce accurate predictions of user actions; however, the model takes time to create and cannot account for higher level human cognitive behaviors such as learning and problem solving.

In Mosqueira-Rey et al. (2002), a GOMS model was used in conjunction with an activity logging method to compare time predictions made a priori with the GOMS simulation versus real a posteriori results obtained from actual use of the system. It was also used to compare the intended use of the system with its actual use and to identify tasks absent from the log, as well as

to estimate their duration, based on other tasks recorded in the log. The VISNU tool was used to integrate the two methods. First, the GOMS model was created for the system in its design phase. Next, the prototype was given activity logging capabilities, which were used to complete empirical studies to capture user data. The log files were analyzed with the different tasks identified and the data was used to instantiate the initial GOMS model comparing the a priori predictions with real a posteriori data. The researchers used this model with an email tool. The initial GOMS model found the same errors as the refined model based on the empirical data.

According to Kobsa (2001), some of the benefits of user modeling include: (1) the demonstration of notions about one or more types of users through models of individual users, (2) the depiction of pertinent common characteristics of users that apply to specific user subgroups working with application systems, (3) the organization or grouping of users in either one or more of the subgroups, (4) the incorporation of typical characteristics of user subgroups into individual user models, (5) the logging of user behavior, (6) and the development of assumptions about the user based on past interaction history. Kobsa (2001) also suggests that new user modeling techniques are focused on personalization for individual users. They have qualities that help with personalization such as quick adaptation, extensibility (where a company can integrate company goals into the model personalization process), load balancing to accommodate the workloads that are encountered in the work environment, and transactional consistency. These types of user models can help to provide information on the user, as needed through interface design modifications. The models discussed in this section can help increase usability by providing valuable resources to designers but they do not evaluate or measure the effectiveness of an interface or determine problematic areas present in the interface.

#### 2.3.2.4 Summary of Quantitative Measures

Quantitative measures can provide meaningful information to companies. Statistical analysis can be prepared with quantitative data. Along with statistical analysis, charts and graphs can be created to present usability evaluation results. Software development company management can determine the impact of usability issues in the design of a system, from such outcomes. Quantitative data is most useful when it is objective because user's bias in opinion does not influence results. Quantitative data is, however, usually gained through user testing, which can be expensive, especially when it is used in iterative design processes.

Quantitative measures appear to be better than qualitative measures for detailed usability problem analysis and formulating design recommendations. However, qualitative measures may be necessary because they can eliminate some problems before user testing begins. Since quantitative measures are often necessary to gain support from management for addressing usability problems in design, a method is needed to reduce the requirement for recruiting subjects for testing during iterative design. User modeling is a quantitative measurement approach that can decrease the cost of iterative testing. Combining user modeling with subjective usability measures may provide an integrated analysis approach that is effective in terms of finding problems and providing an adequate basis for interface redesign.

### *2.4 Operations Research Methods of Usability*

Operations Research (OR) is the use of mathematical models, statistics, and algorithms to aid in decision-making. It is used to analyze real-world and complex systems and improve and optimize performance. Operations Research is a subfield of applied mathematics and it can also be tied to computer science. Some of the primary areas in Operations Research are statistics,

optimization including linear and nonlinear programming, stochastics, game theory, graph theory and simulation. Operations Research has been instrumental in improving the design of entire systems versus specific elements (e.g., Polaris Missile Project (Ravindran, Phillips & Solberg, 1987); or a distribution center's logistic issues). Operations Research is often used in facility layout, telecommunication network design, manufacturing system design, and scheduling. Only a few HCI studies have historically used OR methodologies as a technique for improving usability. However, a considerable amount of research is currently being conducted in this area to determine how usability can be assessed and improved using Operations Research methods. Some of the techniques currently being used to advance usability in HCI are stochastics and optimization. Some stochastic methods include Markov models and probabilistic finites state models. An optimization method that is also being used is the critical path model. The purpose of this section is to present some examples of how OR methods have been used to improve usability in the past and to provide further direction to the current work in developing an objective, model-based measure of usability.

### 2.4.1 Markov Chain models

Markov chains are stochastic processes that take on a finite or countable number of possible values. There is a set of states X. The process is said to be in state *i* at time *n*. Whenever the process is in state *i*, there is a fixed probability $P_{ij}$ that it will next be in state *j*. There is a conditional distribution of any future state $X_{n+1}$, given the past states $X_0$, $X_1$,…, $X_{n-1}$ and the present state $X_n$. The Markov chain value is independent of the past states and depends only on the present state. $P_{ij}$ represents the probability that the process will, when in state *i* , make a transition to state *j*. Since probabilities are nonnegative and since the process must make

a transition into some state, $P_{ij} \geq 0$, $i, j \geq 0$, $\sum_{j=0}^{\infty} P_{ij} = 1$, $i = 0,1,\ldots$. **P** denotes the matrix of transition probabilities $P_{ij,}$ so that **P=** $\begin{Vmatrix} P_{00} & P_{01} & P_{02} & P_{03}\cdots \\ P_{10} & P_{11} & P_{12} & P_{13}\cdots \\ \vdots & \vdots & \vdots & \vdots \ \ldots \\ P_{i0} & P_{i1} & P_{i2} & P_{i3}\cdots \end{Vmatrix}$.

Markov Chains have been used for website customization by creating algorithms that can predict human behavior with interfaces to determine which web links to display to the user next. In the research by Jenamani et al. (2001), algorithms were created to generate and display helpful links that were based on an aggregate profile of visitor behavior, while users navigated the site. The profile was derived from navigational behavior data created during recent visits, a visitor's navigational patterns during the current session, and site parameters. The data was used to create Markov transition matrices. Algorithms, including a company's interest page algorithm, were also created. A company would mainly be interested in increasing customer purchases and how easy it is for a customer to reach that goal state. The company interest page algorithm used the transition matrix and integrated the concept of a company's utility function with values between 0 and 1. The utility function was based on the state the company wanted the user to enter after leaving a previous state in order help increase profits. To conduct this research, data was collected from 100 users and 4 suppliers that ordered computer accessories. The user's actions were collected to generate the transitional probability matrix for the system. The models were then compared against actual human data. The Markov algorithm accurately depicted the actions of the users.

Using Markov chains in the above work created a better user experience by helping the customer to access information needed at the correct time. The system used the stochastic model to provide output that was most expected by a user in order to complete a given task. It also helped the company provide web services that guided users to buy items based on the models considering previous behavior data. Even though this research provided useful output for users, promoting website usability in terms of access to relevant information, the modeling approach did not help eliminate other usability issues (e.g., effective navigation within the menu system).

Markov models have also been used to model user behavior and create algorithms that predict user actions in order to determine measures such as the number of mouse clicks it takes to find relevant information on a website and the number of steps needed to complete a task by simulating human actions. According to Kitajima et al. (2005), usability specialists can use algorithms created by Markov chains to determine the average number of clicks it takes to find, for example, relevant articles on a website. The algorithm was first applied to the original website interface and then used again after applying usability improvement techniques to the site in order to determine if the average number of clicks was reduced. In order to create such Markov models, user testing was initially completed. Behavior state observations were then coded in transitional probability matrices, and an algorithm was created and validated by comparing it to the initial data. This research was considered as a key basis for the new usability evaluation approach developed in this study.

Thimbleby et al. (2001) applied Markov chains to several applications such as a microwave oven interface and a cell-phone dialing interface. The authors used Markov chains, to model states of these devices and to predict the number of steps that it would take for a user to

perform specific tasks. In order to gather data, the authors created a Mathematica simulation of a microwave control panel and recorded the data. The Mathematica simulation modeled the device and not the user. Data was gathered from the simulation to determine ease of use. A shortest path matrix was developed for each interactive task and Thimbleby et al. found that it took 3 button presses to optimally get from one interface state to another. To obtain usability metrics, consideration of user's knowledge was necessary. The authors used a mixture of perfect error-free approaches to performing a task with the original **P** matrix. Then, a user knowledge factor k that ranged from 0 to 1 was defined with k=1 representing a user behaving like a fully knowledgeable user and k=0 representing a user with totally random behavior. The transition matrix to accommodate user knowledge then became kD + (1-k)**P,** where D was the perfect knowledge transition matrix and **P** was the regular transition matrix. Thimbleby et al. (2001) created the **P** and D matrices, where the D matrix contained only "1s" and "0s", as designers know exactly what to push to get from state i to j. The **P** matrix contained values of 1/5 and 3/5 because the probability of pressing any button was equal and there were five buttons in the interface. The authors then incorporated these matrices into a model to simulate the user of the microwave, based the average number of observed transitions it took to achieve a particular system state. It took 120 steps to get from the "clock" state to the "Power2" state (on average) based on completely random selection (k=0). The designers then incorporated a strategy where the LED light showed only the buttons that were necessary to push to go from one state to any other functional state. The LED light provided a better average number of steps when the user performed randomly (i.e., the novice user). Thimbleby et al. then associated the number of steps it took to move from one state to the next with the level of task difficulty. If it took several steps

to get to a specified state, then the interface operation was considered “hard” and if it only took a few steps, it was considered “easy”. This analysis was restricted to homogenous Markov chains so the probabilities were not conditional (based on prior action or learning).

In this research the authors modeled the devices to estimate the average number of button presses to an absorption state. This method of modeling a “random” user produced step counts unrepresentative of actual user behavior in learning a new system. Consequently, the improvement facilitated by the simple device redesign was less than convincing. Modeling actual user behavior can provide a more realistic approach to finding the average number of steps to reach a desired device goal state.

Markov Chains are easy to simulate and this method is fairly accurate. According to Thimbleby (2004), a model interface can be built with a matrix program to capture transitional matrices. It is then easy to simulate a user’s actions and analyze task time and/or number of steps to complete actions. A usability specialist could use this type of method during a prototyping process to determine the level of difficulty of an interface and to formulate specific design changes. The analyst could then use the method later on in the design process with the same state transitional probability matrices with another iteration of the system prototype in order to determine if the degree of difficulty was reduced by design modifications.

