International Journal of Human-Computer Interaction

ISSN: 1044-7318 (Print) 1532-7590 (Online) Journal homepage: http://www.tandfonline.com/loi/hihc20

Designing Inclusive Interfaces Through User Modeling and Simulation

Pradipta Biswas , Peter Robinson & Patrick Langdon

To cite this article: Pradipta Biswas , Peter Robinson & Patrick Langdon (2012) Designing Inclusive Interfaces Through User Modeling and Simulation, International Journal of Human-Computer Interaction, 28:1, 1-33, DOI: 10.1080/10447318.2011.565718

To link to this article: http://dx.doi.org/10.1080/10447318.2011.565718

Accepted author version posted online: 24 Jun 2011.

Intl. Journal of Human–Computer Interaction, 28: 1–33, 2012
Copyright © Taylor & Francis Group, LLC
ISSN: 1044-7318 print / 1532-7590 online
DOI: 10.1080/10447318.2011.565718

Taylor & Francis
Taylor & Francis Group

Designing Inclusive Interfaces Through User Modeling and Simulation

Pradipta Biswas[1], Peter Robinson[2], and Patrick Langdon[1]
[1]Department of Engineering, University of Cambridge, UK
[2]Computer Laboratory, University of Cambridge, UK

Elderly and disabled people can be hugely benefited through the advancement of modern electronic devices, as those can help them to engage more fully with the world. However, existing design practices often isolate elderly or disabled users by considering them as users with special needs. This article presents a simulator that can reflect problems faced by elderly and disabled users while they use computer, television, and similar electronic devices. The simulator embodies both the internal state of an application and the perceptual, cognitive, and motor processes of its user. It can help interface designers to understand, visualize, and measure the effect of impairment on interaction with an interface. Initially a brief survey of different user modeling techniques is presented, and then the existing models are classified into different categories. In the context of existing modeling approaches the work on user modeling is presented for people with a wide range of abilities. A few applications of the simulator, which shows the predictions are accurate enough to make design choices and point out the implication and limitations of the work, are also discussed.

> Not only do physically disabled people have experiences which are not available to the able bodied, they are in a better position to transcend cultural mythologies about the body, because they cannot do things the able-bodied feel they must do in order to be happy, "normal," and sane. If disabled people were truly heard, an explosion of knowledge of the human body and psyche would take place. — Susan Wendell (1996, p. 77)

1. INTRODUCTION

The World Health Organization (2011) states that the number of people aged 60 and older will be 1.2 billion by 2025 and 2 billion by 2050. The very old (age 80 or older) is the fastest growing population group in the developed world. Many of these elderly people have disabilities that make it difficult for them to use computers. The definition of the term "disability" differs across countries and cultures, but the World Bank (2011) estimates a rate of 10 to 12% of the population worldwide having a condition that inhibits their use of standard computer systems. The Americans with Disabilities Act in the United States and the Disability Discrimination Act in the United Kingdom prohibit any discrimination between able-bodied and disabled people with respect to education, service, and employment. There are also ethical and social reasons for designing products and services for this vast population.

However, existing design practices often isolate elderly or disabled users by considering them users with special needs and do not consider their problems during the design phase. Later they try to solve the problem by providing a few accessibility features. Considering any part of the society as "special" can never solve the accessibility problems of interactive systems. Unfortunately, existing accessibility guidelines are also not adequate to analyze the effects of impairment on interaction with devices. So designers should consider the range of abilities of users from the early design process so that any application they develop can either adapt to users with a wide range of abilities or specify the minimum capability of users it requires. For example a smartphone should either automatically adapt the screen content for different zooming levels or specify the minimum visual acuity required to read the screen.

In this article we have presented a simulation system that helps to develop inclusive systems by

- helping designers in understanding the problems faced by people with different range of abilities, knowledge, and skill;
- providing designers a tool to make interactive systems inclusive;
- assisting designers in evaluating systems with respect to people with a wide range of abilities; and

The work is funded by a Gates Cambridge Scholarship and the European Union FP7, Accessible and Assistive Information and Communication Technologies grant: "GUIDE–Gentle User Interfaces for Disabled and Elderly Citizens." We thank the editor and reviewers for their insightful comments in preparing the manuscript and our participants for taking part in our studies.

Address correspondence to Pradipta Biswas, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ United Kingdom. E-mail: pb400@cam.ac.uk

- modifying the design process of interactive system to
  - evaluate the scope of it with respect to the range of abilities of users and
  - investigate the possibilities of adaptation of the interfaces for catering users with different ranges of abilities.

The simulator can predict the likely interaction patterns when undertaking a task using a variety of input devices and estimate the time to complete the task in the presence of different disabilities and for different levels of skill. Figure 1 shows the intended use of the simulator. We aim to help evaluate existing systems and different design alternatives with respect to many types of disability. The evaluation process would be used to select a particular interface, which can then be validated by a formal user trial. The user trials also provide feedback to the models to increase their accuracy. As each alternative design does not need to be evaluated by a user trial, it will reduce the development time significantly.

The article is organized as follows. In the next section, we present a detailed survey on existing works on human behavior simulation. In the context of earlier works, we have presented the simulator in section 3. The rest part of the article describes three applications of the simulator. Each application simulates a particular task (like icon searching or menu selection) and then validates the simulation with user trials involving users with and without disabilities. Section 6 introduces the GUIDE Project, which is currently using the simulation. Finally we discuss the implications and limitations of this work in section 7 and drawn our conclusion in section 8.

2. USER MODELS

A model can be defined as a simplified representation of a system or phenomenon with any hypotheses required to describe the system or explain the phenomenon, often mathematically. The concept of modeling is widely used in different disciplines of science and engineering, ranging from models of neurons or different brain regions in neurology to construction model in architecture or model of universe in theoretical physics. Modeling human or human systems is widely used in different branches of physiology, psychology, and ergonomics. A few of these models are termed as user models when their purpose is to design better consumer products. By definition a user model is a representation of the knowledge and preferences of users (Benyon & Murray, 1993).

Research on simulating user behavior to predict machine performance was originally started during the Second World War. Researchers tried to simulate operators' performance to explore their limitations while operating different military hardware. During the same time, computational psychologists were trying to model the mind by considering it as an ensemble of processes or programs. McCulloch and Pitts' (1943) model of the neuron and subsequent models of neural networks and Marr's (1980) model of vision are two influential works in this discipline. Boden (1985) presented a detailed discussion of such computational mental models. In the late 1970s, as interactive computer systems became cheaper and accessible to more people, modeling human–computer interaction (HCI) also gained much attention. However, models like Hick's Law (Hick, 1952) or Fitts' Law (Fitt, 1954), which predict visual search time and movement time, respectively, were individually not enough to simulate a whole interaction.

The Command Language Grammar developed by Moran (1981) at Xerox PARC could be considered the first HCI model. It took a top-down approach to decompose an interaction task and gave a conceptual view of the interface before its implementation. However it completely ignored the human aspect of the interaction and did not model the capabilities and limitations of users. Card, Moran, and Newell's (1983) Model Human Processor was an important milestone in modeling HCI because it introduced the concept of simulating HCI from the perspective of users. It gave birth to the GOMS family of models (Card et al., 1983) that are still the most popular modeling tools in HCI.

There is another kind of model for simulating human behavior that not only works for HCI but also aims to establish a unified theory of cognition. These types of models originated from the earlier work of computational psychologists. Allen

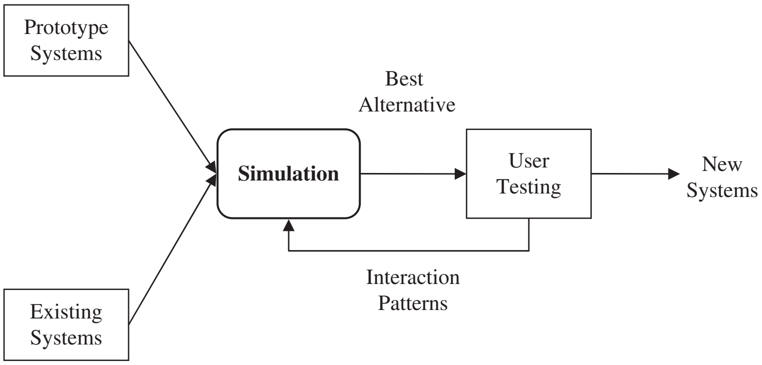


FIG. 1. Use of the simulator.

Newell (1973) pioneered the idea of unifying existing theories in cognition in his famous paper "You Can't Play 20 Questions With Nature and Win" at the 1973 Carnegie Symposium. Since then, a plethora of systems have been developed that are termed as cognitive architectures, and they simulate the results of different experiments conducted in psychological laboratories. Because these models are capable (or at least demanded to be capable) of simulating any type of user behavior, they are also often used to simulate the behavior of users while interacting with a computer. Gray, Young, and Kirschenbaum (1997) asserted that cognitive architectures ensure the development of consistent models over a range of behavioural phenomena due to their rigorous theoretical basis.

So there are two main approaches of user modeling: The GOMS family of models was developed only for HCI, whereas the models involving cognitive architectures took a more detailed view of human cognition. Based on the accuracy, detail, and completeness of these models, Kieras (2005) classified them as low-fidelity and high-fidelity models, respectively. These two types of model can be roughly mapped to two different types of knowledge representation. The GOMS family of models is based on goal-action pairs and corresponds to the Sequence/Method representation, whereas cognitive architectures aim to represent the users' mental model (Carroll & Olson, 1990). The Sequence/Method representation assumes that all interactions consist of a sequence of operations or generalized methods, whereas the mental model representation assumes that users have an underlying model of the whole system.

There is a third kind of model in HCI that evaluates an interface by predicting users' expectations, rather than their performance (e.g., Task Action Language [Reisner, 1981], Task Action Grammar [Payne & Green, 1986], etc.). These models represent an interaction by using formal grammar where each action is modeled by a sentence. They can be used to compare users' performance based on standard sentence complexity measures; however, they have not yet been used and tested extensively for simulating users' behavior (Carroll & Olson, 1990).

In the following sections, we briefly describe these different types of user model, followed by a critical review of existing models, and set out the objectives of this research.

