Pauses in man-machine interactions: a clue to users' skill levels and their user interface requirements

Arbi Ghazarian

Department of Engineering,
College of Technology and Innovation,
Arizona State University,
7171 E Sonoran Arroyo Mall,
Mesa, AZ 85212, USA
E-mail: Arbi.Ghazarian@asu.edu

Arin Ghazarian

Independent Researcher,
475 E. Magnolia Blvd., Apt. A,
Burbank, CA 91501, USA
E-mail: arin.ghazarian@yahoo.com

Abstract: Computer users have motionless periods of time while performing computer-based tasks. Do these pauses relate to the mental and perceptual actions required to perform tasks? Do users essentially pause while they think, wait to retrieve the next step to perform, or search the location of something on the screen? How do the pauses change as the users gain experience and progress from novice to skilled? To answer these questions, we conducted user studies to investigate the link between the pauses observed in users' interactions with computer-based applications and their skill levels. In this paper, we introduce a set of pause-related attributes that can distinguish among different levels of skills in performing Graphical User Interface (GUI) tasks. We employ machine learning algorithms to build skill classifiers based on these attributes. These skill classifiers can be used to create skill-adaptive applications.

Keywords: adaptive user interfaces; pause analysis; skill classifiers; user modeling.

Reference to this paper should be made as follows: Ghazarian, A. and Ghazarian, A. (2013) 'Pauses in man-machine interactions: a clue to users' skill levels and their user interface requirements', *Int. J. Cognitive Performance Support*, Vol. 1, No. 1, pp.82–102.

Biographical notes: Arbi Ghazarian is an Assistant Professor of Computer Science and Software Engineering at Arizona State University. He is also on the faculty for the Simulation, Modelling and Applied Cognitive Science at Arizona State University. He received his PhD in Computer Science from the University of Toronto in 2009. His research interest is primarily in software engineering—in particular, requirements engineering, the application of cognitive theories in software engineering, software traceability, software maintenance and evolution, software comprehension, software reliability, software design and architecture and empirical studies in all of these areas. He has published numerous journal and

conference papers and regularly serves on international technical program committees. In addition to his academic background, he has 15 years of professional experience in the software industry and has been involved in numerous large-scale industrial software projects in various technical and managerial capacities. Ghazarian is also a certified Project Management Professional (PMP). He is a member of the IEEE and the Project Management Institute (PMI).

Arin Ghazarian received his MSc in Computer Science from the Amirkabir University of Technology in 2008. His primary interests lie in the areas of human–computer interaction, machine learning and image processing. He is a senior software consultant specializing in Java and big data.

1 Introduction

Skill levels of computer users are among the most important factors that impact their performance in performing computer-based tasks. The adaptation of user interfaces (UIs) to the skill levels of individual users improves their performance (Benyon, 1993; Trumbly et al., 1993). Computer users vary greatly in their skill levels and expertise. Despite having a particular level of system or general skill, each individual user has different skill levels across different applications and even across different portions of the same application (Fischer, 2001). Moreover, a user's skill level changes over time, as the user gains experience with an application and learns or forgets various features of the application. Because of this great variation in users' skills, a single user interface will not satisfy the needs of all users. To address this problem, adaptive systems monitor the history of users' interactions and attempt to automatically adjust their interfaces or contents to accommodate user differences in aspects like skill. It goes without saying that the development of successful adaptive systems strongly relies on the capability to identify attributes in user behaviour that can accurately detect their skill levels.

Among other possible attributes for detecting users' skill levels, in this paper, we specifically focus on pause-related attributes and measure how expert computer users (i.e., users with a system skill level of expert/skilled) progress from novice to skilled in performing a specific computer-based task. Throughout this paper, we will use the term "task skill" to indicate the skill level of a user in performing a specific task or step in an application.

We report on experiments conducted to investigate the links between the pauses observed in users' interactions with Graphical User Interfaces (GUIs) and their skill levels. Some previous work has used attributes related to users' pauses during their interactions as skill indicator features. However, there is inconsistency in using the term pause in the related literature. Some works, such as Santos and Badre (1994) and Nakamura et al. (1996), have considered the time interval between every two subsequent actions as a pause and not the absolute motionless periods in which the users do not even move the mouse and no physical interaction takes place at all. On the other hand, there are research works, such as Reeder and Maxion (2006), that have considered only absolute motionless periods during the users' interactions as pauses.

The contributions of this paper are two-fold: first, we introduce a number of new attributes related to users' pauses during their interactions with GUIs and demonstrate that they can be used as skill indices. Second, we present skill classifiers for desktop applications, which are created based on these pause-related attributes.

The skill classifiers presented in this paper can be used to build skill-adaptive applications. Our classifiers are built using supervised machine learning algorithms. In our experiments, user interface event streams generated as a result of users interactions with a user interface were logged and pause-related attribute values were extracted from these UI events streams. These attribute values were then used as inputs to our classifiers. The proposed method is capable of detecting skilled versus novice performances of users using a short-term history of the users' interactions with a GUI. The approach presented in this paper operates in both application- and task-independent fashion.

The rest of this paper is organised as follows: In Section II, we review related works and discuss the concepts of skill and skill acquisition. Section III discusses the procedures used in our experiments and Section IV presents the results. We conclude with a summary of our findings and directions for future work in Section V. All tools and data from our experiments are accessible to interested readers.

2 Background

Accomplishing a task with a minimum outlay of time and effort is essential to the skilled performance of that task. In the Human-Computer Interaction (HCI) literature, the terms expertise and skill are often used interchangeably. In this work, we aim to model skill. Skilled behaviour is one of the distinguishing characteristics of experts. Skills are learned with practice and experience. Novice users perform tasks by recognition, i.e., they use knowledge in the world to plan and accomplish tasks, whereas skilled users use knowledge in the head to accomplish tasks (Norman, 1988). The concept of experience generally refers to know-how or procedural knowledge, rather than propositional knowledge. Skills are stored in procedural memory. The procedural memory encodes procedures or algorithms rather than facts (Sun and Giles, 2001). In what follows, we will discuss previous work under two major related topics: (a) skill acquisition and prediction models and (b) skill classifiers.

Skill Acquisition and Prediction Models: A number of models have been developed in cognitive science to explain the development of skill levels from novice to expert. For example, "The Power Law of Practice" states that the logarithm of the reaction time for a particular task decreases linearly with the logarithm of the number of practice trials taken (Newell and Rosenbloom, 1981). Most expertise development models, including Hacker's action theory, Fitts's skill acquisition theory, Rasmussen's decision ladder theory and Anderson's ACT* theory consider the following three stages in the skill acquisition process (Hacker, 1994; Rasmussen, 1983; Anderson, 1982; Anderson, 1983; Fitts, 1964):

1. ***Knowledge-based Level:*** This is the level at which users use their mental capacity to solve a problem that is not predefined. To accomplish a given task, they need to set goals and organise their movements towards the solution. The procedures that are used tend to be memorised and employed as rule-based behaviour, if the situation recurs. This stage demands a significant amount of resources in terms of time and working memory requirements.
2. ***Rule-based Level:*** Actions are defined and users apply the actions consciously going through a predefined procedure. In this stage, a user, having identified the state the system is in, chooses an appropriate rule from a rule set, which then influences the appropriate response. According to the ACT* theory, repetition of this stage causes knowledge compilation to occur, which, in turn, causes transition to the procedural stage.

3. ***Skill-based Level (procedural stage):*** Actions are smooth and highly integrated; they are performed in a nearly unconscious and automated fashion by the users without significant mental work during the realisation of the task. In the procedural stage, a user performs through quickly retrieving the procedures (i.e., production rules) without declarative details, which were lost during the knowledge compilation stage.