### 2.4.2 Probabilistic Finite State Models

Probabilistic finite state models have also been used to evaluate usability. Probabilistic finite state models include time distributions and transitional probabilities while the Markov Chain approach only uses transitional probabilities. Sholl et al. (1991) used probabilistic finite state models to generate user behavior predictions, which could be statistically analyzed for

comparison between alternative human computer interfaces. The purpose of the research was to show the modeling technique only. The modeling technique used an approach in which probabilistic state models of use of prospective human computer interfaces were constructed from data collected during experiments. Traditional models have a certain set of parameters that include transition probabilities, a set of time distributions, etc. These parameters are defined for each interface. Time distributions and state transitions probabilities must be developed through experimentation.

To collect the data in this research, a group of software engineers were trained, evaluated, and ranked on use of two alternative simulator interfaces. After they were ranked, they completed the tasks again using a full simulator of the interfaces with more training. The data was separated based on the initial rankings. The engineer performance data was transformed into time distributions associated with each interface state and probabilities associated with each specific transition of the state transition functions as a part of the models. For each interface model, the researchers performed order-independent parallel state coalescing (used when a set of parallel states are time and transition equivalent) as well as order-independent state splitting (used when criteria indicate that a single state actually represents more than one function of the human computer interface). The purpose of state splitting was to provide the capability of dividing complex states into multiple parallel states, when user actions are dependent on the previous interface state. They then modified the computer model of the interface to reflect coalescing and splitting. They executed a refined model as a predictor and generated derivatives of the state strings that were predicted. Finally, a statistical comparison of derivatives of strings generated by each model was conducted. Given the two interfaces that were compared with the

state splitting, etc. and the data collected from the homogenous population, the algorithm for the interface converged to a single probabilistic finite model that could be used for evaluating interface design alternatives (Sholl et al., 1991). This research however only described the modeling technique.

### 2.4.3 Critical Path Models

Critical Path Analysis, or the critical path method (CPM), is an algorithm that determines the longest time between the start and finish of a project or action sequence. CPM is often used in project scheduling as an OR technique to determine how long a project will take. It can also incorporate stochastic predictions by using the Project Evaluation and Review Technique, commonly known as PERT. CPM involves use of a network that contains nodes and arcs that represent activities of the project and interconnections among activities as well as delay times. CPM takes the critical elements of an action sequence that must be completed and determines the longest time that it will take to complete the sequence. Using this method the time of a project can be determined.

Baber and Mellor (2001) created a model using CPM to predict transaction time of an action sequence in a HCI task. He wanted to create a model that represented human activity better than standard GOMS-KLM (keystroke level models). GOMS-KLM assumes that unit tasks are combined in series and that there is no parallel activity. The CPM method created by Baber and Mellor (2001) assumed that humans do perform actions in parallel. First, he defined times used for unit tasks using published sets of resources, such as Fitt's (1954) law. Next, he set up a sequence of tasks after which he created a CPM model that described that sequence. He studied four types of models that used different dependencies. These models included:

exclusive, alternative, concurrent, and synergistic. The types of models depended on whether the tasks and/or goals were either independent or dependent. Exclusive models included independent goals and dependent tasks. Alternate models included dependent goals and dependent tasks. Concurrent models included independent goals and independent tasks. Synergistic models included dependent goals and independent tasks.

An exclusive model was developed on the task sequence in viewing vehicle registration numbers that were scrolled across a display screen. The user was then required to enter the numbers into a computer by either using speech entry or keyboard entry but no correction was available for either condition. The user could also change the rate of the numbers that scrolled across the display. The cursor keys could be used or the user could say faster or slower to increase and decrease the scroll rate (NLP). The alternative model was used in target identification where the users were required to select a geometric shape, using speech or a trackball. Once the shape was selected, a zoom command was used by speech selection or manual activity and detailed information on the object was reported. Task dependency was demonstrated in this task because one task had to end before another could begin. For the concurrent sequence, the user had to identify targets while also completing an extra activity. The users were given a number before they were asked to look at targets. During target identification they were asked to add and subtract from the given number. In the synergistic model, the operator viewed a display showing an image of a map and was required to search for waypoints that were given to them prior to seeing the map. To scroll through the map, a trackball, cursor or speech could be used. The transaction times of each of these actions was taken from available resources, including Fitt's Law, and placed in a CPM model.

Each of the previously discussed scenarios was modeled to predict transaction times for a given situation. To validate the models, Baber and Mellor (2001) tested them against user transaction times with the same tasks. A simple regression was conducted on the predicted and observed results yielding an R-squared value of 0.987, suggesting that the models accurately depicted real-world actions.

In Gray et al. (1992), CPM-GOMS was used to evaluate workstation designs for the telephone assistance operators (TAO) including an existing system and a proposed workstation. Samples of actual operators used each station and the validity of GOMS modeling for predicting performance was assessed against the human performance data. The existing workstation had functionally related keys that were color coded and spatially grouped and, as a result, common sequencing keys were spread apart. The proposed workstation tried to minimize those distances and icons were added to the display interface. Because TAOs perform tasks in parallel, the method used for modeling their actions was a critical path model. A critical path was used to determine the longest time it took to complete calls of different categories using both systems. The critical path for the proposed system took 0.8 seconds longer than the current system and was projected to result in a $2.4 million increase in operating cost.

The GOMS model developed by the authors predicted that the proposed system would take approximately 3% longer than the current system. The model showed that the proposed system was worse because the system added keystrokes along the critical path and the model recommended the company keep their initial system. Gray et al. (1992) also found that the R-squared value between the predicted and observed performance for the current and the proposed workstations was 0.69 and 0.65, respectively; showing the models adequately predicted real-

world performance. Results demonstrated that the models explained about 70% of the behavior in using the workstations. The GOMS model was considered to do a good job of predicting the actual behavior of the TAOs. This method can be considered a bridge between GOMS models and OR models, in particular, critical path models.

### 2.4.4 Summary on OR methods

Based on this review of research, using OR methods appears to be a viable and useful approach to evaluate interface usability. It provides quantitative and objective measures that can be used to compare design alternatives and create benchmarks for later investigations. Once user testing has been completed, data can be used in, for example, Markov models to repeatedly predict objective outcomes, such as number of clicks and task completion times in application use. Companies using this approach could save money by eliminating some iterative user testing on products. Using OR techniques to facilitate more meaningful output from systems will also be helpful to users. If a user feels as if information presented on a display is relevant to his or her purpose, the amount of cognitive processing needed to determine the appropriate next steps in interaction can be greatly reduced. The user will feel more comfortable in, for example, using a website with output based on a Markovian profile model of user behavior as opposed to others, where he or she may use more cognitive resources to complete a goal. Providing meaningful output and reducing the need for iterative testing by using Markov models is as simple as using a prototype to collect user data. The user data can be collected, for example, through use of a Java script embedded in a web application that records the clicks from one state to another. With enough user data, transitional probabilities can be created for interface states based on the

probability of users actually progressing from one state to another, given a previous history of states.

## 3.0 SUMMARY AND PROBLEM STATEMENT

In order to facilitate usability improvements in complex interface design, there needs to be a foundation or framework to describe how humans and computers communicate. If researchers can model this communication, they can effectively create designs that reduce cognitive and computational loads for performance, effectiveness and efficiency. Of the three frameworks discussed, Norman's model, Abowd and Beale's model and the Pipe-Line model, Abowd and Beale's model of interaction was observed to describe human interaction with computers in a more robust and parsimonious way than the other two models. Norman's model is more concerned with the human thought process and not the continual (parallel) communication cycle between the computer and the human. Furthermore, it does not explicitly represent the interface between the human and machine system in modeling interaction. The Pipe-Line model focuses more on the computational process of the computer and to a lesser extent on the human cognitive process. Abowd and Beale's model takes into account the human, the computer and interfaces, and represents a more continual cycle of interaction. Using this framework, designers and users can conceptually categorize different systems and begin to evaluate usability for selection between a set of usability paradigms; that is, alternative interface designs for particular applications.

Usability paradigms and principles are important because they help usability analysts determine what types of new technologies are available to improve systems and they provide direction in how to evaluate system usability. In this research, different paradigms are considered in order to determine which may effectively reduce both cognitive and computational loads and improve overall system performance. Because the command line interface and the

NLP interface represent extremes of user articulation and machine command translation, they can also be seen as representing the maximum and minimum distances between the user or system and the interface in Abowd and Beale's HCI framework. In this research, the WIMP interface paradigm was tested for accommodating both the user and the system in terms of a new measure of usability. Using a WIMP interface as a basis for an online ordering system design was expected to reduce the conceptual distances between the user and the interface, or the system and the interface, as much as possible (relative to other paradigms), without making the task of language translation complex for either entity.

There are many subjective measures that are used today as a basis for improving usability. Usability analysts use subjective data to make design decisions. These decisions are based solely on the opinions of users. Even though a user may say a task is easy they may have had enough trouble with an interface to cause a test moderator to help the user complete the task. This example suggests that using subjective data alone may not be a sufficient method of measuring and evaluating usability. Beyond this, qualitative data is not easy to analyze. A usability analyst cannot create charts and graphs to present to company management with qualitative data from open-ended survey questions and the ability to conduct statistical analysis on such data is limited. Even though subjective measurement methods tend to be low cost, they are also usually low in discovery of major usability issues, which might otherwise be discovered if user testing was conducted.

Quantitative data on the other hand can be analyzed statistically and charts and graphs can be created to show company management the impact of usability on performance. Objective quantitative data is usually gathered through user testing and is not opinion-based. Because

quantitative data requires iterative user testing it can be very expensive. Given this expense, a method that can reduce the amount of testing involving end-users for quantitative data would be preferable to companies. For this reason, various approaches to user modeling, including GOMS have been developed to predict user behavior. These methods have been demonstrated to be successful in quantitative evaluation of complex interfaces in real applications. Because some subjective measures are instrumental in revealing usability problems and objective, quantitative measures are important for identifying design solutions to actual problems that may be encountered in real work environments, it is generally recommended that both types of measures be used to evaluate the effectiveness of a system.