2.1. The GOMS Family of Models

GOMS stands for Goals, Operators, Method and Selection. It was inspired by the GPS system developed by Newell (Newell & Simon, 1995). It assumes that people interact with a computer to achieve a goal by selecting a method, which consists of a sequence of basic operations. The GOMS model enables a designer to simulate the sequence of actions of a user while undertaking a task by decomposing the task into goals and subgoals (John & Kieras, 1996). There are many variations of the original GOMS model.

The keystroke level (KLM) model (Card et al., 1983) simplifies the GOMS model by eliminating the goals, methods, and selection rules, leaving only six primitive operators:

1. Pressing a key
2. Moving the pointing device to a specific location.
3. Making pointer drag movements.
4. Performing mental preparation.
5. Moving hands to appropriate locations.
6. Waiting for the computer to execute a command.

The durations of these six operations have been empirically determined. The task completion time is predicted by the number of times each type of operation must occur to accomplish the task.

Kieras developed a structured language representation of GOMS model, called Natural GOMS Language (NGOMSL; Kieras, 1994). Originally, it was an attempt to represent the content of a cognitive complexity theory (CCT) model (Johnson, 1992) at a higher level of notation. CCT is a rule-based system developed by Bovaria, Kieras, and Polson (1990) to model the knowledge of users of an interactive computer system. In NGOMSL, the methods of the original GOMS model are represented in terms of production rules of the CCT model. Kieras, Wood, Abotel, and Hornof (1995) also developed a modeling tool, GLEAN (GOMS Language Evaluation and Analysis), to execute NGOMSL. It simulates the interaction between a simulated user with a simulated device for undertaking a task.

John and Kieras (1996) proposed a new version of the GOMS model, called CPM-GOMS, to explore the parallelism in users' actions. This model decomposes a task into an activity network (instead of a serial stream) of basic operations (as defined by KLM) and predicts the task completion time based on the Critical Path Method.

2.2. Cognitive Architectures

Allen Newell (1990) developed the State Operator And Result (SOAR) architecture as a possible candidate for his unified theories of cognition. According to Newell (1990) and Johnson-Laird (1988), the vast variety of human response functions for different stimuli in the environment can be explained by a symbolic system. So the SOAR system models human cognition as a rule-based system and any task is carried out by a search in a problem space. The heart of the SOAR system is its chunking mechanism. Chunking is "a way of converting goal-based problem solving into accessible long-term memory (productions)" (Newell, 1990). It operates in the following way. During a problem-solving task, whenever the system cannot determine a single operator for achieving a task and thus cannot move to a new state, an impasse is said to occur. An impasse models a situation where a user does not have sufficient knowledge to carry out a task. At this stage SOAR explores all possible operators and selects the one that brings it nearest to the goal. It then learns a rule that can solve a similar situation in future. Laird, Rosenbloom, and Newell (1994) successfully explained the power law of practice through the chunking mechanism.

However, there are certain aspects of human cognition (such as perception, recognition, motor action) that can better be explained by a connectionist approach than a symbolic one (Oka, 1991). It is believed that initially conscious processes control our responses to any situation and after sufficient practice, automatic processes are in charge for the same set of responses (Hampson & Morris, 1996). Lallement and Alexandre (1997) classified all cognitive processes into synthetic or analytical processes. Synthetic operations are concerned with low-level, nondecomposable, unconscious, perceptual tasks. In contrast, analytical operations signify high-level, conscious, decomposable, reasoning tasks. From the modeling point of view, synthetic operations can be mapped on to connectionist models, whereas analytic operations correspond to symbolic models. Considering these facts, the Adaptive Control of Thought–Rational (ACT–R) system (Anderson & Lebiere, 1998) does not follow the pure symbolic modeling strategy of the SOAR; rather, it was developed as a hybrid model, which has both symbolic and subsymbolic levels of processing. At the symbolic level, ACT–R operates as a rule-based system. It divides the long-term memory into declarative and procedural memory. Declarative memory is used to store facts in the form of "chunks" and the procedural memory stores production rules. The system works to achieve a goal by firing appropriate productions from the production memory and retrieving relevant facts from the declarative memory. However, the variability of human behavior is modeled at the subsymbolic level. The long-term memory is implemented as a semantic network. Calculation of the retrieval time of a fact and conflict resolution among rules is done based on the activation values of the nodes and links of the semantic network.

The Executive-Process/Interactive Control (Kieras & Meyer, 1990), or EPIC, architecture pioneers to incorporate separate perception and motor behavior modules in a cognitive architecture. It mainly concentrates on modeling the capability of simultaneous multiple task performance of users. It also inspired the ACT–R architecture to install separate perception and motor modules and developing the ACT–R/PM system. A few examples of their usage in HCI are the modeling of menu searching and icon searching tasks (Byrne, 2001; Hornof & Kieras 1997).

The Constraint-based Optimizing Reasoning Engine (Eng et al., 2006; Howes, Vera, Lewis, & McCurdy, 2004; Tollinger et al., 2005), or CORE, system takes a different approach to model cognition. Instead of a rule-based system, it models cognition as a set of constraints and an objective function. Constraints are specified in terms of the relationship between events in the environment, tasks, and psychological processes. Unlike the other systems, it does not execute a task hierarchy; rather, prediction is obtained by solving a constraint satisfaction problem. The objective function of the problem can be tuned to simulate the flexibility in human behavior.

There exist additional cognitive architectures (such as Interactive Cognitive Subsystems [Barnard, 2011], Apex, DUAL, CLARION ["Cognitive Architecture," n.d.], etc.), but they are not yet as extensively used as the previously discussed systems.

2.3. Grammar-Based Models

The grammar-based model (such as task action grammar [Payne & Green, 1986] and task action language [Reisner, 1981]) simulates an interaction in the form of grammatical rules. As for example, task action language models

- Operations by Terminal symbols
- Interaction by a Set of rules
- Knowledge by Sentences

This type of modeling is quite useful to compare different interaction techniques. However, they are more relevant to model knowledge and competence of a user than performance.

2.4. Application-Specific Models

A lot of work has been done on user modeling for developing customizable applications. These models have the following generic structure (Figure 2). They maintain a user profile and use different types of Artificial Intelligence systems to predict performance. The user profile section stores detail about user relevant for a particular application and inference machine use this information to personalize the system. A plethora of examples of such models can be found in the *User Modeling and User-Adapted Interaction* journal and the proceedings of the User Modeling, Adaptation and Personalization conference. This type of models is particularly popular in online recommender or help systems. A few representative applications of such models are as follows.

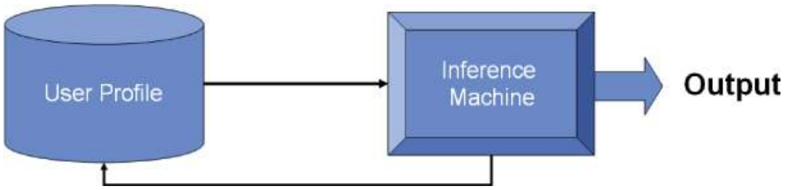


FIG. 2. Application-specific user models (color figure available online).

The generative user model (Motomura, Yoshida, & Fujimoto, 2000) was developed for personalized information retrieval. In this model, input query words are related to user's mental state and retrieved object using latent probabilistic variables. Norcio (1989) used fuzzy logic to classify users of an intelligent tutoring system. The fuzzy groups are used to derive certain characteristic of the user and thus deriving new rules for each class of user. Norcio and Chen (1992) also used an artificial neural network for the same purpose as in their previous work (Norcio, 1989). In their model, users' characteristics are stored as an image, and neural networks are used to find patterns in users' knowledge, goals, and so on. Yoshida and Motoda (1996) similarly developed the Clipboard system to automate complex task execution using a single command based on previously executed commands.

The Lumiere Convenience Project (Horovitz, Breese, Heckerman, Hovel, & Rommelse, 1998) used influence diagram in modeling users. The Lumiere project is the background theory of the Office Assistant shipped with the Microsoft Office application. The influence diagram models the relationships among users' needs, goals, user background, and so on. However, all these models are developed by keeping only a single application in mind and so they are hardly usable to model human performance in general.

2.5. Review

The GOMS family of models is mainly suitable for modeling the optimal behavior (skilled behavior) of users (John & Kieras, 1996). These models assume that for each instance of a task execution, the goal and the plan of a user are determined before the execution is started. During execution of a task, a novice first time user or a knowledgeable intermittent user may not have a fixed plan beforehand and can even change goals (or subgoals) during execution of the task. Even expert users do not follow a fixed sequence of actions every time. So the assumptions of the GOMS model may not hold true for many real-life interactions. In actuality, these models do not have probabilistic components beyond the feature of selecting the execution time of primitive operators from a statistical distribution in order to model the uncertainty involved in the suboptimal behaviour of users. As it fails to model the suboptimal behavior, it cannot be used to predict the occurrences of different errors during interaction. These problems are common for any Sequence/Method representations as these ways of representations overlook the underlying mental models of users (Caroll & Olson, 1990).

On the other hand, cognitive architectures model the uncertainty of human behavior in detail but they are not easily accessible to nonpsychologists, and this causes problem as interface designers are rarely psychologist as well. For example, the ACT–R architecture models the content of a long-term memory in the form of a semantic network, but it is very difficult for an interface designer to develop a semantic network of the related concepts of a moderately complex interface. Developing a sequence of production rules for SOAR or a set of constraints for CORE is equally difficult. The problem in usability issues of cognitive architectures is also supported by the development of the X-PRT system (Tollinger et al., 2005) for the CORE architecture. In addition, Kieras (2005) has shown that a high-fidelity model cannot always outperform a low-fidelity one, though it is expected to do so.

Researchers have already attempted to combine the GOMS family of models and cognitive architectures to develop more usable and accurate models. Salvucci and Lee (2003) developed the ACT–Simple model by translating basic GOMS operations (such as move hand, move mouse, press key) into ACT–R production rules. However they do not model the "think" operator in detail, which corresponds to the thinking action of users and differentiates novices from experts. The model works well in predicting expert performance but does not work for novices.

Blandford, Butterworth, and Curzon (2004) implemented the Programmable User Model (PUM; Young, Green, & Simon, 1989) by using the SOAR architecture. They developed a program, the SOAR Translation from Instruction Language made Easy (or STILE), to convert the PUM Instruction Language into SOAR production rules. However, this approach also demands good knowledge of SOAR on the part of an interface designer. Later, the PUM team identified additional problems with runable user models, and they are now investigating abstract mathematical models (Butterworth & Blandford, 1997).