Measuring changes in behaviour as a result of repeating a task has been used by many researchers to model or measure the skill acquisition process and to understand how people get experienced in performing tasks in different domains (Crossman, 1959; Fitts and Posner, 1968; Neves and Anderson, 1981). A main characteristic of skilled behaviour is smooth and continuous operation. To operate smoothly, users should have fewer pauses of shorter durations. Pauses are points at users' interactions where no physical actions take place (e.g., no mouse movements). When interacting with a computer, users' behaviours typically follow an acquisition–execution cycle. During the acquisition phase, users create a plan, which is then performed during the execution phase (Card et al., 1980). During execution, the retrieved execution plan is stored in the user's short-term memory in coherent units called chunks (Badre et al., 1993). This acquisition–execution cycle repeats until the user's goal is completely accomplished. In the acquisition phase, users are merely involved in problem solving and typically, no physical interaction occurs with computers. These motionless periods of time appear as pauses in the observed behaviour of the users (Santos and Badre, 1994). It is during these pauses that the users retrieve the next chunk to be performed.

The size and contents of a mental chunk can be inferred from the observed behaviour of a user. Novice users form smaller chunks with fewer elements per chunk, while experts form larger chunks in shorter periods of pausing, with less frequency and less variability (Badre et al., 1993; Chase and Simon, 1973; Barfield, 1986). (See Badre et al., 1993; Chase and Simon, 1973; Barfield 1986; Reitman 1976; Badre 1982; and Badre, 1982, for more details on chunk-related differences between experts and novices.) In many routine computer usage tasks like word processing, the lengths of pauses are fairly uniform (Card et al., 1983; John and Kieras, 1996).

In the GOMS (Goals, Operators, Methods and Selection rules) task analysis techniques, which model user behaviour while performing computer tasks, the M cognitive operator is placed before every few actions (Card et al., 1980a). The places where M operators are inserted can be regarded as the points where chunking happens. The GOMS techniques model a user's behaviour while performing routine computer tasks. GOMS breaks down a user's interaction with a computer into its elementary physical, cognitive, or perceptual actions or operators. These models can estimate the time needed for the user to perform a specific task. GOMS derivative models, such as CMN-GOMS and NGOMSL, further break this M into more specific unobservable operators. GOMS-related models are extensively discussed in John and Kieras (1996); Card et al. (1980a); Card et al. (1980b); Salvucci (2009); Kieras (1988); Kieras (in press), Gray et al. (1992); and Anderson (1993).

There is also a large difference between the novices and experts in the number of Ms required. Experienced users spend little time in memory retrieval or screen searches; they know where everything is located. New users, on the other hand, will stop and check feedback from the system after each action, which in itself takes an m. Experienced users skip the verification step and jump to the next action. Moreover, experienced users can overlap mental operators with physical operators. An apparent characteristic of highly practiced performance is the ability to do more than one thing at a time, if it is physically possible. For example, a practiced user might be able to visually locate an icon on the screen

while starting the mouse movement, which results in less M operators (Buxton, 1995; John et al., 2004; Kieras, 1993).

A movement to a target in pointing tasks consists of a sequence of sub-movements (Keates and Trewin, 2005; Meyer et al., 1988; Hwang et al., 2004). Pauses during the mouse movements can relate to these sub-movements. Where a pause is observed during a pointing task, a sub-movement break also occurs at that point. These pauses may indicate periods of motor planning based on visual feedback. For example, the movement optimisation model proposes that movement to a target consists of an initial movement, which covers the majority of the distance to the target, followed by an optional secondary corrective sub-movement that homes in on the target (Meyer et al., 1988).

Researchers from different areas of HCI, such as natural dialogue systems (both speech-based and text-based), educational and e-learning systems, adaptive hypermedia systems and intelligent/adaptive help systems, have been interested in modelling users' expertise/skills and adapting user interfaces to users' skill levels.

In most previous research on expertise modelling, medium or low-frequency events such as help counts, error rate, number of previous visits to a specific feature in a UI, shortcuts, advanced features usage and the number of successful task completions have been used to measure user expertise. Much less research has been conducted on modelling users' skill levels from low-level high-frequency UI events such as mouse moves. The duration of HCI events ranges from less than one second to several years. Event types of shorter duration such as UI events can occur much more frequently and thus, might be referred to as high-frequency band event types. Event types of longer duration, such as sending emails to team members in computer-supported collaborative works, can be referred to as low-frequency band event types (Sanderson and Fisher, 1994). In contrast to most previous works that use low frequency user interface events such as help referring count, a main characteristic of our work is that the features used as skill indices are based on high-frequency user interface events.

In Huang (2003), the pauses during interaction with a command prompt system have been investigated. In command prompt systems, there is no ambiguity regarding aimless movements of the mouse in the pause periods, because there is no mouse input. In these systems, the time intervals between every two consecutive keystrokes that are greater than a specific duration are considered as pauses.

An interesting question raised in some related research works is: How long should a user be motionless or action-less for it to be considered a pause? Different values have been used in different experiments and domains. Santos and Badre (1994) describe an algorithm to detect the user's chunk boundaries by an analysis of the pause lengths between every two consecutive operations, such as keyboard presses or mouse clicks. In their work, pauses were regarded as indicators of mental chunks. Due to the fact that expert users make larger chunks, they concluded that their algorithm can be used to detect users' expertise using the chunk size. They measured the pauses in subjects' interactions while performing a variety of partially-defined goal-oriented tasks.

Nakamura et al. (1996) found that the variance of the operation time interval is a useful index in determining the skill levels of users. They defined operation time interval as the time between the completion of one operation and the start of the next (i.e., the time interval between two consecutive operations). This interval consists of system response time, user thinking time and moving time (time taken to move the mouse or to move the hands to the keyboard). Response time and moving time were found to be short compared to thinking time. Therefore, they concluded that time interval mostly consists of thinking time.

Huang (2003) analysed the pausal behaviour of users when issuing search commands using a command line search system for searching CD-ROM contents. A pause was defined as a discernible stop of three or more seconds during the time when a user is issuing search commands by typing them using a keyboard. The study investigated only pauses that occurred while a searcher was issuing commands and not the pauses that occurred while watching the search results. The study reports results regarding the frequency of pauses, duration of pauses, reasons for pausing, location of pauses, relationship between reasons for pausing and length of pause, the amount of information being processed as chunks at each pause and changes in pausal behaviour over time as searchers gained experience with searching tasks. The study also reports a metric called hesitation rate. The hesitation rate is defined as the ratio between pausing and issuing commands, i.e., the total pausing time divided by the total user input time. The higher the ratio, the more time is spent in pausing. Results from this study demonstrated that searchers paused less frequently and for shorter periods of time as they progressed through searches and gained more experience and practice. Further, the hesitation rate decreased over time with more practice.

Hesitations have been used to determine periods of user difficulty in interacting with computers (Reeder and Maxion, 2006; Maxion and deChambeau, 1995). In Reeder and Maxion (2006), the authors present an automated method for detecting instances of user difficulty based on identifying hesitations during users' interactions with a system. Hesitation is defined as anomalously long pauses in users' interactions with the mouse and keyboard. Anomalously long pauses are identified by computing the latency between every pair of consecutive events in the data stream and outputting those latencies that exceed a certain threshold. The individual differences in mouse and keyboard activity were taken into account by computing a latency average and standard deviation independently for each user. They used their method to detect novice users' difficulties in performing GUI-based goal-directed short tasks. They report that their proposed method works well for identifying GUI defects in those tasks which are goal-directed, have little typing, have short completion time and have limited text to read.