Operations Research has been used to optimize entire systems but has not been used frequently to evaluate usability. Operations Research techniques can be used to create user models and generate quantitative evaluation data, possibly eliminating the need to recruit and pay large samples of users for testing interfaces. Use of Markov Chains is easy to implement and appears to be an accurate way to determine specific quantitative attributes of an interface that influence usability and performance. Markov Chains can be constructed to model user behavior and they can be integrated into interactive systems to drive the presentation of more meaningful output for users (e.g., in web-based applications). Currently, Markov Chains have been used to either measure usability in terms of user input requirements or to predict more meaningful display output for users in search tasks.

The use of Abowd and Beale model of HCI, consideration of contemporary usability paradigms and principles, and use of OR modeling techniques can be combined to evaluate overall interactive system performance, addressing usability issues and computational loads.

This research was expected to contribute to approaches to usability analysis in part by utilizing a well- defined HCI framework to define a new measure of system performance for use with system designers and users. The links between the nodes in Abowd and Beale's framework can be considered as conceptual indicators of the degree of usability and efficiency for system design. The approach investigated here incorporated subjective analysis of these links by key contributors in the design of an online ordering system as well as end users. Designers provided ratings of the importance of the links in the Abowd and Beale framework from a development perspective. Users provided ratings of online ordering system design alternatives for each link in the framework. This data was then integrated for a partial system effectiveness score. After this, OR techniques were applied to assess quantitative attributes of usability of any high scoring, candidate designs. Markov Chains have recently been used to predict human input behavior. Markov Chains can be used to assess task times or the number of (mouse) clicks it takes to complete a task. The use of objective and quantitative data on HCI, generated by using Markov Chains was combined with the partial system effectiveness score, based on the designer and end user ratings, to yield an overall score for the online ordering system. This approach can be used to select among specific system design alternatives to evaluate system effectiveness and is expected to result in a decrease in cost for companies by reducing the number of iterative user tests as a part of a design process.

## 4.0 METHOD

### *4.1 Overview of System Effectiveness score*

The Abowd and Beale interaction framework and designer and user surveys were used in conjunction with Markov model predictions of the average number of clicks at an online interface in order to determine the system effectiveness score. Each link between the nodes in the Abowd and Beale model represents a user or system language translation that affects overall system effectiveness (See Figure 3). More specifically, the link between the user and the interface input (articulation) represents how well the user's goal matches the input state of the system. The link between the input and the system (performance) determines if each input has a logically mapped function. The link between the system and the interface output in the framework (presentation) represents the accuracy and completeness of system state information provided to the user. The link between the output and the user (observation) represents how well a user can interpret the output from the information display. In order to determine a system effectiveness score, all of these links need to be evaluated.

When designing a system for "optimal" usability, designers may consider some links in the framework to have a higher priority than others. The priority or importance of each link may also vary across design paradigms. With this in mind, for usability assessment, there is a need to capture the designer's perception of importance or weight of each link of the Abowd and Beale framework in design. If users are able to rate system design in terms of the links in the framework, weighting factors from designers can then be multiplied by the average user ratings for each link to determine a partial usability score. The novelty here is that the score reflects the designer's intent for the application, as well as the user's perception of the system. For example,

if a designer places emphasis on "articulation" in the framework and the user perceives ease of translation of goal states to action, the interface alternative would yield a high score. If a designer, however, has placed emphasis on "performance" in the framework and the user perceives slow system response, the interface alternative would yield a low score.

A Markov model was used to predict the average number of clicks to goals for each design that is rated. The partial system score was then divided by the click count, yielding a measure of usability for each action with the system. This ratio was then to be determined for interface alternatives and the design with the highest ratio was identified as the best system alternative. The overall system effectiveness (SE) equation is structured as follows:

$$SE = (a_1 \times art + a_2 \times perf + a_3 \times pres + a_4 \times obs) \div avg\ number\ of\ clicks$$

where,

art = average user rating of the ease of translation of goals to input states

perf = average user ratings of system responsiveness to inputs

pres = average user ratings of system speed and accuracy in presenting output

obs = average user ratings of the ease of translation of outputs

$a_1$ = average designer weight for importance of capability to map psychological intention to input states

$a_2$ = average designer weight for importance of capability to map input options to system functions

$a_3$ = average designer weight for importance of capability to accurately represent system states through output

$a_4$ = average designer weight for importance of capability to map display features to user task requirements

### *4.2 Weighting Factor Determination*

In general, designers may be more concerned with the links in the HCI framework that drive user cognitive load. Designers are concerned with the mental effort expended by users. To

identify differences in the perceived importance of the links in the interaction framework among designers, several expert software designers (n = 4) at a major PC manufacturer were surveyed. Average weights across designers were used for evaluating various alternative designs for the target on-line PC ordering website. The designers were presented with Abowd and Beale interaction framework. They then determined, based on the application type and general usability paradigm for the application, how important each link was to system effectiveness. All weighting factors were determined by designer pair-wise comparison of the links in the Abowd and Beale model (see Appendix A). Values ranged between 0 and 0.5 with 0.5 being the most important to system effectiveness. On each line, a designer circled one type of link to indicate greater importance to the application design. The total number of times a particular dimension was circled was divided by the total number of pair-wise comparisons (in this case, 6) to determine a weight for each design dimension. It is appropriate to use pair-wise comparisons of design dimensions to identify tradeoffs that are typically made in real-world design and production because of usability or cost constraints. The weighting factors provided by the different designers were averaged and used in the equation previously presented. Because cognitive load is typically more important to usability, it was expected that designers would produce higher weighing factors for “art” and “obs”, compounding end user usability ratings to impact the overall system score (more so than those links of the HCI framework driving computational load).

### *4.3 Experimental Task*

Half of the participants used an online ordering system to order a Lenovo ThinkPad R60 using an existing Lenovo website interface (see Figure 4 for image of web page). Based on prior

use, the existing interface required a minimum of 11 clicks to complete the task of ordering a product. First, the user goes from the Lenovo home page to a “products” page (or can use a shortcut to a “notebooks” page). Subsequently, the user can choose the “ThinkPad” section of the website from the “products” page among the “Lenovo 3000 notebooks”, “ThinkCentre”, “Lenovo 3000 desktops” and “accessories” sections. After choosing “ThinkPad”, the user would then need to select the “R Series” among four other series possibilities. On the “R series” page, there are three tabs that include: “products”, “features”, and “upgrades and services”. When the “R series” is clicked from the “ThinkPad” page, the user is automatically taken to the features tab, which shows that the “R series” has a fingerprint reader and other features that may be of interest to the customer. The user must then navigate to the “products” tab by clicking it to determine which model has the correct technical specifications (e.g., 15 inch display and price less than $750). Following this, the option to “customize and buy” the “R60” must be selected from within the content in the tab. After choosing the “customize and buy” option, the user then goes through a three-step customization process, adds the notebook to their virtual cart and goes to checkout. The procedure represents use of the existing Lenovo system-purchasing interface.

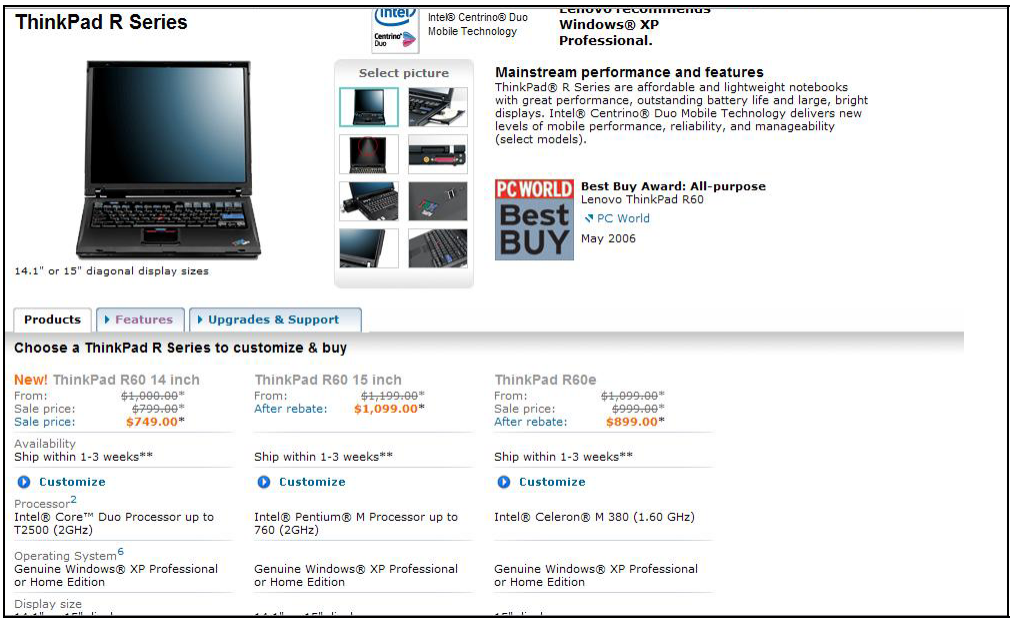



Figure 4: Series page from the existing interface

A new web interface design was prototyped to require a minimum of nine clicks to complete the ordering task. This interface contained multi-level navigation, where the user could mouse over or hover on “products” listed on the main menu bar at the top of the page and a drop down menu would show the options to select a “notebook”, “desktop” or “accessories” (See Figure 5 for image of the new prototype interface). Mousing or hovering over “notebooks” produced another dropdown menu that showed “ThinkPad” and “Lenovo 3000” options. The user would pick the “ThinkPad” based on the task instructions to buy the notebook with certain specifications. The “ThinkPad” page presented the five series of machines that a user could choose from. Below the machine types were technical specifications allowing a user to compare between the series. Below the technical specifications, there was a chart detailing which series included certain state of the art technology. If the user scrolls down this page, more information on the ThinkPad is presented. This same data was included in the “features” tab of the original

interface. Once the user finished viewing the “ThinkPad” page, based on the information presented and the specifications in the task, (s)he would choose the “R series”. The “R series” page in the new interface was set up similar to the ThinkPad page with three models in the “R series” family. A picture gallery, however, was presented before the breakdown of the different machines available in the family. From the “R series” page, the user would choose the “R60” to customize and buy based on the other information and specifications shown on the page. Once the user selected the correct computer, they had to customize the computer, which required the same three-step customization process (as in the original interface), adding it to the cart and going to checkout. Because the new interface included multi-level navigation, navigation to the ThinkPad page could be reached in one click. Based on prior usability testing, more salient buttons, representing the call of action on the web page, were also added to assist the user to the goal state.