The CogTool system (http://cogtool.hcii.cs.cmu.edu/) combines GOMS models and ACT–R system for providing quantitative prediction on interaction. The system simulates expert performance through GOMS modeling, whereas the ACT–R system helps to simulate exploratory behavior of novice users (John, Prevas, Salvucci, & Koedinger, 2004). The system also provides graphical user interfaces to quickly prototype interfaces and to evaluate different design alternatives based on quantitative prediction (John, 2010). However, it does not yet seem to be used for users with disability or assistive interaction techniques.

There also exist some application-specific models that combine GOMS models with a cognitive architecture. For example, Gray and Sabnani (Gray & Sabnani, 1994) combined GOMS with ACT–R to model a VCR programming task, whereas Peck and John (1992) used SOAR to model interaction with a help-browser, which ultimately turned out to be a GOMS model.

Another problem of existing modeling approaches stems from issues related to disability. Researchers have concentrated on designing assistive interfaces for many different applications, including

- Web browsers
- Augmentative and alternative communication aids
- New interaction techniques
- Scanning interfaces
- Gravity wells
- Novel hardware interfaces

- Head-mounted switches
- Eye gaze trackers
- Brain–computer interfaces

Most of these works concentrate on a particular application or a set of users, which reduces the scalability of the overall approach. Furthermore, developing systems for a small segment of market often makes the system very costly.

There is not much reported work on systematic modelling of assistive interfaces. McMillan (1992) felt the need to use HCI models to unify different research streams in assistive technology, but his work aimed to model the system rather than the user. The AVANTI project (Stephanidis & Constantinou, 2003; Stephanidis et al., 1998) modeled an assistive interface for a web browser based on static and dynamic characteristics of users. The interface is initialized according to static characteristics (such as age, expertise, type of disability, etc.) of the user. During interaction, the interface records users' interaction and adapts itself based on dynamic characteristics (such as idle time, error rate, etc.) of the user. This model works based on a rule-based system and does not address the basic perceptual, cognitive, and motor behavior of users, so it is hard to generalize to other applications.

The EASE tool (Mankoff, Fait, & Juang, 2005) simulates effects of interaction for a few visual and mobility impairments. However the model is demonstrated for a sample application of using word prediction software but not yet validated for basic pointing or visual search tasks performed by people with disabilities.

Keates, Clarkson, and Robinson (2000) measured the difference between able-bodied and motor-impaired users with respect to the Model Human Processor (Card et al., 1983), and motor impaired users were found to have a greater motor action time than their able-bodied counterparts. The finding is obviously important, but the KLM model itself is too primitive to model complex interaction and especially the performance of novice users.

Serna, Pigot, and Rialle (2007) used ACT–R cognitive architecture (Anderson & Libiere, 1998) to model progress of dementia in Alzheimer's patients. They simulated the loss of memory and increase in error for a representative task at kitchen by changing different ACT–R parameters (Anderson & Libiere, 1998). The technique is interesting, but their model still needs rigorous validation through other tasks and user communities.

Our previous user model (Biswas, Bhattacharyya, & Samanta, 2005) also took a more generalized approach than the AVANTI project. It broke down the task of user modeling into several steps that included clustering users based on their physical and cognitive ability, customizing interfaces based on user characteristics, and logging user interactions to update the model itself. However, the objective of this model was to design adaptable interfaces and not to simulate users' performance.

Gajos, Wobbrock, and Weld (2007) developed a model to predict pointing time of users with mobility impairment and adapt interfaces based on the prediction. They estimated the movement time by selecting a set of features from a pool of seven functions of movement amplitude and target width, and then using the selected features in a linear regression model. This model shows interesting characteristics of movement patterns among different users but fails to develop a single model for all. Movement patterns of different users are found to be inclined to different functions of distance and width of targets.

3. THE SIMULATOR

Based on the previous discussion, Figure 3 plots the existing general purpose HCI models in a space defined by the skill and physical ability of users. To cover most of the blank spaces in the diagram, we need models that can

- Simulate HCI of both able-bodied and disabled users.
- Work for users with different levels of skill.
- Be easy to use and comprehend for an interface designer.

To address the limitations of existing user modeling systems, we have developed the simulator (Biswas, 2010) as shown in Figure 4. It consists of the following three components:

The Application model represents the task currently undertaken by the user by breaking it up into a set of simple atomic tasks following KLM model (Card et al., 1983).

The Interface model decides the type of input and output devices to be used by a particular user and sets parameters for an interface.

The User model simulates the interaction patterns of users for undertaking a task analyzed by the task model under the configuration set by the interface model. It uses the sequence of phases defined by Model Human Processor (Card et al., 1983).

- The perception model simulates the visual perception of interface objects. It is based on the theories of visual attention.

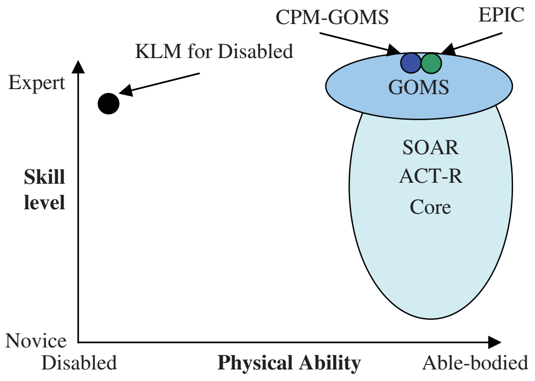


FIG. 3. Existing human–computer interaction models with respect to skill and physical ability of users (color figure available online).

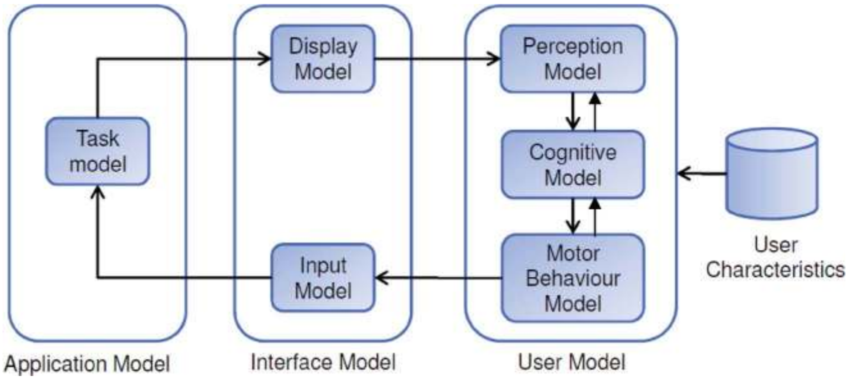


FIG. 4. Architecture of the simulator (color figure available online).

- The cognitive model determines an action to accomplish the current task. It is more detailed than the GOMS model (John & Kieras, 1996) but not as complex as other cognitive architectures.
- The motor behavior model predicts the completion time and possible interaction patterns for performing that action. It is based on statistical analysis of screen navigation paths of disabled users.

The details about users are store in xml format in the user profile following the ontology shown in Figure 5. The ontology stores demographic detail of users like age and sex and divide the functional abilities in perception, cognition, and motor action. The perception, cognitive, and motor behavior models takes input from the respective functional abilities of users.

The perception model (Biswas & Robinson, 2009a) simulates the phenomenon of visual perception (like focusing and shifting attention). We have investigated eye gaze patterns (using a Tobii X120 eye tracker) of people with and without visual impairment. The model uses a backpropagation neural network to predict eye gaze fixation points and can also simulate the effects of different visual impairments (like Maccular Degeneration, color blindness, Diabetic Retinopathy, etc.) using image processing algorithms. Figure 6 shows the actual and predicted eye movement paths (green line for actual, black line for predicted) and points of eye gaze fixations (overlapping green circles) during a visual search task. The figure shows the prediction for a protanope (a type of color blindness) participant, and so the right-hand figure is different from the left hand one as the effect of protanopia was simulated on the input image.

The cognitive model (Biswas & Robinson, 2008) breaks up a high-level task specification into a set of atomic tasks to be performed on the application in question. The operation of it is illustrated in Figure 7. At any stage, users have a fixed policy based on the current task in hand. The policy produces an action, which in turn is converted into a device operation (e.g., clicking on a button, selecting a menu item, etc.). After application of the operation, the device moves to a new state. Users have to map this state to one of the states in the user space. Then they again decide a new action until the goal state is achieved.

Besides performance simulation, the model also has the ability to learn new techniques for interactions. Learning can occur either offline or online. The offline learning takes place when the user of the model (such as an interface designer) adds new

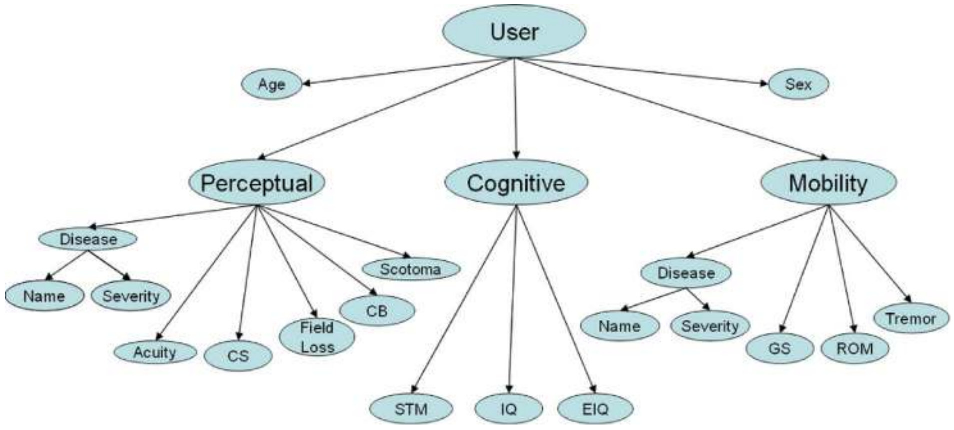


FIG. 5. User ontology. *Note.* STM = short-term memory; EIQ = emotional intelligent quotient; CS = Contrast Sensitivity; CB = Colour Blindness; GS = Grip Strength, ROM = Range of Motion (color figure available online).