Skill Classifiers: Hurst et al. (2007) built a decision-tree-based classifier, which is able to classify users' actions as novice or skilled behaviour with an accuracy of 91%. To identify users' skill levels, their classifier only used the interactions that occurred while users worked with menus in an application. They used attributes related to high frequency UI events such as mouse motion velocity, mouse motion acceleration, menu item visit counts and selected menu item dwell time. The detected skill level was used to provide tailored intelligent help to users. To collect training data for their classifier, they asked subjects to perform a specific paint task with a drawing software for seven trials. The repetition would cause the users to progress from novice to skilled. All UI events were logged during their interactions and attribute values were extracted from these logs. Data from the first trials were labelled as novice, while the seventh trials were labelled as expert. Their classifier operated in an application-independent fashion and without a prior knowledge of the task. However, its operation is limited to situations where users interact with menus in applications that contain menus.

Leung and Fulcher (1997) used neural networks to classify users' expertise levels while using a text editor. They used multi-layer perceptron classifiers with output data fuzzification to classify the users into one of five expertise levels. Only keyboard inputs were logged and used. Attributes such as pause times, number of keystrokes, number of advanced features

and hotkey (a.k.a. shortcut) usages were used as input parameters to their classifiers. They achieved a classification accuracy of about 80%.

Other researchers have studied pausal behaviour in silent reading, writing and problem-solving as a way of determining decision making (Foulin, 1998; Tavalin, 1995). Ghazarian and Noorhosseini (2010) created automatic skill classifiers for desktop applications. They used machine learning algorithms to build statistical predictive models of skill. Two categories of skills were considered for users: system skill (i.e., general skill in using computers) and task skill (i.e., skill in performing a specific task in an application). Their classifiers are capable of detecting both system skill levels and task skill levels. Attribute values were extracted from high frequency user interface events, such as mouse motions and menu interactions and were used as inputs to their models. To collect training data for their classifier, they asked novice and skilled system skill users to perform a specific paint task with a drawing software for 15 trials. The repetition would cause the users to progress from novice to skilled in performing that specific task. All their interactions were logged automatically. Attributes were extracted from these log files and used for building classifiers. Although they used some pause-related attributes, such attributes were not central to their study. In contrast, in this paper, we specifically focus on pause-related attributes.

3 Experimental procedures

The main objective of the experiment conducted in our study was to investigate the connections between the observable pauses in users' interactions with GUIs and their skill levels. We were not necessarily looking for performance improvements in skill classification merely using pause-related attributes, but rather to explore the usefulness of pause-related attributes in building skill classifiers. To accomplish this objective, we asked 15 subjects to perform a specific task for 15 trials. Subjects were expert computer users, mostly consisting of IT professionals and were selected through an interview. The average subject age was 29.5. All subjects were right-handed and none reported any kind of visual impairments. All experiments were conducted in a typical office environment and in the afternoons. Each subject worked on their tasks in a single experimental session. Subjects were briefed and gave consent before participating in the experiments. While performing the task trials, all subjects' interactions were automatically logged. The experiment involved a rather simple task with a paint software called TerpPaint [50]. The paint software used is an open source image editing software implemented in Java. To ensure all subjects interacted with the GUI under similar conditions, we fixed the size and the location of the window of the paint program and did not allow the subjects to change these settings. The task was purposefully designed to make the subjects interact with the most common GUI interactors (i.e., GUI components) such as popup menus, buttons, combo boxes, etc.

The task involved a seven-step image manipulation. It consisted of (a) opening a default image; (b) performing some manipulations on it, including selecting a region of the image, copying the selected region, pasting the selected region, moving the pasted region to another location on the image; (c) picking up a particular colour from the image using the colour pick tool, selecting the stroke style and drawing a tick mark on the image using the selected colour and stroke style; (d) writing text on the image at a specific location using the specified font type and size; (e) applying a particular imaging filter; (f) saving the resulting image; and finally, (g) exiting the program. The events that were logged during subjects' interactions

were mostly high-frequency UI events such as mouse moves, menu selections, button presses, etc.

Participants were given a document containing instructions on how to perform the task. They were allowed to refer to it whenever needed during the experiment. In addition, they were given training on the task before performing the task. All subjects performed on PCs with similar configurations. We asked the subjects to remain concentrated while performing the task and to try to conduct the task in a natural and efficient way. They were told that they should work steadily at the task. Subjects were asked to perform the specified task for 15 trials. We used repetition to observe and measure how the users progress from novice to skilled. Users were given a rest after the seventh trial. This was done to block or minimise the effect of fatigue on performance after a high number of trials.

To log the UI events occurring during the users' interactions with the paint software, we developed our own software tool called (JSpoor). JSpoor is capable of logging all UI events such as mouse moves, keyboard presses, menu selections and button presses. JSpoor was developed using Java and AspectJ programming languages. JSpoor logs each UI event with a set of related data. For instance, the type of the event, the location of the mouse cursor, the timestamp at which the event occurred and the name of the GUI component where the event occurred were some of the data items that were recorded during users' interactions.

JSpoor also has a feature extraction module, which is capable of extracting attributes such as mouse velocity, pause counts, pause durations, etc. JSpoor calculates and prints attribute values in arff format (Witten and Frank, 2005), to be used as input to a (Weka) data mining tool. JSpoor and its source code are available at http://sourceforge.net/projects/jspoor.

In addition to automatic data collection, we used shadowing to gain further insight into where, how long and why the users pause. The data analysis process was further facilitated by the replay functionality of JSpoor; it allowed us to replay the logged interactions and review the users' interactions. JSpoor also visualises the mouse path and shows where and when pauses happen along with other useful information while replaying the log files. We watched all subjects' interactions again using this tool.

Due to technical problems, we had to discard logs for two subjects from our dataset. The data from 13 subjects, including 6 female and 7 male users, were used in the calculations. A shadowing data collection technique was also used; we observed users during their performances and recorded any incorrect steps they performed as well as problems or irregular interruptions they encountered. These problematic steps were eliminated from the calculations by explicitly providing the trial number and the step number of the noisy steps to JSpoor via a file. We purposefully did not take into account the data from the first step of any trial. The reason was that we could not discern when the users had started the first step of that trial. For other steps (from 2 to 7), we assumed that each step started when the previous step completed. Also, we removed data related to steps 6 and 7, which consisted of selecting well-know and common menu items (e.g., save and exit). We removed the data from these steps to block the effect of transferable skills on the outcomes of the experiment. Therefore, steps 2 through 5 provide the main data for our computations.

4 Experimental results

Our observations during the experiment are consistent with the related literature regarding the locations and the reasons of pausing. For example, we observed that in the early trials,

subjects had aimless mouse motions accompanied with pauses before every step, i.e., they would move the mouse aimlessly, then pause for some time, then they would move the mouse aimlessly again and so on. Both the related literature and our measurements indicate that users pause less frequently and for shorter periods as they become skilled in performing a task.

A notable observation during the experiments was that when users became skilled after many trials, they did not wait until the visual effect of the action becomes visible to continue to the next step. Instead, immediately after performing an action (e.g., selecting a menu item) and before verifying the effect of that action, they would move toward the next target to perform the next action. This phenomenon could have contributed to the less-frequent and shorter pauses during the high trials.