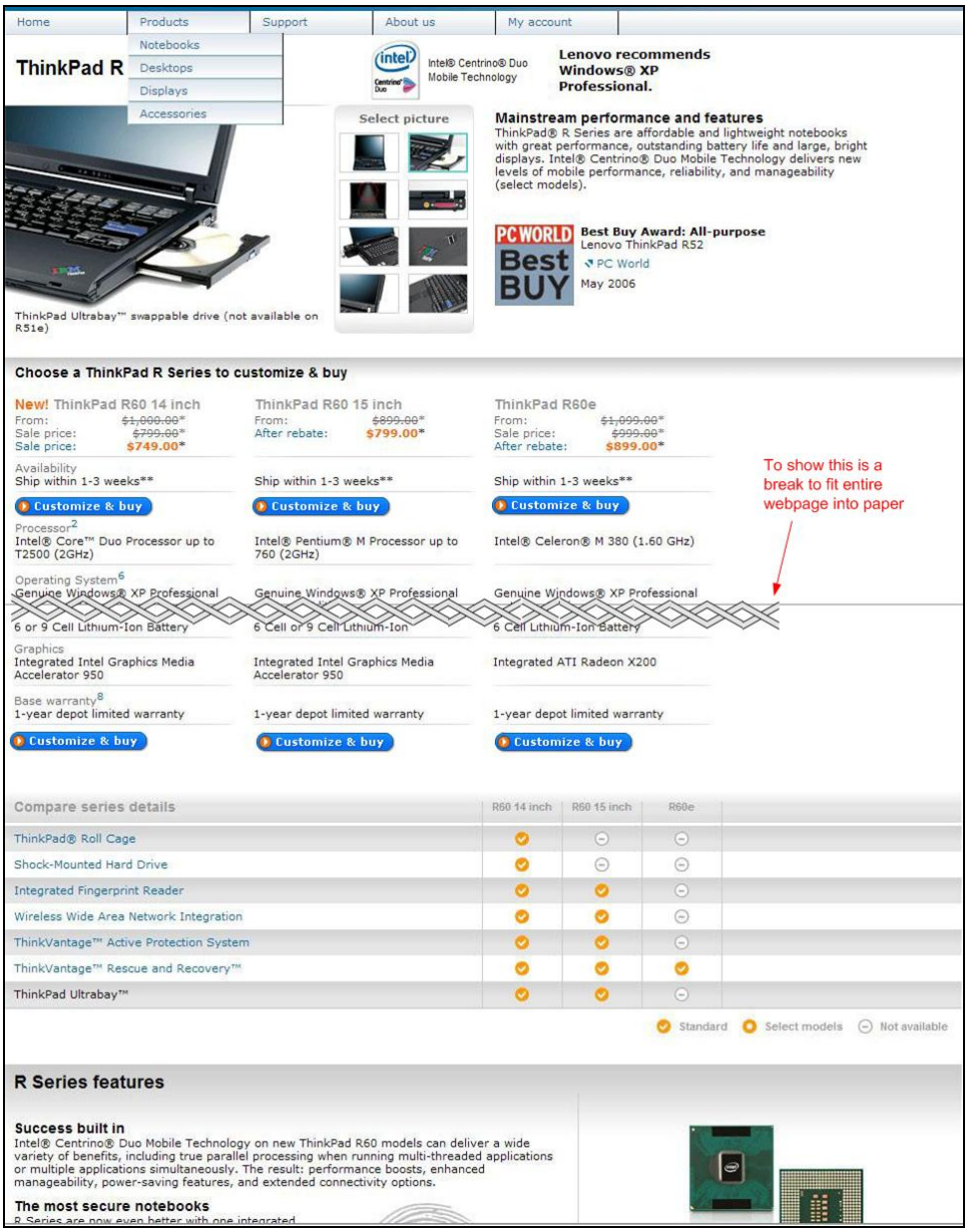



Figure 5: Series page for the new interface

### *4.4 Developing the Markov Chain Models*

In order to create a Markov model that helps a designer objectively choose between interface alternatives, from a usability perspective, transitional probabilities for transitioning among the various interface states were determined. Twenty participants, who were computer

literate and accustomed to working with online ordering systems, were selected to complete the given task of finding and ordering a ThinkPad (with a 15 inch display and under $750). The subjects included 11 males and 9 females ranging in age from 17 – 25 years. Half the subjects used the existing Lenovo interface and half used the new prototype.

The existing online ordering system interface was used to identify the state features (within a typical ordering process) that are currently employed in the interface (e.g., a tab, link or menu option). A state was considered anything that takes the user to a different page. Pop-ups were not included because a user remains on the same web page. Radio buttons and checkboxes also were not considered states because they did not navigate the user away from a page. A user's action sequence with either the old or new interface was recorded in order to create a state transitional probability matrix for each design. The matrices revealed the probabilities of subjects for going from one interface state to another, based on the actual counts of a user selecting to go from state *i* to state *k*. For example, if there were 10 clicks (across participants) from state 1 (home) to state 2 (products) and 2 clicks from state 1 (home) to state 3 (notebooks) then the probability from state 1 to state 2 would be 83.33% and the probability from 1 to 3 would be 16.667%. The actual number of clicks it took for a person to reach the goal state (a purchased ThinkPad R60) was recorded in order to validate the Markov model prediction of clicks. Assumptions of the Markov model included: (1) the sum of each row in the matrices must equal one; and (2) the probability of the next interface state must only depend on the current state.

Using a Java Script embedded in the web page code, a record of each option selection made by a user was collected with the existing and new interfaces. Based on the current states,

and the user action sequence, transitional probability matrices were created for the existing and new interfaces. To determine the average number of clicks to the goal state, an equation used by Kitajima et al. (2005) was employed. The equation is: $u^{(i)} = 1 + \sum_{k \in T} P_{ik} u^{(k)}$. The equation states that 1 step is required to progress from state *i* to another state. It also takes an average of $u^{(k)}$ steps to progress from the transferred state *k* to the absorption state (in this case checkout). So each $P_{ik}$, the probability of moving from state *i* to state *k* by the group of users, was multiplied by the average number of steps it would take to get from state *i* to state *k* taking direct paths to the goal or (absorption state). The likely number of clicks to go from one state to the next in a sequence to the goal state were then added together, to determine the average number of clicks.

The new interface was expected to improve usability through the replacement of the normal menu system with a multi-level menu system and by adding more pronounced buttons highlighting the actions on the page. The proposed interface was given to a different set of ten participants to avoid the effects of learning in the experiment. The Java Script was again used to record user navigation data to determine the state transitional probabilities for the proposed interface. The same analytical procedure was also used to predict the average number of clicks for the new interface. It was expected that the average number of clicks should be reduced after creating the multi-level navigation system and highlighting the possible actions with "bigger" and "brighter" buttons, making more salient the next possible steps in the purchase process, based on the current interface state. The actual average number of clicks in use of the new interface was also recorded in order to validate the Markov model.

If a company wanted to eliminate the need for iterative user testing, the same probability matrix from the existing interface could be used to predict the average number of clicks for the

new interface. In the equation to find the average number of clicks for the new interface, the average number of steps it takes to transition from state *i* to state *k* would change for the new interface, because many of the states can be reached with one click (with the new navigation). Therefore, the average number of steps to go to a state with the new interface was multiplied by the same probabilities used from the existing interface. However, because the average number of steps was reduced (in comparison with the existing interface) the product was expected to be lower for the new interface, which would reduce the number of predicted clicks.

### *4.5 Rating System Effectiveness Based on the Abowd and Beale framework*

Considering the Abowd and Beale framework, the objective was to identify an interface design providing superior usability and performance. The links rated by the end user for each interface design, in order to assess usability, include the articulation link (art) between the user's goal and the input state of the system (cognitive distance) and the observation link (obs) where the user must correctly interpret the output from the system (cognitive distance). The performance link (perf) between the system functions and the input options (computational load), and the presentation link (pres) representing the degree of speed and accuracy in system state output, are links that were rated by the users to evaluate the responsiveness of each alternative design to commands. (The rating form completed by the users is shown in Appendix B.)

In order to capture link ratings, the end users were presented with the Abowd and Beale interaction framework after completing the tasks in the usability testing sessions. They rated the links for either the existing or new interface designs. They were asked to rate the links on a scale from 1 to 10, where 10 represents "perfect" usability. The ratings were averaged for each of the designs (across users) to provide a value for use in the structured equation to quantify overall

system effectiveness. The users rated both the links representing usability dimensions and the links representing system performance.

### *4.6 Calculating the Overall System Effectiveness Score*

In order to find a total score of system effectiveness to compare the interface design alternatives, all of the previous information was required (weighting factors, link ratings and predicted average number of clicks). The average weighting factors were multiplied by the average ratings for each link for each interface and divided by the average number of clicks to determine the ratio of perceived usability (or effectiveness) per interface action. A simple t-test was then used to determine if the partial and overall system effectiveness scores for each interface were significantly different. The interface design with the highest ratio was considered to have better system effectiveness. It was expected that the new interface would yield a higher system effectiveness score.