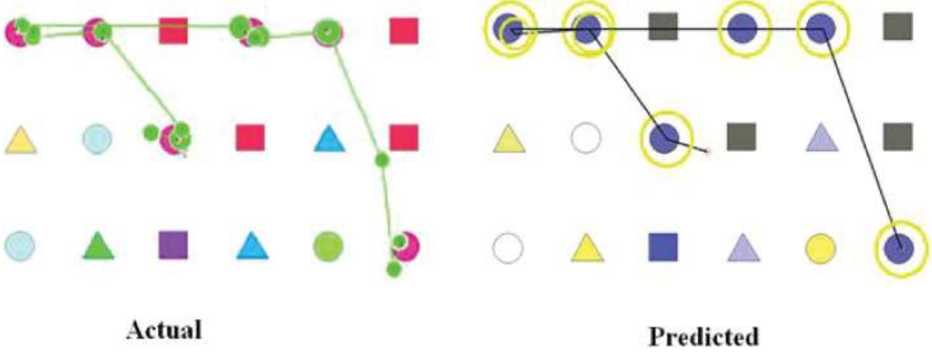


FIG. 6. Eye movement trajectory for a user with color blindness (color figure available online).

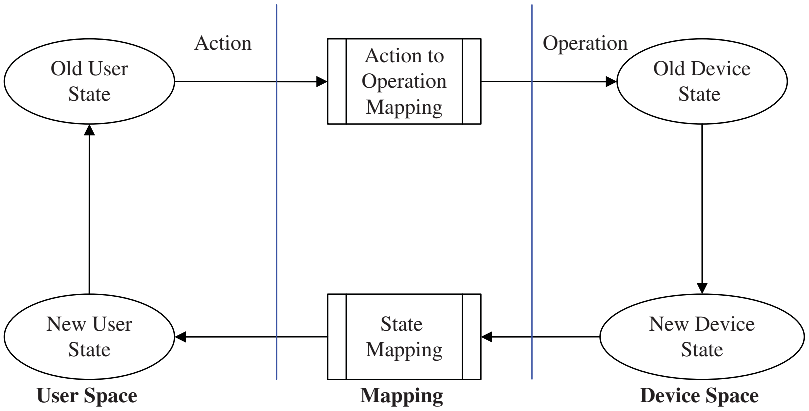


FIG. 7. Sequence of events in an interaction (color figure available online).

states or operations to the user space. The model can also learn new states and operations itself. During execution, whenever the model cannot map the intended action of the user into an operation permissible by the device, it tries to learn a new operation. To do so, it first asks for instructions from outside. The interface designer is provided with the information about previous, current, and future states, and he can choose an operation on behalf of the model. If the model does not get any external instructions, then it searches the state transition matrix of the device space and selects an operation according to the label matching principle (Rieman & Young, 1996). If the label matching principle cannot return a prospective operation, it randomly selects an operation that can change the device state in a favorable way. It then adds this new operation to the user space and updates the state transition matrix of the user space accordingly. In the same way, the model can also learn a new device state. Whenever it arrives in a device state unknown to the user space, it adds this new state to the user space. It then selects or learns an operation that can bring the device into a state desirable to the user. If it cannot reach a desirable state, it simply selects or learns an operation that can bring the device into a state known to the user.

The model can also simulate the practice effect of users. Initially the mapping between the user space and the device space remains uncertain. It means that the probabilities for each pair of state/action in the user space and state/operation in the device space are less than 1. After each successful completion of a task the model increases the probabilities of those mappings that lead to the successful completion of the task, and after sufficient practice the probability values of certain mappings reach 1. At this stage the user can map his space unambiguously to the device space and thus behave optimally.

The motor behavior model (Biswas & Robinson, 2009b) is developed by statistical analysis of cursor traces from motor-impaired users. We have evaluated hand strength (using a Baseline 7-pc Hand Evaluation Kit) of able-bodied and motor-impaired people and investigated how hand strength affects HCI. Based on the analysis, we have developed a regression model to predict pointing time. Figure 8 shows an example of the output from the model. The thin purple line shows a sample trajectory of mouse movement of a motor-impaired user. It can be seen that the trajectory contains random movements near the source and the target. The thick red and black lines encircle the contour of these random movements. The area under the contour has a high probability of missed clicks as the movement is random there and thus lacks control.

These models do not need detailed knowledge of psychology or programming to operate. They have graphical user interfaces

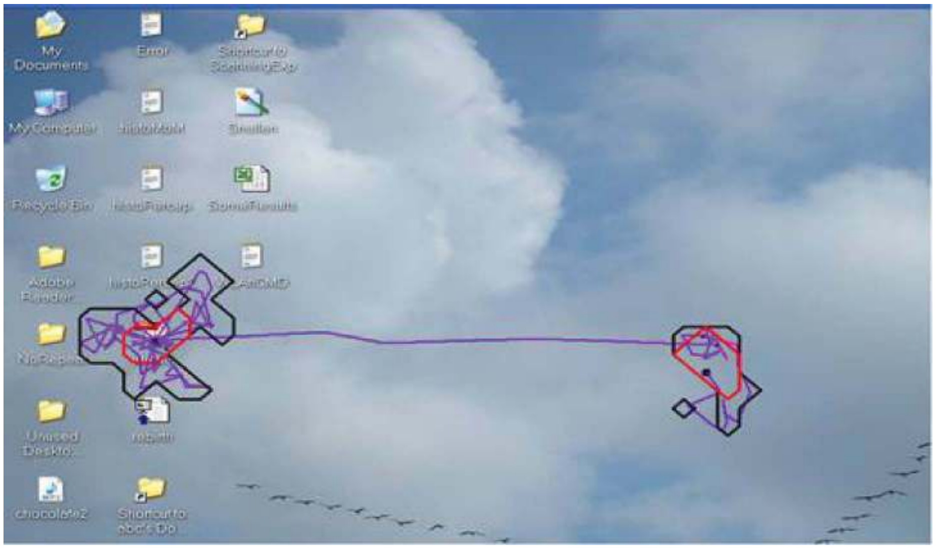


FIG. 8. Mouse movement trajectory for a user with cerebral palsy (color figure available online).

to provide input parameters and showing output of simulation. Figure 9 shows a few interfaces of the simulator.

At present it supports a few types of visual and mobility impairments. For both visual and mobility impairment, we have developed the user interfaces in three different levels:

- In the first level (Figure 9a) the system simulates different diseases.
- In the next level (Figure 9b) the system simulates the effect of change in different visual functions (like visual acuity, contrast sensitivity, visual field loss, etc.) and hand strength metrics (like grip strength, range of motion of forearm, wrist, etc.).
- In the third level (Figure 9c), the system allows different image-processing algorithms to be run (such as high pass filtering, blurring, etc.) on input images and to set demographic detail of users.

The simulator can show the effects of a particular disease on visual functions and hand strength metrics and in turn their effect on interaction. For example, it can demonstrate how the progress of dry macular degeneration increases the number and sizes of scotoma (dark spots in eyes) and converts a slight peripheral visual field loss into total central vision loss. Similarly it can show the perception of an elderly color-blind user, or in other words the combined effect of visual acuity loss and color blindness. We have modeled the effects of age and gender on hand strength, and the system can show the effects of cerebral palsy or Parkinson's disease for different age group and gender. A few sample screenshots can be found at http://www.cl.cam.ac.uk/~pb400/Demo.htm.

3.1. Validation of the Models

Each of the perception, cognitive, and motor behavior models were calibrated and validated separately involving people with and without visual and mobility impairment (Biswas, 2010).

The perception model was validated through an eye gaze tracking study for a visual search task. We compared the correlation between actual and predicted visual search time, eye gaze, and investigated the error in prediction. The actual and predicted visual search time correlated statistically significantly with less than 40% error rate for more than half of the trials (Biswas & Robinson, 2009a).

The cognitive model was used to simulate interaction for first-time users, and it can simulate the effect of learning as well (Biswas & Robinson, 2008).

The motor behavior model was validated through ISO 9241 pointing task. The actual and predicted movement time correlated statistically significantly with less than 40% error rate for more than half of the trials (Biswas & Robinson, 2009b).

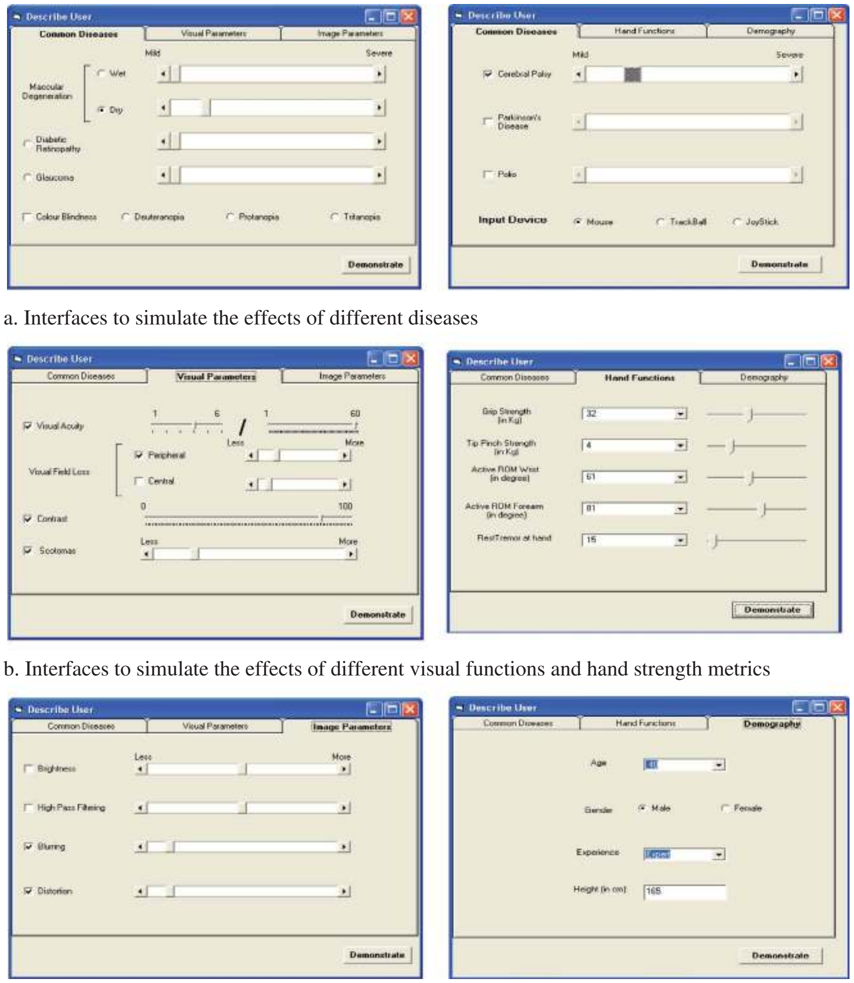
3.2. Working Principle

The simulator works in the following three steps.