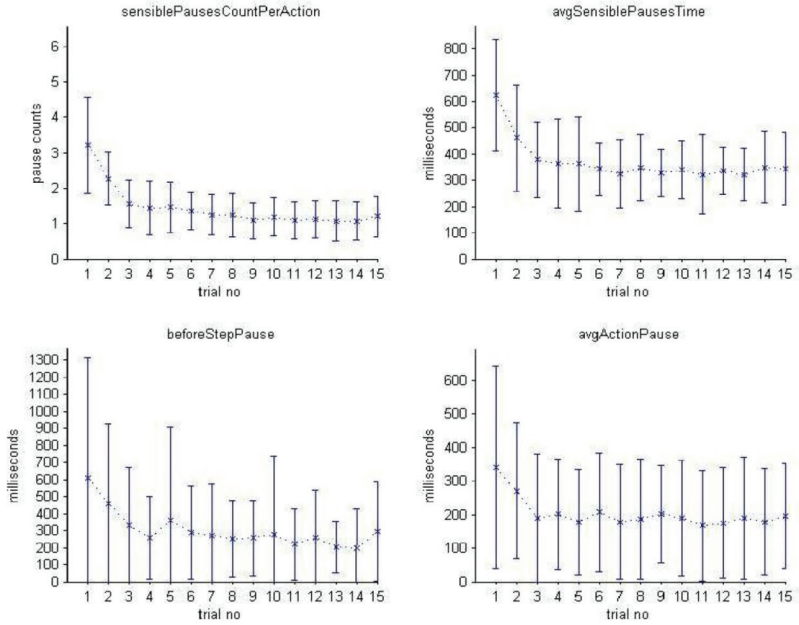
We extracted attributes related to users' pauses from the interaction logs, which are described in Table 1. The mean values of these attributes across the 15 trials are depicted in Figures 1–4. Some of these attributes relate to absolute motionless periods, while others relate to the time intervals between every two consecutive actions such as mouse button presses.

Instances from the first trials of the users were labelled novice, from the second trials were labelled as intermediate and from trials 14 and 15 were labelled as skilled. These decisions emerged from the statistical analyses of the subjects' performance data across trials. The differences in mean performance times between the identified task skill groups were statistically significant. We tried other possible combinations for mapping trials to skill levels, but the results were not significant. In the intermediate trials, the performances were not yet steady, so we did not use the instances from these trials in training the classifiers.

Table 1 Pause-related attributes

Attribute name	*Description*	*Unit*
beforeStepPause	Pause time before step (motionless period right after finishing a step and before starting the movement for the next step). This is the absolute motionless pause.	millisecond
beforeFirstActionSensible PausesSum	The sum of absolute sensible pauses before the first action of each step (and after finishing the previous step). This is absolute motionless pause.	millisecond
beforeFirstActionPauseTime	The time duration after finishing a step and before starting the next step. This is time between two subsequent activity pause	millisecond
beforeFirstClickDistanceRatio	Ratio of the distance travelled by the mouse to the minimum distance required to move the mouse to perform the first action of the step. This is related to time between two subsequent activity pause.	unitless
avgActionPause	Average duration of the pauses happening after the actions (such as after pressing buttons). This is the absolute motionless pause.	millisecond
sensiblePausesCountPerAction	Count of sensible pauses (i.e. pauses longer than 180 ms). This is the absolute motionless pause.	unitless
avgSensiblePausesTime	Average duration for sensible pauses (i.e. pauses longer than 180 ms). This is the absolute motionless pause.	millisecond

Figure 1 The average values for sensiblePausesCountPerAction, avgSensiblePausesTime, beforeStepPause and avgActionPause attributes across trials with deviation bars, pause duration >180 ms (see online version for colours)



It was observed that pauses longer than 180 ms had the greatest information gain. As we stated earlier, we divided the entire task into seven steps. During the experiments, we observed that subjects would generally retrieve related actions together; they often had a long pause before every few related actions in which they would acquire the next few related actions to execute.

Users learn every few related actions as a single mental chunk. For example, the subjects would acquire the selection, copy and paste actions together. The

Figure 2 The average values for beforeFirstActionSensiblePausesSum sum across trials with deviation bars (see online version for colours)

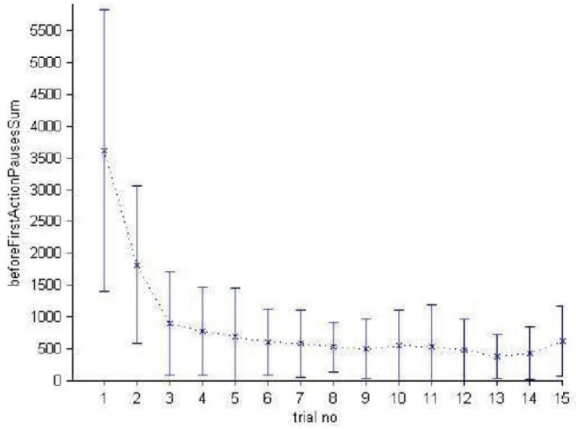
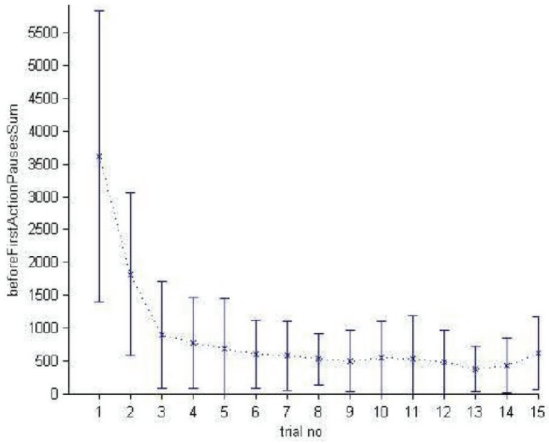
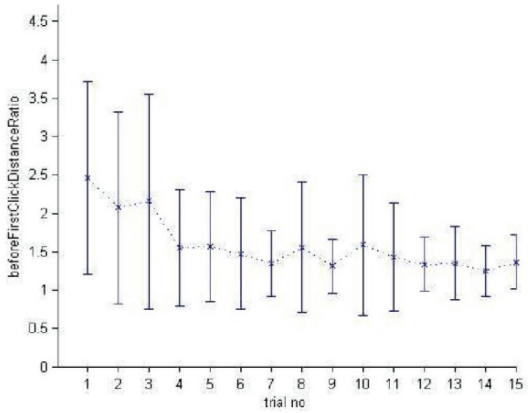


Figure 3 The average values for beforeFirstActionPauseTime sum across trials with deviation bars (see online version for colours)



sensiblePausesCountPerAction attribute measures, on average, how many times a user pauses to perform each action. *avgSensiblePausesTime* is the average duration of these sensible pauses. *sensiblePausesCountPerAction* and *avgSensiblePausesTime* only consider those interruptions in users' actions as pauses that take longer than 180 milliseconds. As we stated earlier, an important parameter in the analysis of users' pauses is the minimum duration of pause, i.e., what is the minimum amount of time that the user should be motionless or action-less for it to be considered a pause? To obtain this minimum sensible pause duration quantity (180 ms), we first asked some users to perform a number of tasks with a computer. Their interactions were logged to files and analysed. We observed that the minimum duration of a pause, which was sensible (i.e., it was detectable by a human subject), was 150 ms or longer. To optimise this skill index and to further tune it, we tested the values around 150 ms to ensure that we selected an optimised minimum pause duration value. We computed the information gain for the different minimum pause duration values within the range 100–250 ms by an interval of 10, i.e., 100, 110, 120, through 250. The aim was to choose the optimal

Figure 4 The average values for beforeFirstClickDistanceRatio sum across trials (see online version for colours)



minimum pause duration so that the resulting *sensiblePausesCountPerAction* attribute would better discriminate between the novice, intermediate and the skilled performances. We refer to the pauses greater than 180 ms as *sensible pauses.*

The first attribute shown in Table 1 is *beforeStepPause.* It was observed that after completing every step and before starting the next step, users had a noticeable pause to retrieve the next step. The *beforeStepPause* attribute measures this type of pauses. In these pause periods, users mostly retrieve the next step to perform, i.e., this pause time is related to the mental chunking. *avgActionPause* represents the average duration of pauses that take place after each action. Events such as dragging, selecting menu items and menus, pressing buttons, etc., are considered as actions. *avgActionPause* mostly measures the time taken by the users to verify the results from a previous action. It can also relate to the time required to find an item on the screen, or chunking the next action.