### *4.7 Markov Model Validation*

A t-test was also used to determine if the actual and predicted average number of clicks from the Markov model for each interface were significantly different. It was expected that the predicted and actual number of clicks would not be significantly different for either web interface design alternative. This test was expected to provide evidence of the validity of the modeling approach for objectively evaluating user performance with the online ordering system.

## 5.0 RESULTS

### *5. 1 Computation of average number of clicks*

Click data was collected on each participant. The transitional probability matrices were formed as follows: (1) A "from-to" matrix was setup using the observed click data. The number of times participants went from state *i* to state *k* was recorded in the *ik* cell. (2) The sum was determined for each column and each row. (3) The proportion of clicks was calculated by dividing an individual cell by the sum of its row. (This procedure was used to calculate the proportions for both the new and existing interfaces.) (4) The proportions were converted into probabilities, yielding the original transitional probability matrices.

In using Markov models, one assumption is that a transition from one state must only be dependent on the current state. The observed user data revealed the click history (where the participant started from and where they clicked). This data was replaced by the probability $p_{ik}$, corresponding to the "from-to" data for each subject. The probability matrix for the new interface appeared constrained; however, there are in fact two paths to reach the "R series" state. The main menu navigation was the preferred path for nine out of the ten users who took a single path to reach the "R series" state. The transitional probability matrices for the new and old interfaces are presented in Figures 6 and 7.

$$P_{ik}\ \text{new} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 6: Transitional probability matrix for the new interface

$$P_{ij}\ \text{existing} = \begin{pmatrix} 0 & .73 & .27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .40 & .50 & 0 & .10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .89 & .11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .07 & 0 & 0 & .87 & 0 & .07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .50 & .50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .08 & .08 & 0 & .08 & 0 & 0 & .76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 7: Transitional probability matrix for the existing interface

A Durbin-Watson test was used to determine if there was a first-order autocorrelation of the state transition probabilities and to assess if the assumption of the Markov Chain was met. When the Durbin-Watson statistic is greater than 2, there is no significant autocorrelation. The Durbin-Watson test revealed a significant positive correlation for the data on the existing interface with a statistic of 1.2879. Consequently, a two-stage normalization procedure was necessary. The new interface transitional probability matrix yielded a Durbin-Watson statistic of 2.0815; however, to ensure no correlation, the two-stage normalization procedure was also conducted on the new interface. In order to normalize the data, the "from-to" matrix was used again. The individual

cells were first divided by the column sums. New row sums were then calculated. The individual cells were then divided by the row sums, creating a normalized matrix. The normalized transitional probability matrices for the new and old interfaces are presented in Figures 8 and 9.

$$P_{ik}\ \text{new normalized} = \begin{pmatrix} 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{pmatrix}$$

Figure 8: Normalized transitional probability matrix for the new interface

$$P_{ik}\text{existingnormalized} = \begin{pmatrix} 0 & .71 & .29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .34 & .27 & .38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .53 & .47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .09 & 0 & 0.83 & 0 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .47 & .53 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .46 & .05 & 0 & .3 & 0 & 0 & .46 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 9: Normalized transitional probability matrix for the existing interface

The same Durbin-Watson procedure was again preformed on the probabilities resulting from the normalization process. The data collected from the new interface showed no correlation with a Durbin-Watson statistic of 2.2716; however, the existing interface still revealed significantly correlated probabilities, although slightly less than the original data (a Durbin-Watson statistic of 1.3920). Since the two interfaces were similar in nature, and the Durbin-Watson test revealed a

mix of evidence of autocorrelation, the normalized transitional probability matrices were used for the Markov Chain modeling. Related to this, from an interface perspective, it is important to note that websites for online ordering or purchasing products are meant to reduce memory loads by showing users, on one page, all information necessary to make a decision to go to a next step. Once that next step is made, the previous information is no longer needed, or it is provided again on the current page, as a basis for decision making or to reduce the need to remember information across pages. Therefore, from the web page design perspective it seems reasonable to assume that user actions are primarily based on current page information.

As part of the Markov modeling, an additional matrix was formed to present the average number of steps it should take a user to get from a state *i* to state *k*. The designers, who know the feasible paths, from any one state to another, were interviewed to create this matrix. Because there are different paths to any state, the number of clicks from one state to another in each path is summed and divided by the number of paths to find the average number of steps. The matrices of average number of steps for the existing and new interfaces were as follows:

$$M_{ijexisting} = \begin{pmatrix} 0.00 & 1.00 & 1.50 & 2.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 & 8.00 & 9.00 & 10.00 & 11.00 \\ 1.00 & 0.00 & 2.00 & 2.00 & 2.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 & 8.00 & 9.00 & 10.00 \\ 1.00 & 1.00 & 0.00 & 1.50 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 & 8.00 & 9.00 & 10.00 \\ 1.00 & 1.00 & 1.50 & 0.00 & 2.50 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 & 8.00 & 9.00 \\ 1.00 & 1.00 & 1.50 & 2.50 & 0.00 & 3.50 & 4.50 & 5.50 & 6.50 & 7.50 & 8.50 & 9.50 & 10.50 & 11.50 \\ 1.00 & 1.00 & 2.00 & 2.00 & 2.00 & 0.00 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 & 8.00 \\ 1.00 & 1.00 & 2.00 & 2.33 & 2.50 & 2.67 & 0.00 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 & 7.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.67 & 4.00 & 0.00 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 6.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.33 & 3.67 & 4.00 & 0.00 & 1.00 & 2.00 & 3.00 & 4.00 & 5.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.50 & 4.00 & 4.33 & 4.67 & 0.00 & 1.00 & 2.00 & 3.00 & 4.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.50 & 4.50 & 4.67 & 5.00 & 5.33 & 0.00 & 1.00 & 2.00 & 3.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.50 & 4.50 & 5.50 & 5.33 & 5.67 & 6.00 & 0.00 & 1.00 & 2.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.50 & 4.50 & 5.50 & 6.50 & 6.00 & 6.33 & 6.67 & 0.00 & 1.00 \\ 1.00 & 1.00 & 2.00 & 2.50 & 2.50 & 3.50 & 4.50 & 5.50 & 6.50 & 7.50 & 6.67 & 7.00 & 7.33 & 0.00 \end{pmatrix}$$

Figure 10: Average number of steps from state i to state k for the existing interface

$$M_{ik}\text{new} = \begin{pmatrix} 0 & 1 & 1.3 & 2 & 2 & 3 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 0 & 1 & 1 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 1 & 0 & 1 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 1 & 1 & 0 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 1 & 1 & 1.5 & 1.5 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2.5 & 2.5 & 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 3 & 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3.5 & 3.5 & 3.5 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 4 & 5 & 4.5 & 4.5 & 4.5 & 0 \end{pmatrix}$$

Figure 11: Average number of steps from state i to state k for the new interface

The equation specified by Kitajima, $u^{(i)} = 1 + \sum_{k \in T} P_{ik} u^{(k)}$ was used to find the average number of clicks from the initial state to the absorption state, which in this research, was the confirmation page after purchasing a computer. The probability of going from state *i* to state *k* (from the normalized matrix) was multiplied by the average number of steps along paths to reach the absorption state. For example, because a user can either take the path from "home" to "products" to "ThinkPad" to "R series features" to "R series products" to the customization process and then to checkout, or and they can take the path from "home" to "notebooks" to "ThinkPad" to "R series features" to "R series products" to the customization process and then to checkout, the products of these two paths were summed to find the average number of clicks to the absorption state (start of machine purchase to finish of purchase).

The predicted average number of clicks, used in the overall system effectiveness score (based on the equation specified by Kitajima) was 11.5 and 9 (the actual average number of clicks was 12.9 and 9.2) for the existing and new interfaces, respectively. The predicted number of clicks

for each participant across interfaces was compared. Two t-tests were used in order to determine whether the number of clicks was significantly different across the two interfaces. The first t-test compared the actual observed number of clicks for the two interfaces. The test revealed a significant difference between the two interfaces with a t value of -4.30 and a p-value of 0.0004. The number of clicks with the new interface was significantly less than the original interface. A second t-test was conducted to compare the actual click count to the predicted click count. The predicted click count for each participant for each interface was compared to the ten observations of clicks with the same interface. The test showed that for both the new and existing interfaces there was no significant difference among the predicted and actual observed number of clicks ($p = 0.8439$ and $p = 0.0605$ for the existing and new interfaces, respectively). To determine whether the predicted number of clicks was different across interfaces, another t-test was conducted on the predicted clicks for each participant across interfaces. The test showed the number of clicks across interfaces to be significantly different with a p-value of 0.0033. These tests provided evidence that the new interface actually reduced the predicted number of clicks, and that the Markov models accurately predicted the average number of clicks.

### *5.2 Partial system effectiveness score*

Each participant rated the interfaces in terms of articulation, presentation, performance and observation according to the Abowd and Beale model on a scale of 1-10 with 10 representing a "perfect" interface for the particular usability dimension. Each designer completed a pair-wise comparison of the importance of the four dimensions. This data was averaged across the four designers. The averages for articulation, presentation, performance and observation were 0.4583, 0.04167, 0.125 and 0.375, accordingly. These weights were multiplied by the average

participant ratings for each category for the two different interfaces and then added together. The average ratings for articulation, performance, presentation and observation for the new and old interfaces by participants are presented in Table 2. This procedure resulted in a partial system effectiveness score for each interface. The partial scores for the new and existing interfaces were 8.6 and 5.2, respectively.

Table 1: Average participant ratings for each of the four dimensions of usability

| | Articulation | Observation | Presentation | Performance |
|---|---|---|---|---|
| **New** | 8.4 | 9.1 | 8 | 8.1 |
| **Existing** | 4.8 | 4.9 | 7.2 | 6.4 |

In order to determine if the partial system effectiveness scores for the new interface were better than that of the existing interface, a t-test was applied to the scores for all of the participants using the new interface and all of scores from participants using the existing interface. There was a significant difference with a t-value of 5.08 and a p-value less than .00001. The partial system effectiveness score for the new interface was significantly higher.