1. While a task is undertaken by participants, a monitor program records the interaction. This monitor program records

    a. A list of key presses and mouse clicks (operations),
    b. A sequence of bitmap images of the interfaces (low-level snapshot), and
    c. Locations of windows, icons, buttons and other controls in the screen (high-level snapshot).

2. Initially, the cognitive model analyzes the task and produces a list of atomic tasks (detailed task specification).



a. Interfaces to simulate the effects of different diseases

Grip Strength (in Kg) 32
Tip Pinch Strength (in Kg) 4
Active ROM Wrist (in degree) 61
Active ROM Forearm (in degree) 81
Rest Tremor at hand 15

b. Interfaces to simulate the effects of different visual functions and hand strength metrics

Gender ● Male ○ Female
Height (in cm) 165

c. Interfaces to run image processing algorithms and set demographic detail of users

FIG. 9. A few interfaces of a prototype of the toolbox (color figure available online).

3. If an atomic task involves perception, the perception model operates on the event list and the sequence of bitmap images. Similarly, if an atomic task involves movement, the motor behavior model operates on the event list and the high-level snapshot.

In the remaining sections of this article, we demonstrate the use of the simulator through an icon-searching and menu selection task. In the first application, the simulation accurately predicts performance of users with visual and mobility impairment. In the second case, the simulator is used to identify the accessibility problems of menus and thus redesign a menu selection interface.

4. CASE STUDY 1—ICON SEARCHING TASK

In graphical user interfaces, searching and pointing constitute a significant portion of HCI. Users search for many different artifacts like information in a web page, button with a particular caption in an application, e-mail from a list of mails, and so on. We can broadly classify searching in two categories.

Text searching includes any search that only involves searching for text and not any other visual artifact. Examples include menu searching, keyword searching in a document, mailbox searching, and so on.

Icon searching includes searching for a visual artifact (such as an icon or a button) along with text search for its caption. The search is mainly guided by the visual artifact, and the text is generally used to confirm the target.

In this section, we present a study involving an icon searching task. We simulated the task using the simulator and evaluated the predictive power of the model by comparing actual task completion time with prediction in terms of correlation and percentage error in prediction.

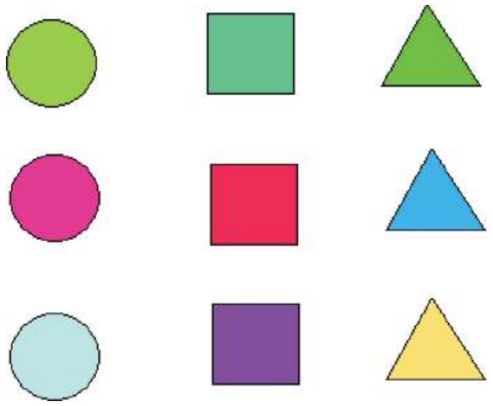


FIG. 10. Corpus of shapes (color figure available online).

4.1. Process

We conducted trials with two families of icons. The first consisted of geometric shapes with colors spanning a wide range of hues and luminance (Figure 10). The second consisted of images from the system folder in Microsoft Windows to increase the external validity (Figure 10) of the experiment. Each icon bears a caption underneath (Figure 11). The first two letters and length of all the captions were kept nearly same to avoid any pop-out effect of the captions during visual search.

Previous work found that alignment and grouping of screen elements have most influence on subjective preference of users, which was also correlated to the search time (Parush, Nadir, & Shtub, 1998). So this experiment was a mixed design with two measures and a between-subject factor. The within-subject measures were spacing between icons and font size of captions (see Figure 12). We used the following three levels for each measure:

- Spacing between icons
  - Sparse: 180 pixels horizontally, 230 pixels vertically. This was the maximum separation possible in the screen.
  - Medium: 150 pixels horizontally, 200 pixels vertically.

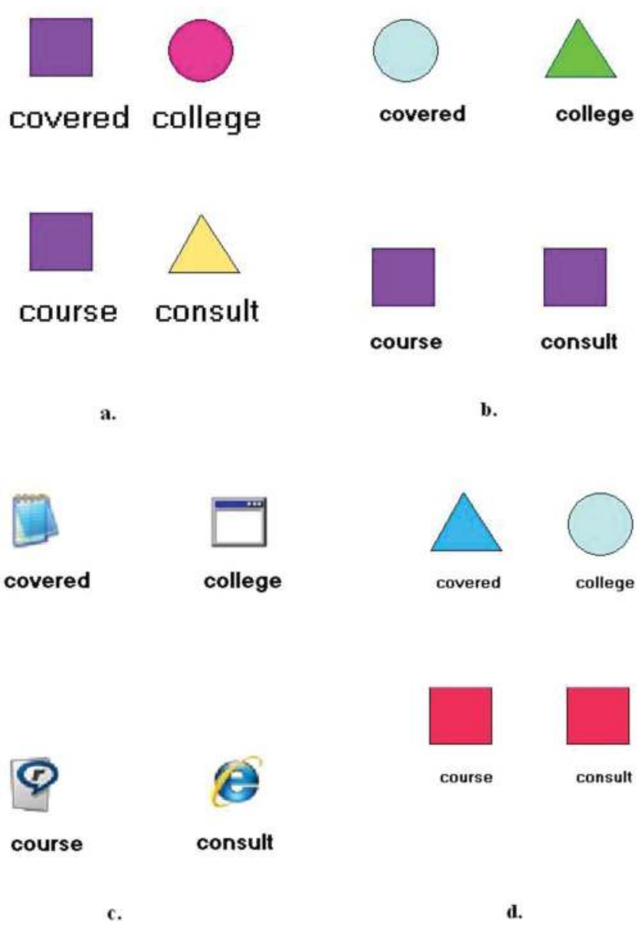


FIG. 12. Sample screenshot of the study. (a) Dense *Spacing* Big *Font*, (b) Medium *Spacing* Medium *Font*, (c) Sparse *Spacing* Medium *Font*, (d) Dense *Spacing* Small *Font*) (color figure available online).

  - Dense: 120 pixels horizontally, 170 pixels vertically. This was the minimum possible separation without overlapping the icons.
- Font size
  - Small: 10 point.

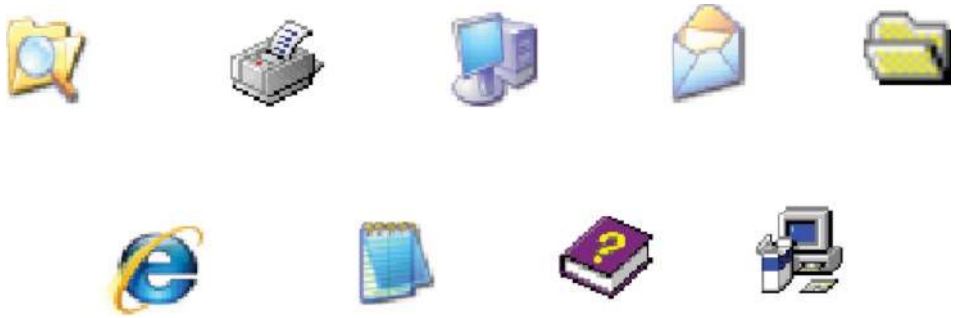


FIG. 11. Corpus of icons (color figure available online).

  - Medium: 14 point (as suggested by the Royal National Institute of Blind People; http://www.rnib.org.uk/Pages/Home.aspx).
  - Large: 20 point.

The between-subjects factor is

- Group
  - Able bodied
  - Visually impaired
  - Motor impaired

The experimental task consisted of shape searching and icon searching tasks. The task was as follows:

- A particular target (shape or icon with a caption) was shown.
- A set of 18 candidates for matching was shown.
- Participants were asked to click on the candidate, which was same as the target both in terms of icon and caption.

The sequence of the trials was randomized using a Latin square. Each participant undertook eight trials for each combination of the within-subject measures. Each participant performed 72 searching and pointing tasks in total. They were trained for the task before start of the actual trial. However one of the participants (P4) retired after undertaking 40 trials.

4.2. Material

We used a 1280 × 800 LCD color display driven by a 1.7 GHz Pentium 4 PC running the Microsoft Windows XP operating system. We also used a standard computer Mouse (Microsoft IntelliMouse® Optical Mouse) for clicking on the target.

4.3. Participants

We collected data from two able-bodied, two visually impaired, and three motor-impaired participants (Table 1). All were expert computer users and used computers more than once a week.

4.4. Simulation

Initially we analyzed the task in light of the cognitive model. Because the users undertook preliminary training, we considered them as expert users. We followed the GOMS analysis technique and identified two subtasks:

- Searching for the target.
- Pointing and clicking on the target.

The prediction is obtained by sequentially running the perception model and the motor behavior model. The predicted task completion time is the summation of the visual search time (output by the perception model) and the pointing time (output by the motor behavior model).

TABLE 1
List of participants

	Age	Gender	Impairment
C1	27	M	Able-bodied
C2	30	M	
P1	27	M	Myopia (–4.5 / –4.5 Dioptre)
P2	26	M	Myopia (–5.5 / –5.5 Dioptre)
P3	30	M	Hypokinetic motor impairment resulted from cerebral palsy, restricted hand movement, wheelchair user
P4	42	M	Cerebral palsy, restricted hand movement, also suffering tremor in hand, wheelchair user
P5	45	M	Hyperkinetic motor impairment resulted from stroke, significant tremor in fingers, wheelchair user

4.5. Results

Figure 13 shows the correlation between actual and predicted task completion times. We also calculated the relative error $\frac{\mathit{Predicted}-\mathit{Actual}}{\mathit{Actual}}$ and show its distribution in Figure 14. The superimposed curve shows a normal distribution with same mean and standard deviation as the relative error. We found that the correlation is $\rho = 0.7$ ($p < .001$) and 56% of the trials have a relative error within ± 40%. The average relative error is +16% with a standard deviation of 54%. The model did not work for 10% of the trials, and the relative error is more than 100% in those cases. For the remaining 90% of the trials, the average relative error is +6% with a standard deviation of 42%.