It can be seen in Figure 1 that the *sensiblePausesCountPerAction* attribute is superior to the other pause attributes in showing the differences across the trials. The average distribution of pauses per action with different durations in trials 1, 2 and 15 are shown in Figures 13–15, respectively. We also fitted the mean values of *sensiblePausesCountPerAction* to the power law functions; the correlation coefficient was 0.908. The fitted curve is depicted in Figure 5.

The time required to accomplish a task is the most widely used index to measure skill level. In general, the curve of the task completion time versus the trial numbers follows the Power Law functions (Newell Rosenbloom, 1981). We created the KLM-GOMS model for the task and calculated KLM-GOMS task time predictions for the task steps. We used a tool called (CogTool), which facilitates the creation of KLM-GOMS models. Step completion times were divided by the KLM-GOMS time predictions. This new attribute is called *KLM-Ratio* (Kurosu et al., 2002; Hurst et al., 2007). By dividing the step times by the KLM-GOMS times we normalised these values. Normalisation is done to make the range of attributes from different tasks and steps identical or to make the attribute values from different task steps comparable to each other. For example, a value of two for the *KLM-Ratio* attribute means that the task completion has taken twice as long as the KLM-GOMS predicted time and a value of one means that the user has performed the task as fast as the KLM-GOMS

Figure 5 SensiblePausesCountPerAction fitted to a power law function (see online version for colours)

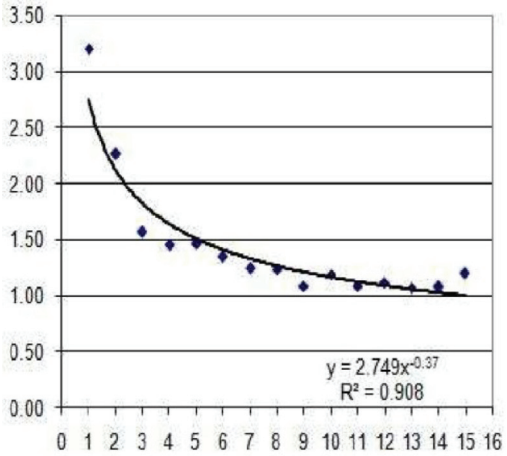


Figure 6 Mean KLM-Ratio values across trials with deviation bars (see online version for colours)

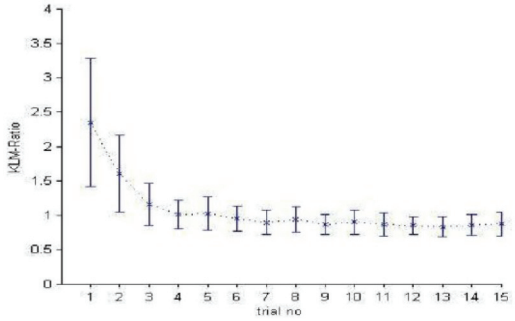
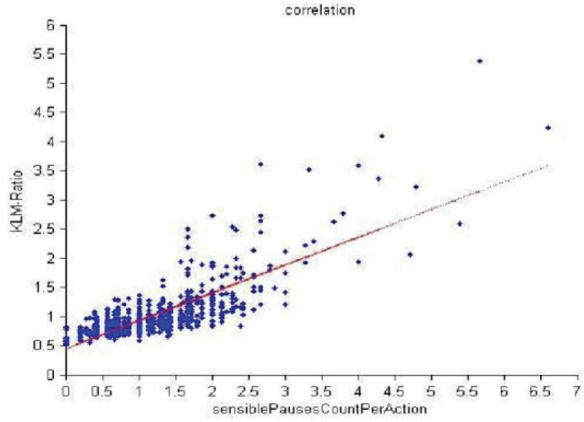


Figure 7 Correlation between KLM-ratio and sensiblePausesCountPerAction (see online version for colours)



predicted time. Figure 6 shows the mean values for *KLM-Ratio*s across the trials. When we fitted this curve to power law functions, the correlation coefficient was 0.872. Figure 7 shows the correlation between *sensiblePausesCountPerAction* and *KLM-Ratio.* As depicted in this figure, there is almost a linear relation between them. The correlation coefficient between *sensiblePausesCountPerAction* and *KLM-ratio* was 0.78398.

Further, to make it possible to compare our attributes with pause-related attributes used in the previous works, we computed the average values for hesitation ratio, average verification pause and operation interval variance (explained in Section II). These values are depicted in Figures 8–10. A comparison was made among all attributes from the classification ability point of view, i.e., how well each attribute is able to classify the three skill levels, namely the novice, the intermediate and the skilled levels. The metrics used in this comparison was information gain (Mitchel, 1997). The results are shown in Table 2. The information gain method ranks attributes according to their entropy reduction property. KLM-ratio's information gain was 0.78 +/–0.036.

Further, an analysis of variance (ANOVA) was performed to obtain the significance values (p-value) for these attributes for the three groups of skill levels. The first group consisted of the data related to trial 1 (novice skill level), the second group consisted of the data related to trial 2 and the third group consisted of the data from trials 14 and 15 (skilled

Figure 8 Mean values for *hesitationRatio* across the 15 trials. *hesitationRatio* is defined as the total pausing time (we considered absolute motionless pauses) divided by the total task completion time (Huang, 2003). When we fitted the curve to the power law function, the correlation coefficient was 0.873 (see online version for colours)

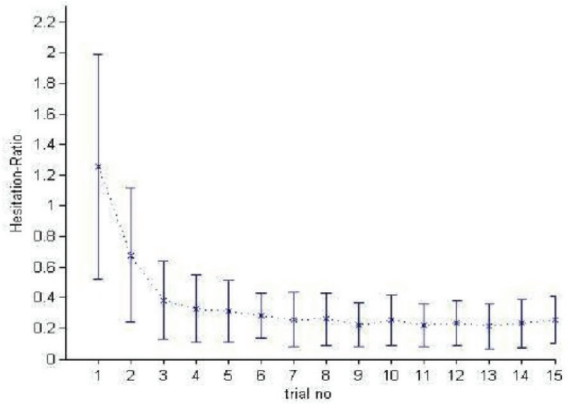
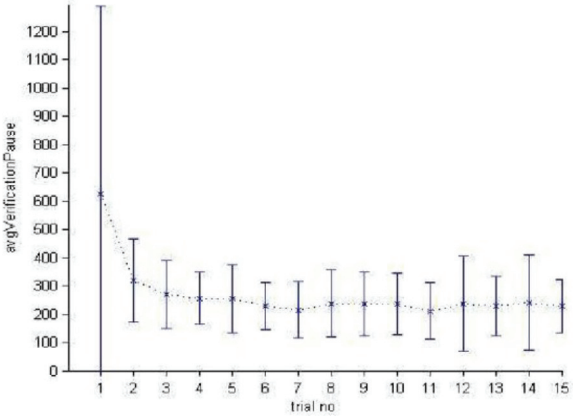


Figure 9 Mean values for *avgVerificationPause* across the 15 trials. *avgVerificationPause* is defined as the duration of the pause which happens right before clicking on the target (Keates and Trewin, 2005; Hwang et al, 2004). Absolute motionless pauses were considered. When we fitted the curve to the power law function, the correlation coefficient was 0. 701 (see online version for colours)



skill level). Single factor ANOVA was used to compute the significance of each attribute. A significance level of $p < 0.001$ was adopted, where only p values less than 0.001 were considered significant. The results of this analysis are summarised in Table 3. An advantage of pause-related attributes, in comparison to non-pause-related attributes, is that despite their high information gain, which is almost equal to the KLM-ratio, modelling skill using pause-related attributes is simpler; they are easily extracted from UI event streams and can be measured dynamically and in real time. Table 3 shows the results of ANOVA between novice and skilled groups (the intermediate group is removed).