In order to determine if designers found observation and articulation more important than performance and presentation, t-tests were again conducted. The significant levels of these comparisons are presented in Table 3. For articulation compared with performance and presentation, the dimensions were found to be significantly different with p-values of 0.0004 and 0.0013, respectively. For observation compared with performance and presentation, the dimensions were found to be significantly different with p-values of 0.0013 and 0.0055, respectively. These analyses showed that both articulation and observation, as interface design dimensions, were ranked significantly higher by designers than presentation and performance, in terms of importance for usable design.

Table 2: p-values for articulation and observation vs. performance and presentation

| | p-value | |
|---|---|---|
| | **Performance** | **Presentation** |
| **Articulation** | p = 0.0004 | p = 0.0013 |
| **Observation** | p = 0.0013 | p = 0.0055 |

### *5. 3 Overall system effectiveness score*

The overall system effectiveness score was calculated by dividing the partial usability score for the interfaces by the predicted average number of clicks from the Markov models. The calculated overall system effectiveness scores (perceived usability per click) for the new and existing interfaces were 0.939 and 0.475 respectively. To determine if there was a significant difference in the overall scores, a t-test was conducted on the individual system effectiveness scores for each participant across the two interfaces. The individual partial scores were divided by the individual predicted number of clicks and compared across interfaces. The t-test revealed the two interfaces to be significantly different with a test statistic of 5.62 and a p-value less than .0001. The test showed that the overall system effectiveness score for the existing interface was worse than the score for the new interface.

### *5.4 Reducing experimentation*

The purpose of using Markov chains to predict the number of clicks in a web interface is to reduce the amount of user testing needed, possibly reducing the amount of money and time spent in evaluating interfaces. In order to determine the predicted number of clicks for the new interface (assuming the absence of actual user data), the normalized probabilities for the existing interface were multiplied by the average number of steps from state *i* to state *k* (contained in the matrix of average number of steps created by the designers for the new interface). This

procedure resulted in a predicted number of clicks for each participant using the new interface. Using the individual normalized state transitional probabilities for the existing interface, and multiplying them by the average number of steps it should take to reach state *k* from state *i* in the new interface, yielded the predicted number of clicks for each of the participants. A t-test was used to determine if the predicted number of clicks was the same as the observed number of clicks for the new interface. The test revealed that the click values were not significantly different with a test statistic of 1.15 and a p-value of 0.270. Therefore, the average number of clicks in use of the new interface could be accurately predicted with the Markov model without actual user testing. The predicted number of clicks using the Kitajima equation was 9.35. The actual observed average number of clicks for the new interface was 9.2.

Although use of the Markov model may preclude the need for experiments to observe user performance in order to determine the system effectiveness score for the new interface, when no new user data is available, there would still be a need for a focus group to collect interface ratings in terms of the design dimensions in the Abowd and Beale framework. However, this would still significantly reduce the amount of time for interface usability evaluation compared to actual user experimentation. The partial score for the initial interface would be calculated based on focus group ratings. The partial scores could then be divided by the predicted number of clicks for the new interface, creating an overall system effectiveness score for the new interface.

## 6.0 DISCUSSION

The hypotheses as part of this work included: (1) the average designer weighting factors for articulation and observation were expected to be higher than the average weighting factors for presentation and performance; (2) the new interface was expected to improve perceived usability based on the interface design changes; (3) the new interface was expected to produce a higher system effectiveness score; and (4) the Markov Chain models were expected to be accurate for predicting the average number of clicks at the old and new interfaces.

It was hypothesized that designers would give higher weighting factors for the articulation and observation dimensions of the Abowd and Beale framework. The average weighting factors for articulation and observation were, in fact, higher than the weighting factors for the performance and presentation dimensions. These results indicate that in this interface application, the designers were generally more concerned about the cognitive load than the computational load in system design. If the customer cannot find what (s)he is looking for with a web-based ordering application it may create frustration and result in users simply leaving the site to find the product elsewhere. Frustration with the site usually leads to lost customers resulting in lost revenue. With this knowledge, designers realize that effectively reducing cognitive load, through control and display design can serve to retain customers, which drives revenue.

It was expected that the new interface would improve usability through the implemented design changes. A multi-level navigation menu was used to reduce cognitive load, making it easier to view all options in order to accurately choose the next step. The new interface also included new, more prominent buttons, which aided in easily identifying the next step in the

ordering sequence. After the participants used an interface and rated it along the four dimensions (articulation, performance, presentation and observation) of system effectiveness, they were asked which components of the interface were most usable. Survey responses from the participants using the existing interface showed that the most usable features were not the navigation or tabs but the picture gallery. The surveys from the new interface users showed that the most usable features were the multi-level navigation menu and the buttons that highlight the actions. All ten participants using the new interface thought the multi-level navigation menu was the most usable feature. Another indicator of better usability in the new interface was the higher partial system effectiveness score. The partial score for the old interface was 5.2, where the partial score for the new interface was 8.6. Because the weighting factors were the same across both interfaces, the user ratings for each link for the new interface were higher than for the existing interface. The new interface showed improved usability with higher partial scores.

The new interface was proven to be more usable because of the new navigation menu that allowed users to directly find what they were looking for. The existing interface had usability problems that were crucial to a user's performance. After going through several steps to identify the correct series of notebook, the user was taken to a tab that showed the features of the notebook series, where many participants found it hard to determine where to go next. Most participants expected the "information to buy" to be on that tab. Some participants thought they made a mistake and went to the previous page to look for the correct information. Others searched the page for several minutes before they saw the products tab that contained the information necessary to purchase a computer (refer to Figure 4). To alleviate this problem in the new interface, the feature information was merged with the purchase information, eliminating

the tabs and the number of steps required to select a model to purchase. Once the user found and selected the products tab, using the existing interface, some found it hard to determine what the next steps were. Many scrolled up and down the page and finally discovered the small button and link to “customize and buy”. In the new interface more pronounced “blue” buttons were used to draw attention to the call to action of “customize and buy” and many of the users were able to detect the button and correctly choose it. Each of these modifications reduced confusion at this step. By creating a multi-level navigation system, pages were easier to navigate with a smaller number of clicks, eliminating the individual decisions needed on each of the pages such as the “notebooks” page. The new navigation system, consolidation of pages, and new buttons created less confusion when participants used the new interface.

It was expected that the new interface would yield a higher overall system effectiveness score. The system effectiveness score for the old interface was 0.475 and the score for the new interface was 0.939, representing perceived usability per predicted click. A t-test revealed that the score for the new interface was significantly higher. The higher system effectiveness score can be attributed to the usability changes, such as the multi-level navigation menu and the new prominent buttons. The changes to the interface reduced the number of clicks required to purchase a product, creating a higher ratio for a higher system effectiveness score. The multi-level navigation menu reduced the average number of clicks necessary to reach a state. The new buttons made it easier for the user to determine the correct action on the page necessary to continue the ordering process, reducing the probability of a user clicking another button. By eliminating the tabs and placing all the information on the same page, the number of clicks and the amount of confusion for participants on where to go to buy was greatly reduced.

The Markov chain model was expected to accurately predict the average number of clicks. Using the transitional probabilities, and the equation from Kitajima, the average number of clicks was determined. The actual average number of clicks was also collected as users completed the task with the individual interfaces. On average, with the new interface, it took a user 9.2 clicks to reach the goal state. With the existing interface, on average, it took a user 12.9 clicks to reach the goal state. Using the Markov chain model, the average predicted number of clicks for the new interface was 9, while the average predicted for the old interface was 11.5 clicks. The t-test comparing actual with the predicted clicks, showed these values were not significantly different. Thus, it was concluded that the Markov chain model accurately represented the user's behavior.

The average number of clicks with the existing interface was less accurately predicted than the number of clicks with the new interface. The old interface produced a higher variance in performance than the new interface. This was likely attributable to the fact that the new interface not only reduced the average number of clicks by eliminating pages (through re-engineering of the website) but by also reducing the number of errors committed by users. In fact, the reduction in the average number of clicks was mainly due to error reduction, when using the new interface to order a notebook computer. When predicting the number of clicks for each participant, results also showed a varying degree of performance of the model matching the actual observed data (see Figure 12). The data when predicting the number of clicks for the new interface produced relatively stable results, which also reflected the observed data with new interface (see Figure 13).

The Markov model was effective in predicting the variability across the participants; however, not as effective at predicting the magnitude of the errors. It can be seen in Figure 12 that participants had trouble in navigating the interface, taking several clicks to do so. The model shows an increase in clicks for those people who had trouble or took suboptimal paths, but it does not show the extent or magnitude of their mistakes. Because Markov models are stochastic processes that attempt to predict the random nature of a process, it is appropriate to use such models to predict human behavior in HCI applications, which is not deterministic because of individual differences.