We also analyzed the effects of font size and icon spacing on the task completion time and investigated whether the prediction reflects these effects as well. So we conducted two 3 × 3 analyses of variance (Spacing × Font × Group) on the actual and predicted task completion times, respectively. We investigated both the within-subject effects and results of a multivariate test. In the analyses of variance, we did not consider the trials for which the relative error was more than 100%, as the model did not work for those trials. Participant P4 did not also complete the trial, leaving us with 40 rows of data ($N = 40$).

For calculating the within-subject effects, the Greenhouse-Geisser correction was used if the Mauchy's test detected violation from sphericity assumption (Field, 2009) giving fractional values for the degrees of freedom. In this study, the main effect of *Spacing* did not violate sphericity assumption ($W = .854$, $\chi^2 = 5.69$ in actual, $W = .99$, $\chi^2 = 0.37$ in prediction, $p > .05$), whereas the main effect of Font ($W = .825$, $\chi^2 = 6.93$ in actual, $W = .836$, $\chi^2 = 6.43$ in prediction, $p < .05$) and the

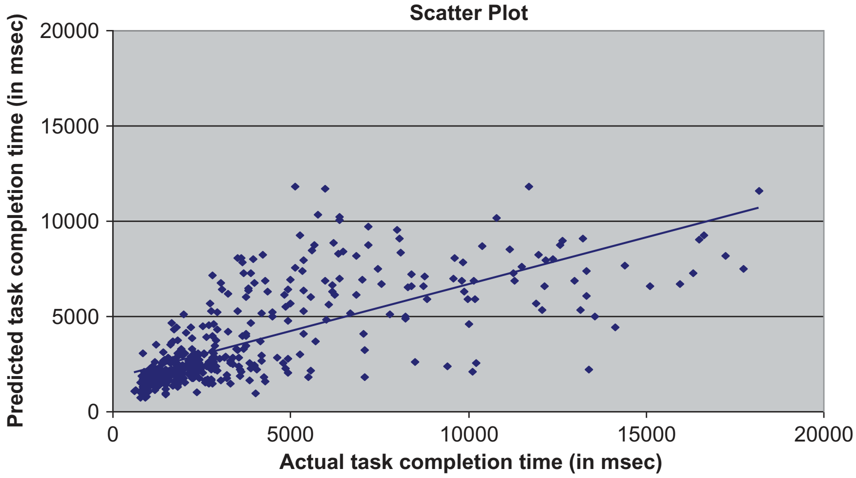


FIG. 13. Scatterplot between actual and predicted task completion time (color figure available online).

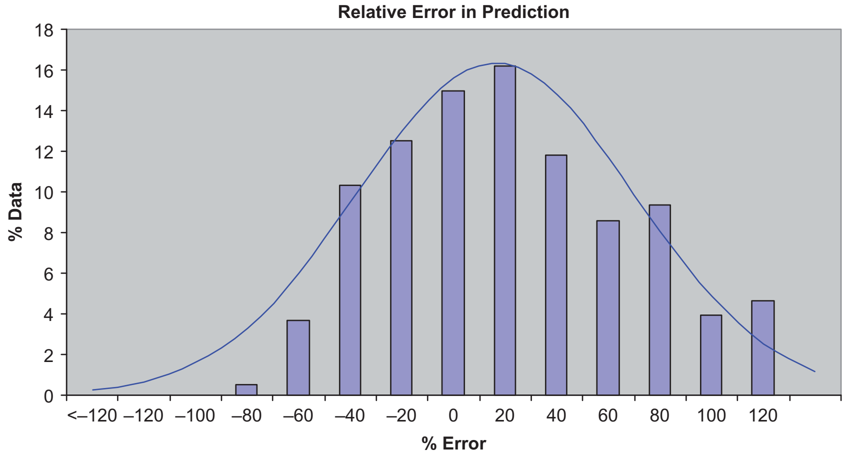


FIG. 14. Relative error in prediction (color figure available online).

interaction effect of Spacing and Font ($W = .244$, $\chi^2 = 49.94$ in actual, $W = .539$, $\chi^2 = 21.91$ in prediction, $p < .05$) violated sphericity assumption. Tables 2 and 3 show results of the within-subjects tests and multivariate tests on the actual and predicted task completion times, respectively. The tables list the degrees of freedom, F value, and corresponding significance for different measures. Table 2 shows that three sources have significant effects on both actual and predicted task completion time. They are

- A main effect of spacing, $F(2, 74) = 5.44$, $p < .05$, on actual task completion time.
- A main effect of spacing, $F(2, 74) = 6.95$, $p < .05$, in predicted task completion time.
- An interaction effect of spacing and group, $F(4, 74) = 3.15$, $p < .05$, on actual task completion time.
- An interaction effect of spacing and group, $F(4, 74) = 4.64$, $p < .05$, on predicted task completion time.
- An interaction effect of font and group, $F(3.4, 62.97) = 5.02$, $p < .05$, on actual task completion time.
- An interaction effect of font and group, $F(3.44, 63.6) = 3.75$, $p < .05$, on predicted task completion time.

The main effect of font and interaction effects between font and group and spacing, font and spacing do not have significant effects on both actual and predicted task completion times. We confirmed these effects through a multivariate test (Table 3), which is not affected by the sphericity assumption. Table 3 shows the following effects:

- A main effect of spacing (Wilks's $\lambda = 0.762$), $F(2, 36) = 5.62$, $p < .05$, on actual task completion time.
- A main effect of spacing (Wilks's $\lambda = 0.741$), $F(2, 36) = 6.28$, $p < .05$, in predicted task completion time.
- A main effect of font (Wilks's $\lambda = 0.817$), $F(2, 36) = 4.05$, $p < .05$, in predicted task completion time.

TABLE 2
Test of within-subjects effects on task completion time

	Actual			Predicted		
Source	*df*	*F*	Sig.	*df*	*F*	Sig.
Spacing	**2.0**	**5.44**	**.006**	**2.0**	**6.95**	**.002**
Spacing × Group	**4.0**	**3.15**	**.019**	**4.0**	**4.64**	**.002**
Error(Spacing)	74.0			74.0		
Font	1.7	0.22	.770	1.7	2.89	.071
Font × Group	**3.4**	**5.02**	**.002**	**3.4**	**3.75**	**.012**
Error(Font)	63.0			63.6		
Spacing × Font	2.3	1.03	.370	3.3	1.54	.204
Spacing × Font × Group	4.7	0.83	.528	6.5	1.32	.250
Error(Spacing × Font)	86.3			121.0		

Note: Significant values are in boldface

TABLE 3
Multivariate test on completion time

	Actual			Predicted		
Effect	*df*	*F*	Sig.	*df*	*F*	Sig.
Spacing	**2**	**5.62**	**.008**	**2**	**6.28**	**.005**
Spacing × Group	**4**	**2.78**	**.033**	**4**	**3.97**	**.006**
Font	2	0.31	.739	**2**	**4.05**	**.026**
Font × Group	**4**	**6.39**	**0**	**4**	**5.05**	**.001**
Spacing × Font	4	1.41	.253	4	2.18	.093
Spacing × Font × Group	8	2.15	.043	8	1.74	.106

Note: Significant values are in boldface

- An interaction effect of spacing and group (Wilks's $\lambda = 0.750$), $F(4, 72) = 2.78$, $p < .05$, on actual task completion time.
- An interaction effect of spacing and group (Wilks's $\lambda = 0.671$), $F(4, 72) = 3.97$, $p < .05$, on predicted task completion time.
- An interaction effect of font and group (Wilks's $\lambda = 0.545$), $F(4, 72) = 6.39$, $p < .05$, on actual task completion time.
- An interaction effect of font and group (Wilks's $\lambda = 0.610$, $F(4, 72) = 5.05$, $p < .05$, on predicted task completion time.

It can be seen from Tables 2 and 3 that the prediction captures all effects at 99.95% confidence level in both within-subject test and multivariate test. Figures 15 and 16 show that the effect sizes (η^2) are also fairly similar in the prediction as in the actual. The maximum difference is below 10% in within-subject test and below 20% in multivariate test. This suggests that the simulator successfully explained the variance in task completion time for different factors. As these factors include both interface parameters and physical characteristics of users, we can infer that the simulator has successfully explained the effects of different interface layouts on task completion time for people with visual and motor impairment.

Figures 17 and 18 show the effects of font size and spacing for different user groups. In Figures 17 and 18, the points depict the average task completion time and the bars show the standard error at a 95% confidence level. It can be seen from Figures 17 and 18 that the prediction is in line with the actual task completion times for different font sizes and icon spacing.

However the prediction is less accurate in one of the nine conditions - the medium font size and medium spacing for the motor impaired users (see Figures 19, 20). We found that, in these cases, the model underestimates the task completion times and fails to capture the variability in it. We have further analyzed the effects of spacing and font size for each user group separately (Table 4).

It can be seen from Table 4 that in terms of significance at $p < .05$, the prediction deviates from the actual in the following two cases (highlighted in bold):

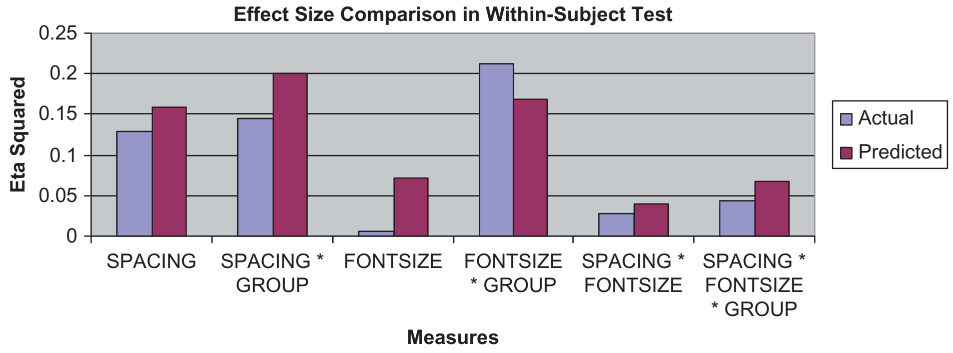


FIG. 15. Effect size comparison in analysis of variance (color figure available online).