Figure 10 Mean values for *operationIntervalVariance* across the 15 trials. *operationIntervalVariance* is defined as the variance of the time intervals between consecutive operations in performing a task (Nakamura et al, 1996). The time between two subsequent activities is considered as pause. When we fitted the curve to the power law function, the correlation coefficient was 0. 824 (see online version for colours)

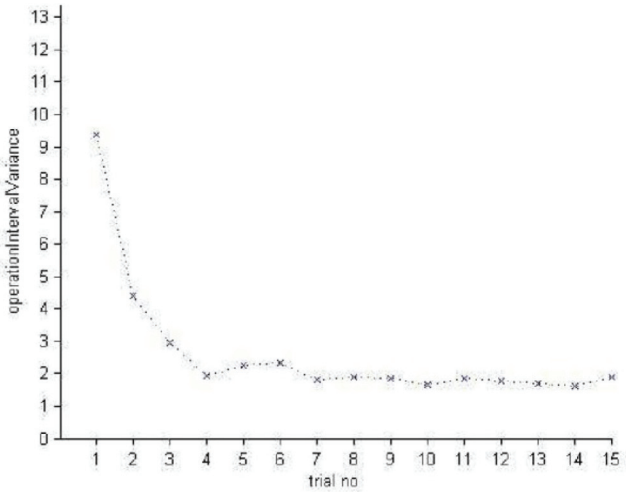


Table 2 Information gains

attribute	*Information gain*
before First Action Pause Time	0.631 ± 0.039
hesitation Ratio	0.614 ± 0.039
sensible Pauses Count Per Action	0.606 ± 0.039
before First Action Sensible Pauses Sum	0.522 ± 0.04
before First Click Distance Ratio	0.268 ± 0.022
avg Action Pause	0.267 ± 0.023
avg Sensible Pauses Time	0.236 ± 0.023
avg Verification Pause	0.203 ± 0.019
before Step Pause	0.011 ± 0.032

Table 3 Results from ANOVA between novice and skilled trials groups, p = 0.001

attribute	*P*
hesitationRatio	1.76E–22
sensiblePausesCountPerAction	1.47E–24
beforeFirstActionPauseTime	1.54E–27
beforeFirstActionSensiblePausesSum	4.22E–23
avgActionPause	1.54E–08
beforeFirstClickDistanceRatio	2.75E–09
avgSensiblePausesTime	3.37E–12
avgVerificationPause	3.3E–07
beforeStepPause	0.000281

5 Training and evaluating skill classifiers

We built classifiers that can detect users' skill levels using pause-related attributes. Machine learning techniques were utilised. The training instances comprised of values of the pause-related attributes defined in the previous section, which were extracted from the interactions logs for the task steps from all trials. Each instance was assigned a task skill level label according to its trial number. Instances from the first and second trials were labelled as novice and intermediate, respectively. Instances from trials 14 and 15 were labelled as skilled. These labelled instances were then used as input for training the classifiers. Our aim was to create a single skill classifier that is capable of detecting the skill levels of different users within different tasks and applications. The Weka [53] data mining tool was used to build the skill classifiers. The C4.5 decision tree machine learning algorithm (Quinlan, 1993) was selected. An advantage of decision trees is that they are easily interpretable by human analysts. To measure the accuracy of our classifiers, we used the 10-fold cross-validation method. The ranges of decision tree training parameters were selected in such a way as to avoid the over-fitting of the tree, i.e., we didn't sacrifice extensibility for accuracy. The accuracy of the resulting classifier was 62.68%. This classifier is visualised in Figure 11. Note that, as mentioned earlier, the purpose was not to achieve higher classification accuracy compared to previous classifiers, but rather to evaluate the usefulness of pause-related attributes in building skill classifiers. Our results suggest that although a classifier that is merely based on pause-related attributes may achieve a lower classification accuracy compared to classifiers based on non-pause-related attributes, a combination of both pause-related and non-pause-related attributes can yield results that will outperform existing classifiers. We also trained a decision tree which is capable of classifying novice vs. skilled instances, i.e., instances from the first trials—labelled novice—and instances from the trials 14 and 15—labelled skilled—were used to train the classifier. The accuracy of the resulting decision tree was 87.24%. This classifier is visualised in Figure 12.

6 Conclusions and discussion

Users usually pause to plan or retrieve the next step to perform as mental chunks, to perceive something such as finding an item on screen or verifying the result of previous action or to plan the sub-movements of a pointing task based on the visual feedback. This research

Figure 11 Skill classifier created using pause-related attributes. Accuracy = 62.68%

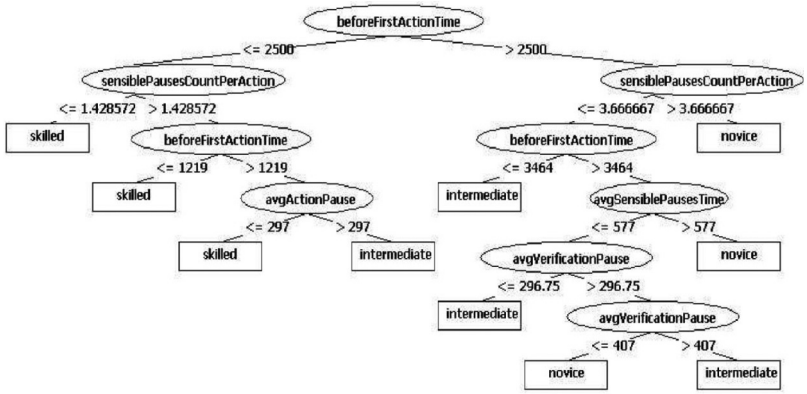


Figure 12 Two-Level (novice/skilled) Skill Classifier. Accuracy = 87.24% (see online version for colours)

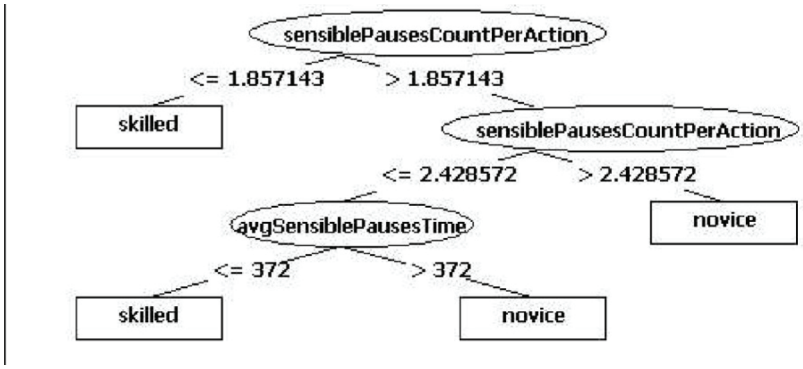


Figure 13 Average distribution of the pausesPerAct ion with different durations in trial 1. Each horizontal tick is 100 ms (see online version for colours)