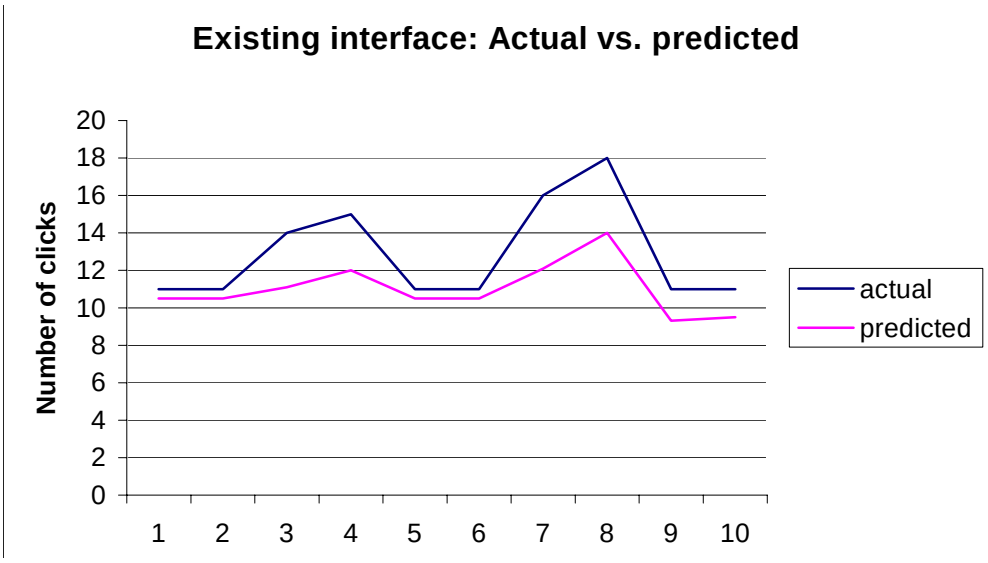



Figure 12: Actual versus predicted clicks for the existing interface

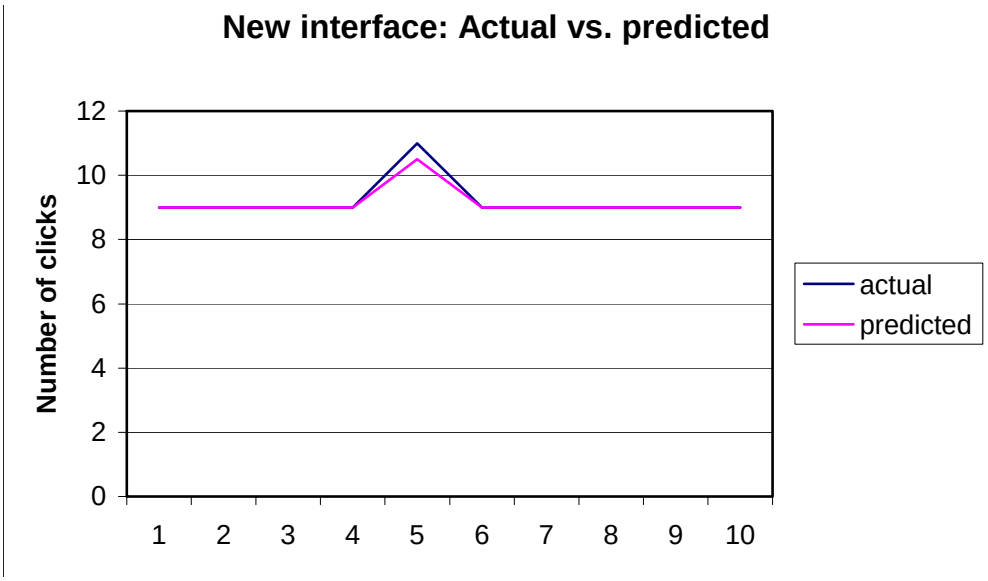



Figure 13: Actual versus predicted clicks for the new interface

In summary, all of the testable hypotheses for this work were supported by the data and analysis. Based on the comparison of the average weighting factors, the designers were generally more concerned with user cognitive load, articulation and observation. Based on the post-task survey and the partial usability scores, the new interface, using a multi-level navigation menu, more salient buttons aiding in identification of the next step in the ordering process and fewer pages to facilitate ordering a product, all resulted in a more usable application. Increases in user ratings of usability may have been due in part to reductions in the numbers of errors they made in using the interfaces. The new interface also received a higher overall system performance score because of the usability improvements mentioned earlier and their effect, in particular, the reduction of the average number of clicks. One of the main contributions of this work from an interface design perspective is that the new approach not only appeared to reduce

clicks by eliminating web pages but as a result of reducing errors. Based on simple t-tests comparing the predicted number of clicks using the Markov model with the actual number of observed clicks, the two were not significantly different. Therefore, the Markov chain model accurately predicted human behavior in the online ordering task.

## 7.0 CONCLUSION

The specific aim of this research was to create a new measure of usability to assess whether the goals of interface designers are attained and to objectively evaluate user behavior as a basis for design recommendations. In HCI (human computer interaction) applications, translation of information between the human and computer are unsuccessful because of interface design problems. Current HCI research has focused on methods to make interactive systems more usable. Unfortunately, there are few quantitative measures of usability, among many subjective measures. Subjective analysis of an interface is typically insufficient for identifying all serious usability problems and is inadequate for justifying design changes from a management perspective; thus, objective components need to be included in usability analysis. A measure combining both objective and subjective data was developed in this research. The research supports subjective and objective analysis of usability by employing the Abowd and Beale HCI framework and a Markov model to predict the average number of user actions (e.g., mouse clicks) required for task performance in order to determine an overall system effectiveness score.

Weighting factors for interactive system usability and performance dimensions were determined by a group of designers by deciding how important each link in the HCI framework is to communication between the human and computer. (Weighting factors can also be established for various usability paradigms, if interface alternatives represent different paradigms.) After the weighting factors are determined, OR techniques can applied for analyzing user behavior with prototype interface alternatives applying a particular usability paradigm (WIMP). In this case, an online ordering application was the subject of study. In

particular, Markov Chains were used to determine the average number of clicks it took for users to reach a particular interface goal state. Finally, end users were surveyed to rate the usability and performance of the design alternatives by considering all of the links in Abowd and Beale HCI framework (representing distances between the user language and interface and the system language and interface). These ratings were combined with the designers' weighting factors to produce a weighted sum for each design alternative. This result was then combined with the average number of clicks predicted by the Markov chain to yield an overall system effectiveness or usability score. The interface alternative resulting in the higher score from this methodology, which in this research was the new interface, can be said to have better usability. The overall system effectiveness score provided a mix of subjective and objective measures for determining the perceived usability per click.

In general, the above procedure is easy to implement and does not require the usability analyst to have advanced knowledge of Markov models. In the research by Thimbleby, Mathematica, was used with a prototype interface to record user actions and extract a state transitional probability matrix. Designers can then be consulted to develop a matrix of the average number of steps to reach a state *k* from a state *i* in a new design alternative. The usability analyst can then use the Kitajima equation to determine the average number of clicks to the absorption state for the new interface.

With respect to the present usability evaluation method, when the usability analyst conducts user testing to determine the transitional probability matrices, the Abowd and Beale framework should also be presented to the participants to determine ratings for the four dimensions of usability, articulation, performance, presentation and. Participants should also be

asked which interface elements they found most usable. This study used 10 participants and the Markov model accurately predicted clicks and the differences between the interfaces were accurately captured; therefore, at least 10 participants are appropriate for this methodology.

A very important aspect of this analysis is determining the designer ratings of the dimensions of system usability. These ratings help determine if the intent of the interface is appropriately achieved. Based on the present study, at least three designers should be surveyed. The designers should be presented with the Abowd and Beale framework and asked to complete pair-wise comparisons of the four dimensions of usability (these should be explained to the designers in layman terms). The system effectiveness score can be determined with all the available information for the original interface.

Once a prototype for the new interface is created, Mathematica can be used to find the average number of steps to reach a state *k* from a state *i*. The transitional probabilities found in user testing with the original interface can be multiplied by the predicted average number of steps to task completion (according to system designers) using the Kitajima equation to find the average number of clicks with the new interface. In order to collect ratings on the new interface on the four dimensions of usability, as specified by Abowd and Beale, users can be brought in for a focus group. The advantage of this approach is that less time and money may be required, as compared to actual user interface testing. However, it is important to note that since the user does not directly interact with the system, usability ratings may be less accurate than in actual testing. The overall system effectiveness score can then be calculated for the new interface and compared with that of the existing interface to determine if the new interface actually has improved usability.

This form of analysis could be instrumental in choosing among alternative interface designs. All designs would need to be prototyped but only one would require user usability ratings. The procedure described above would yield the overall system effectiveness scores for all interfaces so that the usability analyst can pick among the alternate interfaces. Creating prototypes for several interfaces takes a lot of work and resources so it is suggested that the analyst use task analysis as a basis to develop two to three interfaces to test with this method. If there are several alternatives, early paper prototypes and walkthroughs, could be used in selecting the two best interfaces for assessment with the present method.

In web applications, the time it takes a user to complete a task is very important. Several consequences may be incurred if the user does not find what they are looking for, right away, of if they encounter errors. The user may become frustrated and leave the site. Once the user has left the site, (s)he is likely to find the product from another supplier. This results in a lost sale and many future sales by the individual. This analysis does not take into account time on a task, which may be another important factor.

Using Markov models is sufficient for modeling web interface behavior, provided the information necessary to make a decision on the next interface action is all contained on the current page. Web applications are often designed in this fashion. In addition, other desktop computing applications, with which users may be highly experienced (e.g., Microsoft Word, Excel, etc.), may also be suitable for analysis with Markov models. More experienced users tend to navigate through application menus based on memory and not necessarily by gaining information from the current display screen in order to determine the next selection. An experienced user can bring prior knowledge that was gained from previous interactions with the

interface, but (s)he may not use knowledge gained during the current interaction from any other state except the current state to make a decision about the next state. If this is the case, the scenario of use does not violate the assumptions of the Markov model. That is, the Markov model is not entirely memoryless and knowledge at state 0 is acceptable for use in predicting a change in state *i* to *j*. With that said, applications and displays in other domains, such as aviation systems, are often not designed to reduce user cognitive load, like web pages, and often require pilots to remember novel information across system states and stages (e.g., Kaber, Tan, Riley and Endsley, 2001). This type of setup violates the basic assumptions of a Markov model and the number of actions to reach a goal state will not be accurately predicted for such interfaces. The Markov modeling also cannot be used in situations where a model on an original interface design is to be used to predict behavior with a radical redesign, where the majority of states of the old interface are no longer present in the new interface.

In the future, research should examine the use of Markov models to predict the most likely future action of a user and to create salient options to improve usability. The effectiveness of this design approach could be assessed using the same procedure presented in this research. It is also possible that there are other mathematical models that can be used to represent user behaviors with interfaces that do require the user to remember information from previous state and stages. Current research has been conducted on multiple dynamic models associated with Kalman filters for predicting human behavior more accurately than Markov models (Pentland and Liu, 1999). Another issue that could be explored is how to include time-on-task as a factor in the present usability measure. This information may yield itself to prediction using Markov models or perhaps some other stochastic process or mathematical model, such as a GOMS model

fuzzy set logic, or critical path models. A GOMS model can be used to predict times once actual observed times for each step in an application are recorded and averaged. Once the time is predicted or recorded, the number of clicks can be divided by time for a measure of clicks per time that would also show the effectiveness of the interface. The number of clicks per unit time can then be divided into partial scores to determine an overall system effectiveness score that incorporates the effects of time.