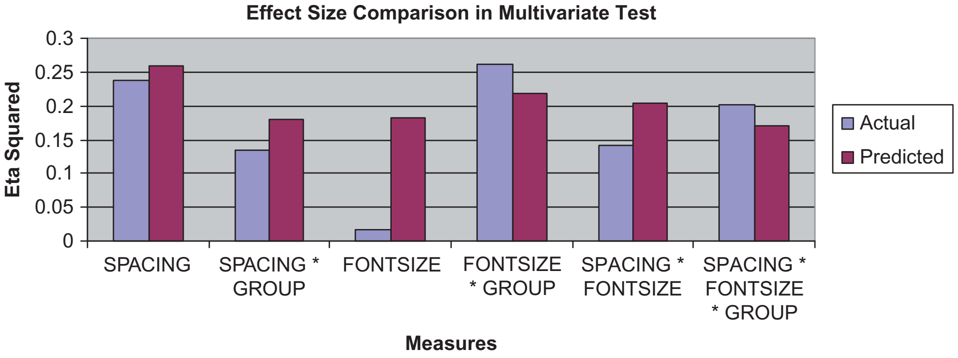


FIG. 16. Effect size comparison in multivariate analysis of variance (color figure available online).

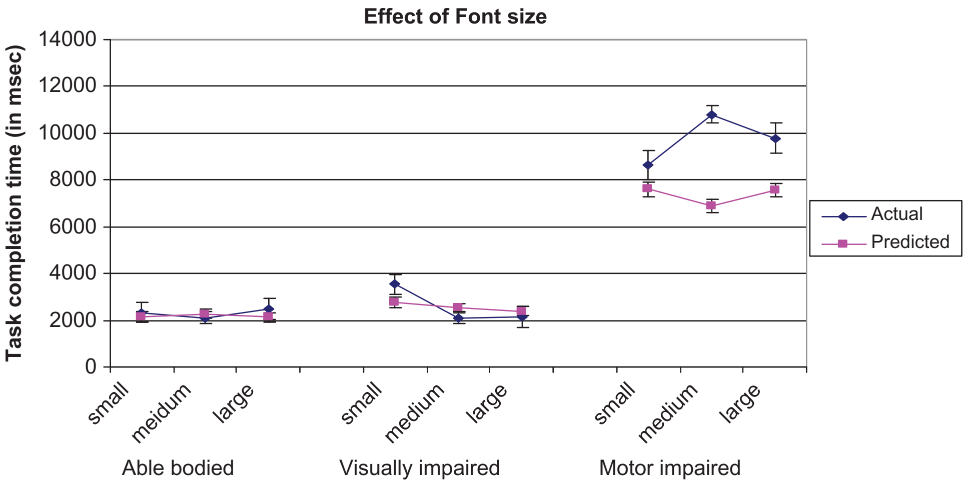


FIG. 17. Effect of *font size* in different user groups (color figure available online).

- Interaction effect of spacing and font for able-bodied users: $F(4, 60) = 1.78$, $p > .05$ for actual, $F(4, 60) = 2.69$, $p < .05$, for prediction.
- Effect of spacing for motor-impaired users: $F(2, 14) = 2.93$, $p > .05$, for actual, $F(2, 14) = 3.78$, $p < .05$, for prediction.

Finally we compared the mean and standard deviation of the actual and predicted task completion times for each condition. Table 5 lists the relative difference $\frac{Predicted-Actual}{Actual}$ in mean and standard deviations between actual and predicted task completion time.

It can be seen from Table 5 that only in four conditions (highlighted in bold) is the average predicted time different from the

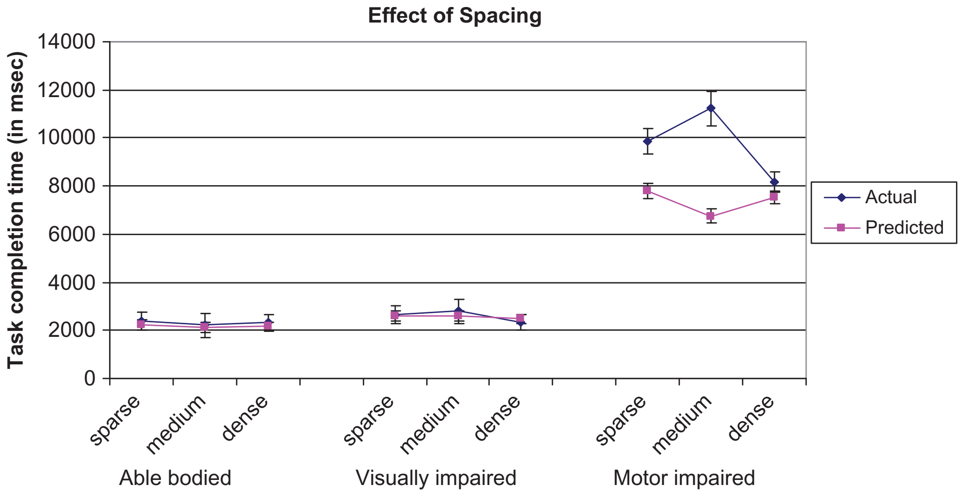


FIG. 18. Effect of *spacing* in different user groups (color figure available online).

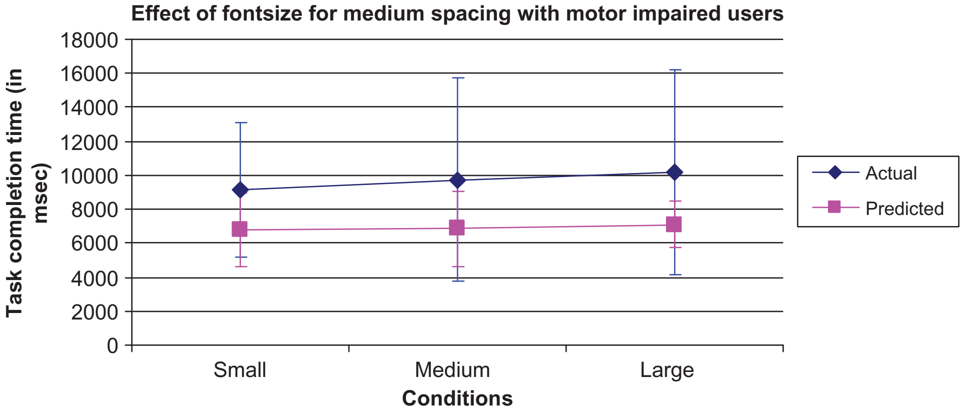


FIG. 19. Effect of medium *spacing* in motor-impaired users (color figure available online).

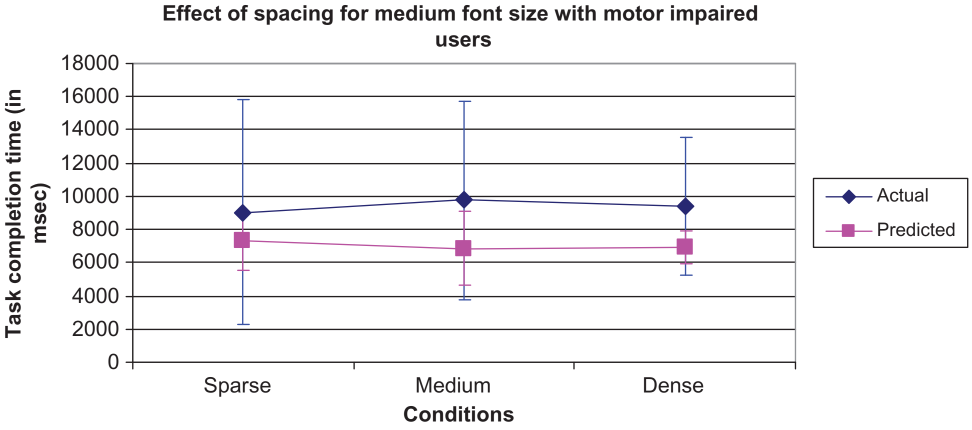


FIG. 20. Effect of medium *font size* in motor-impaired users (color figure available online).

actual predicted time by more than ±40%. However the standard deviation is predicted quite less than in actual in many occasions. The difference is less severe for visually impaired users than the other two groups. One possible reason for the difference may be the effects of learning and fatigue as able-bodied users might work quickly due to learning effect and motor impaired users might feel fatigue. So, we have analyzed the effects of usage time through a regression model.

TABLE 4
Analysis of variance for each user group

	Actual				Predicted			
Source	*df*	*F*	Sig.	η^2	*df*	*F*	Sig.	η^2
Able bodied								
Spacing	2	0.21	.815	0.014	2	0.38	.688	0.025
Error(*Spacing*)	30				30			
Font	2	0.72	.495	0.046	2	2.73	.081	0.154
Error(*Font*)	30				30			
Spacing × *Font*	4	1.78	.144	0.106	**4**	**2.69**	**.039**	**0.152**
Error(*Spacing* × *Font*)	60				60			
Visually impaired								
Spacing	1.4	0.52	.54	0.034	2	0.81	453	0.051
Error(*Spacing*)	21.3				30			
Font	1.4	8.39	.004	0.359	2	5.72	.008	0.276
Error(*Font*)	21.5				30			
Spacing × *Font*	1.5	2.90	.089	0.162	4	0.21	.933	0.014
Error(*Spacing* × *Font*)	22.3				60			
Motor impaired								
Spacing	2	2.93	.087	0.295	**2**	**3.78**	**.049**	**0.350**
Error(*Spacing*)	14				14			
Font	2	1.53	.251	0.179	2	1.56	.245	0.182
Error(*Font*)	14				14			
Spacing × *Font*	4	0.26	.904	0.035	4	0.67	.62	0.087
Error(*Spacing* × *Font*)	28				28			

Note: Significant values are in boldface

Analyzing effect of usage time. We have considered the predicted task completion time and the usage time as independent variables and the actual task completion time as the dependent variable. The usage time for each trial measures the total time spent (in seconds) from beginning of the session to the end of the trial. Table 6 shows the regression coefficients.

It seems that usage time can significantly ($p < .005$) affect the actual time though the improvement in ΔR^2 is only 2%. The inclusion of usage time in the regression model also reduces the change in R^2 from .39 to .01, which means it increases the genralizability of the model (Biswas & Robinson, 2009b). The positive value of coefficient B indicates that the task completion time was directly proportional to the usage time. Figure 21 shows a weak positive correlation ($\rho = 0.42$) between usage time and task completion time. Perhaps it means that users felt fatigue or bored as the session went on and took more time to complete the task in later trials.