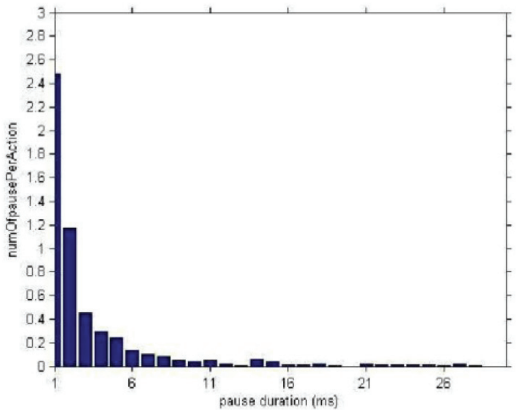


Figure 14 Average distribution of the pausesPerAction with different durations in trial 2. Each horizontal tick is 100 ms (see online version for colours)

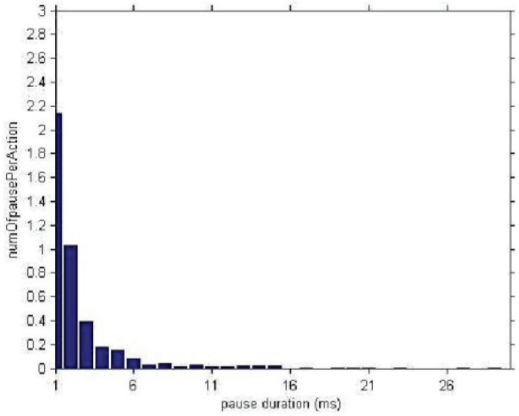
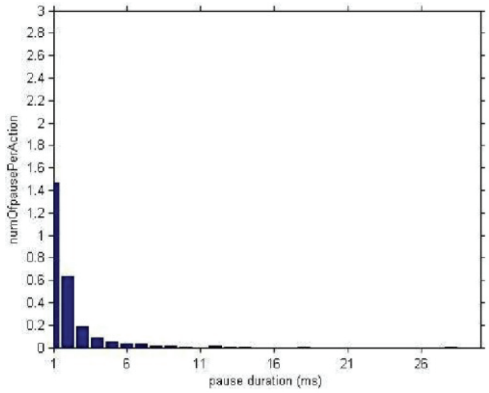


Figure 15 Average distribution of the pausesPerAct ion with different durations in trial 15. Each horizontal tick is 100 ms (see online version for colours)



investigated and measured the link between pauses in the observed behaviour of users and their skill levels. In an experiment, subjects were asked to perform a specific task using a paint software repeatedly so that their task skill—skill in performing a specific task in an application—progressed from novice to skilled. All their interactions were logged automatically. Attributes related to the pauses during interactions were extracted from these log files. The mean values of these pause-related attributes were depicted across the trials. Further, other pause-related attributes from previous works were computed and reported for comparison purposes.

Both the number and duration of pauses in the observed behaviour of users decreased as the users gain more experience and become skilled in performing a task. This decrease in the number and duration of pauses is due to the facts that experienced users

(a) organise the components of their mental process in larger chunks than that of the novices.

(b) spend little time in memory retrieval or finding/verifying something on screen

(c) skip the verification step after each action

(d) can overlap mental operators with physical operators.

Based on these pause-related attributes, skill classifiers were built using machine learning algorithms. The classifiers operate in an application and task-independent fashion. In a sense, our automatic skill detectors operate in the opposite direction in which the GOMS methods operate. GOMS models try to simulate and generate or synthesise skilled behaviour, while our proposed skill classifiers try to detect the skilled behaviour from the users' actions.

Our work is different from the related previous works in one major respect. In contrast to most previous works on skill and expertise detection, attributes related to counts of undo or cancel operations, shortcut usages, help referring counts, error rates and other similar features were not used. Instead, attributes related to pauses in users' interactions were used, which were extracted from low-level high-frequency UI events such as mouse moves and clicks. We demonstrated that the *sensiblePausesCountPerAction* attribute is a useful index of skill level. The main advantage of the pause-related attributes lies in their modelling simplicity.

Our method has limitations which make it unsuitable for some situations. We mostly used high-frequency GOMS-level measures to build the skill classifiers. GOMS techniques fail to capture users' cognitive states—such as focused, tired, etc.—and individual differences —all users are assumed to be exactly the same (Olson and Olson, 1995). Similar to GO MS techniques, our proposed model does not take into account such differences. The subject cohort we used in our experiment was composed of skilled professionals so that they could learn at a different rate compared to users with lower skill levels. Another limitation of our method is its inefficiency in detecting a user's skill level when he or she performs creative and less goal-directed tasks such as free-form and creative drawing by a paint program. The proposed method is geared towards measuring goal-directed tasks, in which the user performs a series of sub-goals and actions serially to reach a predetermined goal. For example, in most painting software, data from creative lengthy activities such as selecting colours where the user interrupts his or her drawing task and investigates the colour palette to select a colour could lead to incorrect classification. The attributes as well as the skill classifiers introduced in this paper work well for goal-directed and short tasks in which the users perform a GUI task mostly using the mouse and have less keyboard typing. We only consider pauses related to the mouse usages, i.e., the pauses during keyboard typing were not considered. To ascertain the external validity of our findings as well as the application-independency of our classifiers, further measurements and experiments are required in different applications and domains.

References

Anderson, J. (1993) '*Rules of the Mind*', Lawrence Erlbaum Associates, Hillsdale, NJ.

Anderson, J.R. (1982) 'Acquisition of cognitive skill', *Psychol. Rev.,* Vol. 89, No. 4, pp.369–406.

Anderson, J.R. (1983) '*The Architecture of Cognition',* Harvard University Press, Cambridge.

Badre, A. (1982) 'Selecting and representing information structures for visual presentation', *IEEE Transactions on Man, System and Cybernetics*, pp.495–504.

Badre, A. (1982) 'Designing chunks for sequentially displayed information', In Badre and Shneiderman, Eds., Directions in Human/Computer Interaction, Ablex.

Badre, A.N., Hudson, S.E. and Santos, P.J. (1993) 'An environment to support user interface evaluation using synchronized video and event trace recording', *Georgia Institute of Technology, GVU Center. Report* # GIT-GVU-93-16.

Barfield, W. (1986) 'Expert-Novice Differences for Software: Implications for Problem-Solving and Knowledge Acquisition', *Behav. Inf. Technol.* Vol. 5, No. 1, pp.15–29.

Benyon, D. (1993) 'Adaptive systems: a solution to usability problems', *User Model. User-Adapt. Interact*., Vol. 3, No. 1, pp.65–87.

Buxton, W.A. (1995) 'Chunking and phrasing and the design of human-computer dialogues', In Human-Computer Interaction: Toward the Year 2000, Baecker, R.M., Grudin, J., Buxton, W.A. and Greenberg, S., Eds. Morgan Kaufmann Publishers, San Francisco, CA, pp.494–499.

Card, S.K., Moran, T.P. and Newell, A. (1980a) 'The keystroke-level model for user performance time with interactive systems', *Commun. ACM.,* Vol. 23, NO. 7, pp.396–410.

Card, S.K., Moran, T.P. and Newell, A. (1980b) 'Computer text-editing: an information processing analysis of a routine cognitive skill', *Cognitive Psychology,* Vol.12, pp.32–74.

Card, S.K., Newell, A. and Moran, T.P. (1983) '*The Psychology of Human-Computer Interaction',* L. Erlbaum Associates Inc.

Chase, W. and Simon, H. (1973) 'Perception in chess', *Cognitive Psychology*, Vol. 4, pp.55–81.

Crossman, E.R.F.W. (1959) 'A theory of the acquisition of speed-skill', *Ergonomics* Vol. 2, pp.153–156.

Fischer, G. (2001) 'User modeling in human-computer interaction', *User Model. User-Adapt. Interact*., Vol.11, No.1–2, pp.65–86.

Fitts, P.M. (1964) 'Perceptual-motor skill learning', In: Melton, A.W. (Ed.), *Categories of Human Learning*, Academic Press, New York and London.