## REFERENCES


Baber, C., and Mellor, B. (2001). Using critical path analysis to model multimodal human computer interaction. *International Journal of Human Computer Studies, 54,* 613-636.

Abowd, G.D. and Beale, R. (1991). Users, systems and interfaces, a unifying framework for interaction. In D. Diaper and N. Hammond (Eds.), *HCI '91 People and Computers VI* (pp. 73-87). Cambridge, NY: Cambridge University Press.

Biström, J., Cogliati, A. and Rouhiainen, K. (2005). Post-WIMP user interface model for 3D web applications, Helsinki University of Telecommunications Telecom Software and Multimedia Laboratory.

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

Chang, E., and Dillon, T.S., (2006). A Usability-Evaluation Metric Based on a Soft Computing Approach. In *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans, 36*(2), 356-372.

Clamman, M.P. and Kaber, D.B. (2004). Authority in adaptive automation applied to various stages of human-machine system information processing. *Proceedings of the Human Factors and Ergonomics Society 47th annual meeting*. Denver, Colorado: Human Factors and Ergonomics Society, (pp. 543-547).

Cockton, G., Lavery, D., and Woolrych, A. (2002). Inspection-Based Evaluations. In J. Jacko, and A. Sears (Eds.), *The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications (*pp. 1118-1138). Mahwah, NJ: Lawrence Erlbaum Associates.

Dix, A., Finlay, J. E., Abowd, G. D. and Beale, R. (1993). *Human-Computer Interaction.* Europe: Prentice Hall.

Fischer, G. (2000). User Modeling in Human-Computer Interaction. *User Modeling and User Modeling and User-Adapted Interaction, 11*(1-2), 65-86.

Gould, J. D. (1988). How to Design Usable Systems. In M. Helander, T.K. Landauer, and P. Prabhu (Eds.), *Handbook of Human Computer Interaction* (pp. 757-789). North Holland: Elsevier.

Gray, W., John, B. E., and Atwood, M. E. (1992). The Précis of Project Ernestine or an Overview of a Validation of GOMS. In *Proceedings of the 1992 Conference on Human Factors in Computing Systems, CHI,* (pp. 307-312). New York, NY: ACM Press.

Hart, S.G. and Staveland, L.E. (1988). Development of NASA-TLX: Results and empirical and theoretical research. In P.A. Hancock and N. Mashkati (Eds.), *Human Mental Workload* (pp. 239-250). Amsterdam: North Holland Press.

Hornbaek, K. (2005). Current Practice in Measuring Usability: Challenges to Usability Studies and Research. *International Journal of Human-Computer Studies, 64,* 79-102.

Jenamani, M., Mohapatra, P. K. J., & Ghose, S. (2002). Online customized index synthesis in commercial web sites. *IEEE Intelligent Systems, 17*(6), 20-26.

Kaber, D. B., Riley, J., Tan, K-W and Endsley, M. R. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics, 5(1),* 37-57.

Karat, J. (1997). User-Centered Software Evaluation Methodologies. In M. Helander, T.K Landauer, P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 689-704). North Holland: Elsevier Science B.V.

Kitajima, M., Kariya, N., Takagi, H., & Zhang, Y. (2005). Evaluation of website usability using Markov Chains and latent semantic analysis. *IEICE Transactions on Communications, E88-B*(4), 1467-1475.

Kobsa, A. (2001). Generic User Modeling Systems. *User Modeling and User Adapted Interaction, 11*, 49-63.

Lewis, C. (1982). Using the "thinking aloud" method in cognitive interface design. *IBM Research Report.* Yorktown Heights, New York: IBM

Molich, R and Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM, 33,* 338-342.

Mosqueira-Rey, E., Rivela-Carballal, J., & Moret-Bonillo, V. (2004). Integrating GOMS models and logging methods into a single tool for evaluating the usability of intelligent systems. In *2004 IEEE International Conference on Systems, Man and Cybernetics, SMC 2004,* (pp. 5142-5147). San Diego, California: ACM.

Mulhem, P., and Nigay, L. (1996). Interactive information retrieval systems: From user centered interface design to software design. *Proceedings of the 1996 19th Annual International*

*ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 96*, (pp. 326-334). Zurich, Switzerland: ACM.

Nagasawa, S. (1999). Fuzzy kansei evaluation of VCRs' usability. In *1999 IEEE international conference on systems, man, and cybernetics 'human communication and cybernetics'*, (pp. 290-293). Tokyo, Japan: Institute of Electrical and Electronics Engineers Inc.

Newman, W. Lamming, M., and Michael, G. (1995). *Interactive System Design.* Boston, MA: Addison-Wesley Publishers Ltd.

Pentland and Liu. (1999). Modeling and Prediction of Human Behavior. *Neural Computation 11*, 229–242.

Ravindran, A., Phillips, D.T. and Solberg, J.J. (1987). Operations Research: Principles and practice. Wiley: New York

Rosson, M. B., and Carroll, J. M. (2002). *Usability Engineering: Scenario-Based Development of Human-Computer Interaction.* San Francisco, California: Morgan Kaufmann Publishers.

Roussou, M. (2000). Immersive interactive virtual reality and informal education. In Proceedings of the i3 Spring Days 2000-Workshop on Interactive Learning Environments for Children, (pp. ). Athens, Greece: ICS-Forth.

Shneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA: Addison Wesley Longman, Inc.

Sholl, H. A., Ammar, R. A., and Weiss, W. S. (1991). Using probabilistic finite state models to evaluate human-computer interfaces. In *Conference proceedings of the 1991 IEEE international conference on systems, man, and cybernetics,* (pp 1145-1150). Piscataway, NJ: IEEE

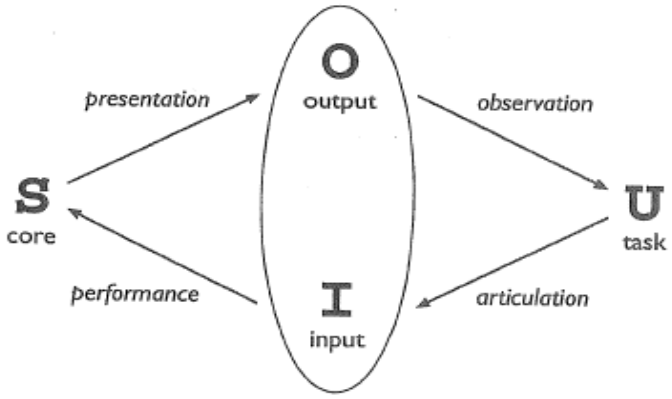
Thimbleby, H., Cairns, P., and Jones, M. (2001). Usability Analysis with Markov Models. *ACM Transactions on Computer-Human Interaction, 8*(2), 99-132.

Thimbleby, H. (2004). User interface design with matrix algebra. *ACM Transactions on Computer-Human Interaction, 11*(2), 181-236.

Wixon, D. and Wilson, C. (1997). The Usability Engineering Framework for Product Design and Evaluation. In M. Helander, T.K Landauer, P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp 653-688). North Holland: Elsevier Science B.V.

# APPENDICES

## Appendix A: Designer Survey




**Abowd and Beale Interaction Framework**

Each link between the nodes in the Abowd and Beale model represents a user or system language translation that affects overall system effectiveness.

- The link between the user and the interface input, articulation (art), represents how well the user's goal matches the input state of the system.
- The link between the input and the system, performance (perf), determines if each input has a logically mapped function.
- The link between the system and the interface output in the framework, presentation (pres), represents how accurately the system presents data to the user.
- The link between the output and the user, observation (obs), represents how well a user can interpret the output from the information display.

Consider the intent of the type of interface design under evaluation and circle the dimension of design you think is most important to achieving this intent among the pairs presented in each row of the below table.

art perf

art pres

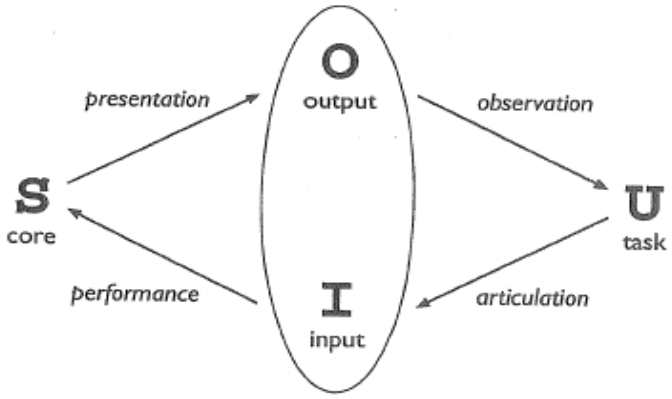
art obs

perf pres

perf obs

pres obs

## Appendix B: End User Survey




**Abowd and Beale Interaction Framework**

Each link between the nodes in the Abowd and Beale model represents a user or system language translation that affects overall system effectiveness.

- The link between the user and the interface input, articulation, represents how well your goal matched the input options available through the interface.
- The link between the input and the system (performance) and between the system and the output (presentation) in the framework represent the responsiveness of the interface to your commands.
- The link between the output and the user, observation, represents how well the output reveals the information you expected.

Based on your simulation experience, rate how well the interface addressed each of the links described above on a scale of 1 to 10, with 10 being extremely well.

1. Articulation: _____
2. Performance: _____
3. Presentation: _____
4. Observation: _____

Please check, using the check boxes, which elements of the interface you found most usable.

☐ Tabs

☐ Navigation

☐ Buttons

☐ Picture Gallery

☐ Other (please describe below): _____