Fitts, P.M. and Posner, M.I. (1968) '*Human Performance*', Brooks/Cole Publishing Company, Belmont.

Foulin, J. (1998) 'To what extent does pause location predict pause duration in adults' and children's writing?', *Cahiers de Psychologie Cognitive*, Vol. 17, No. 3, pp.601–620.

Ghazarian, A., Noorhosseini, S.M. (2010) 'Automatic detection of users' skill levels using high-frequency user interface events', *User Modeling and User-Adapted Interaction,* Vol. 20, No. 2. pp. 109–146.

Gray, W.D, John, B.E. and Atwood, ME. (1992) 'The Precis of Project Ernestine or an Overview of a Validation of GOMS', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Hacker, W. (1994) 'Action theory and occupational psychology', Review of German empirical research since, *Ger. J. Psychol.,* Vol. 18, No. 2, pp.91–120.

Huang, M. (2003) 'Pausal behavior of end-users in online searching', *Inf. Process. Manage*. Vol. 39, No. 3 pp.425–444.

Hurst, A., Hudson, S.E. and Mankoff, J. (2007) 'Dynamic detection of novice versus skilled use without a task model', In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp.271–280. San Jose, California, USA.

Hwang, F., Keates, S., Langdon, P. and Clarkson, J. (2004) 'Mouse movements of motion-impaired users: A submovement analysis', Proceedings of ASSETS, USA, October, pp.102–109.

John, B.E. and Kieras, D.E. (1996) 'The GOMS family of user interface analysis techniques: comparison and contrast', *ACM Trans. Comput.-Hum. Interact.* Vol. 3, No. 4, pp.320–351.

John, B.E., Prevas, K., Salvucci, D.D. and Koedinger, K. (2004) 'Predictive human performance modeling made easy', In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Vienna, Austria, April 24–29. CHI '04. ACM, New York, NY, pp.455–462. DOI=http://doi.acm.org/10.1145/985692.985750.

Keates, S. and Trewin, S. (2005) 'Effect of age and Parkinson's disease on cursor positioning using a mouse'. In *Proceedings of the 7th international ACM SIGACCESS Conference on Computers and Accessibility,* Baltimore, MD, USA, October 09–12, Assets '05. ACM, New York, NY, pp.68–75.

Kieras, D. (1993) '*Using the Keystroke-Level Model to Estimate Execution Times'*. The University of Michigan, Unpublished Report, Online Version as of August 2006, http://www.pitt.edu/~cmlewis/KSM.pdf.

Kieras, D.E. (1988) Towards a practical GOMS model methodology for user interface design', In Helander, M. (Ed.), *The Handbook of Human-Computer Interaction,* pp.135–158, Amsterdam: North-Holland.

Kieras, D.E. (1996) 'A guide to GOMS model usability evaluation using NGOMSL'. In M. Helander & T. Landauer (Eds.), The handbook of human-computer interaction. (Second Edition). Amsterdam: North-Holland.

Kurosu, M., Urokohara, H., Sato, D., Nishimura, T. and Yamada, F. (2002) A new quantitative measure for usability testing: NEM (novice expert ratio method). Poster session presented at the annual conference of the usability professionals' association on humanizing design, Orlando, Florida.

Leung, S.C. and Fulcher, J. (1997) 'Classification of user expertise level by neural networks', *Intl. J. Neural Syst*., Vol. 8, NO. 2, pp.155–171.

Maxion, R.A. and deChambeau, A.L. (1995) 'Dependability at the user interface', In *Twenty-Fifth International Symposium on Fault-Tolerant Computing,* pp.528–535, Los Alamitos, CA, June. IEEE Computer Society Press.

Meyer, D., Abrams, R., Kornblum, S., Wright, C. and Smith, J. (1988) 'Optimality in human motor performance: Ideal control of rapid aimed movements', *Psychological Review*, Vol. 95, No. 3, pp.340–370.

Mitchel, T.M. (1997) '*Machine Learning*', McGrawHill, New York.

Nakamura, Y., Kato, Y. and Mitsunaga, Y. (1996) 'Proposal of a skill level index based on user's thinking time', *Electronics and Communications in Japan* (*Part III*: *Fundamental Electronic Science*). Vol. 79, No. 8, pp.47–55.

Neves, D.M. and Anderson, J.R. (1981) 'Knowledge compilation mechanisms for the automatization of cognitive skills', In: Anderson, J.R. (ed.) *Cognitive Skills and Their Acquisition*, pp.57–84. Wiley, Hillsdale.

Newell, A. and Rosenbloom, P. (1981) 'Mechanisms of skill acquisition and the law of practice', In: Anderson, J.R. Cognitive Skills and Their Acquisition, pp.2–55. Erlbaum Associates, Hillsdale.

Norman, D. (1998) *Design of Everyday Things*. Doubleday, New York.

Olson, J.R. and Olson, G.M. (1995) The growth of cognitive modeling in human-computer interaction since GOMS. In: Baecker, R.M., Grudin, J., Buxton,W.A., Greenberg, S. (Eds.) Human-Computer Interaction: Toward the Year 2000, pp.603–625. Morgan Kaufmann Publishers, San Francisco.

Quinlan, J.R. (1993) 'C4.5: *Programs for Machine Learning*', Morgan Kaufmann, Los Altos.

Rasmussen, J. (1983) 'Skills, rules and knowledge; signals, signs and symbols and other distinctions in Human performance models', *IEEE Trans. Syst. Man Cybern,*. Vol. 13, No. 3, pp.257–266.

Reeder, R.W. and Maxion, R.A. (2006) User interface defect detection by hesitation analysis. In *Proceedings of the International Conference on Dependable Systems and Networks,* June 25–28, DSN. IEEE Computer Society, Washington, DC, pp.61–72.

Reitman, J. (1976) 'Skilled perception in go: deducting memory structures from inter-response times', *Cognitive Psychology*, vol. 8, pp.336–377.

Salvucci, D.D. (2009) 'Rapid prototyping and evaluation of in-vehicle interfaces', *ACM Trans. Comput.-Hum. Interact.* Vol. 16, No. 2, pp.1–33.

Sanderson, P.M., Fisher, C. (1994) 'Exploratory sequential data analysis: foundations', *Hum. Comput. Interact. Special Issue ESDA* Vol. 9, Nos. 3–4, pp.251–317.

Santos, P.J. and Badre, A.N. (1994) 'Automatic chunk detection in human-computer interaction', In: Costabile, M.F.,Catarci, T., Levialdi, S., Santucci, G. (Eds.) *Proceedings of the Workshop on Advanced Visual Interfaces*, pp.69–77. Bari, Italy.

Sun, R. and Giles, C.L. (2001) 'Sequence learning: from recognition and prediction to sequential decision making', *UEEE Intell. Syst*., Vol. 16, No. 4, pp.67–70.

Tavalin, F. (1995) 'Context for creativity: Listening to voices, allowing a pause', *Journal of Creative Behavior*, Vol. 29, No. 2, pp.133–142.

Trumbly, J.E., Arnett, K.P. and Martin, M.P. (1993) 'Performance effect of matching computer interface characteristics and user skill level', *Int. J. Man-Mach. Stud*., Vol.38, No. 4, pp.713–724.

Witten, I.H., and Frank, E. (2005) 'Data Mining: Practical Machine Learning Tools and Techniques', 2nd Edn., Morgan Kaufmann, San Francisco.

WEBSITE

http://www.cs.umd.edu/~atif/TerpOfficeWeb/TerpOfficeV5.0/ TerpPaintVindex.html

http://sourceforge.net/projects/jspoor

http://www.cs.waikato.ac.nz/ml/weka/

http://cogtool.hcii.cs.cmu.edu/