4.6. Discussion

Choosing a particular interface from a set of alternatives is a significant task for both design and evaluation. In this study, we considered a representative task and the results showed that the effects of both factors (separation between icons and font size) were the same in the prediction as for actual trials with different user groups. The prediction from the simulator can be reliably used to capture the main effects of different design alternatives for people with a wide range of abilities.

However, the model did not work accurately for about 30% of the trials where the relative error is more than 50%. These trials also accounted for an increase in the average relative error from zero to 16%. In particular, the predicted variance in task completion times for motor impaired users was smaller than the actual variance. This can be attributed to many factors; the most important ones are as follows.

Effect of usage time. Fatigue and learning effects: The trial continued for about 15 to 20 minutes. A few participants (especially one user in the motor impaired group) felt fatigue. On the other hand, some users worked more quickly as the trial proceeded. The model did not consider these effects of fatigue and learning. It seems from the analysis that the usage time can significantly affect the total task completion time. In the future we would like to analyze the effect of usage time in more detail and plan to incorporate it into the input parameters of the model.

User characteristics. The variance in the task completion time can be attributed to various factors such as expertise, usage time, type of motor impairment (hypokinetic vs. hyperkinetic),

TABLE 5
Relative differences in mean and standard deviation

			% Difference in	
	Spacing	Font	*M*	*SD*
Able bodied	Sparse	Small	−7.58	−55.65
		Medium	21.25	−27.62
		Large	−21.29	−83.04
	Medium	Small	−11.16	−66.72
		Medium	12.80	−45.72
		Large	−9.61	−24.4
	Dense	Small	−7.93	−76.85
		Medium	−4.94	−44.12
		Large	−7.34	−1.27
Visually impaired	Sparse	Small	−26.97	−69.06
		Medium	12.70	−1.19
		Large	32.51	17.21
	Medium	Small	**−41.33**	−78.7
		Medium	**58.14**	21.87
		Large	14.93	−20.11
	Dense	Small	28.62	7.59
		Medium	0.01	−19.48
		Large	−5.84	−13.97
Motor impaired	Sparse	Small	6.19	−48.68
		Medium	−35.31	−76.38
		Large	−24.97	−80.72
	Medium	Small	**−40.10**	−67.8
		Medium	**−43.22**	−71.32
		Large	−36.46	−83.7
	Dense	Small	9.52	−61.08
		Medium	−29.82	−74.52
		Large	4.23	7.57

Note: errors higher than 40% are in boldface

TABLE 6
Effect of usage time

Model		*B*	SE	Beta
1	(Constant)	502.56	212.43	
	Predicted Time	0.67	0.04	0.62*
2	(Constant)	372.16	214.53	
	Predicted Time	0.59	0.05	0.55*
	Usage Time	2.77	0.91	0.14*

Note. $\Delta R^2 = 0.62^*$ for Model 1, $\Delta R^2 = 0.64^*$ for Model 2 ($^*p < .005$).

interest of the participant, and so on. Currently, the model characterizes the extent of motor impairment of the user only by measuring the grip strength (Bernard, Mills, Peterson, & Storrer, 2001); in the future more input parameters may be considered.

5. CASE STUDY 2—MENU SELECTION TASK

In this study, we have investigated the accessibility of program selection menus for a digital TV interface. Previous work on menu selection investigated selection time of different menu items based on their position (Nilsen, 1992) and menu-searching strategies (Hornof & Kieras, 1997) for able-bodied users. Researchers worked on menu interaction for cell phones (Mardsen & Jones, 2001; Ruiz & Lank, 2010)), but there is not much reported work on accessibility issues of menus, in particular for digital TV interfaces. Existing approaches like target expansion (McGuffin & Balakrishnan, 2005) or target identification (Hurst, Hudson, & Mankoff, 2010) are not very suitable for menu selection as menu items are more densely spaced than other types of targets like buttons or icons in a screen. Ruiz's approach (Ruiz & Lank, 2010) of expanding target region has also not been found to reduce menu selection time significantly. There is also not much reported work on the legibility issues of menu captions. Most researchers do not find difference in

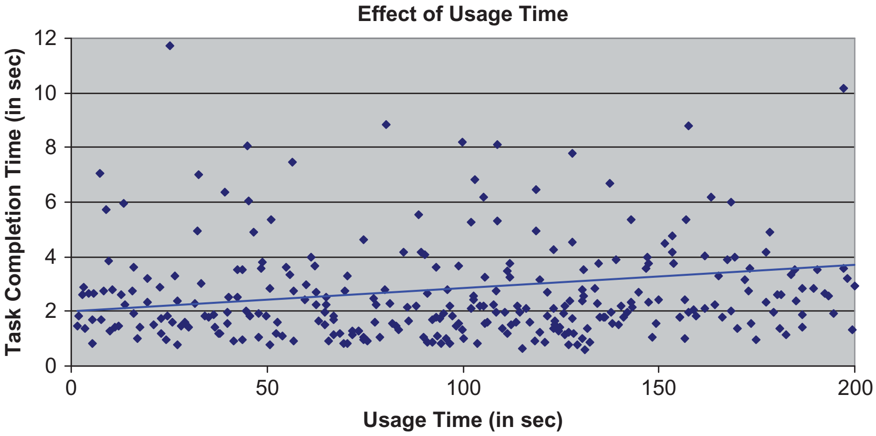


FIG. 21. Effect of usage time (color figure available online).

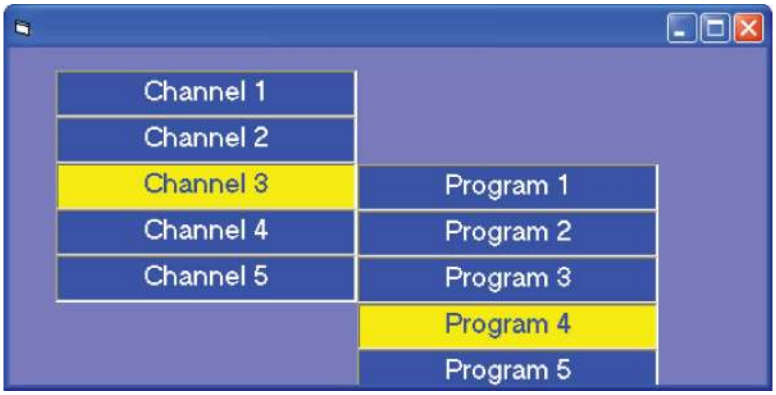


FIG. 22. Interface used in the study (color figure available online).

terms of reading time due to font types with respect to online reading tasks (Bernard, Liao, & Mills, 2001; Beymer, Russell, & Orton, 2007; Boyarski, Neuwirth, Forlizzi, & Regli, 1998). Although Bernard and colleagues (Bernard, Liao, et al., 2001; Bernard, Mills, et al., 2001) reported significant difference in reading times between Tahoma and Corsiva fonts for a reading task of two pages, but the difference may turn insignificant during reading short captions. However, there is significant difference in reading time and legibility due to font size. Beymer and colleagues (2007) preferred the 12-point size, whereas the Royal National Institute of Blind People and Bernard, Liao, et al. (2009a) preferred the 14-point size.

We take help from the simulator in identifying the accessibility problems of program selection menu with respect to visually and mobility impaired users. Based on the results of the simulation we have designed new interfaces. The study consists of the following three stages:

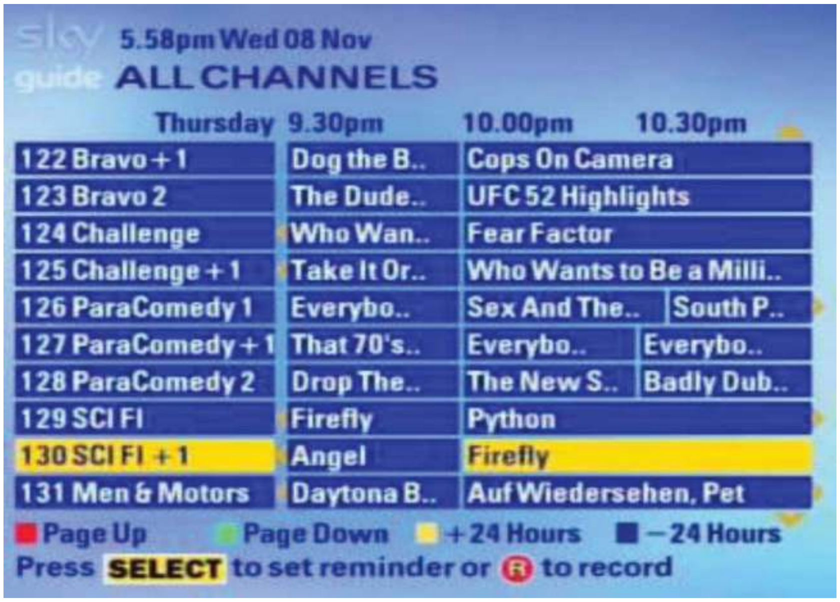
- Problem identification through simulation
- New interface evaluation through simulation
- Validation of the simulation through a controlled experiment

5.1. The Study

Initially we designed the following interface (Figure 22), which looks similar to existing systems (Figure 23). In this particular work, we investigated

- Sensory problems of
  - People with less visual acuity
  - People having color blindness
- Interaction problems of
  - People with motor impairment using a pointing device

In this particular study, the simulator takes a sample task of selecting a menu item and the screenshot of the interface as input and shows the perception of visually impaired users and cursor trajectory of motor-impaired users as output. In the simulation study, we did not bother with the particular words used as captions because the simulation results are not to be used by participants. We used captions like Channel 1, Program 1, or Time 1 as captions. However in the validation study we used



5.58pm Wed 08 Nov

ALL CHANNELS

Thursday	9.30pm	10.00pm	10.30pm
122 Bravo + 1	Dog the B..	Cops On Camera	
123 Bravo 2	The Dude..	UFC 52 Highlights	
124 Challenge	Who Wan..	Fear Factor	
125 Challenge + 1	Take It Or..	Who Wants to Be a Milli..	
126 ParaComedy 1	Everybo..	Sex And The..	South P..
127 ParaComedy + 1	That 70's..	Everybo..	Everybo..
128 ParaComedy 2	Drop The..	The New S..	Badly Dub..
129 SCI FI	Firefly	Python	
130 SCI FI + 1	Angel	Firefly	
131 Men & Motors	Daytona B..	Auf Wiedersehen, Pet	

🟥 Page Up 🟩 Page Down 🟨 +24 Hours ■ −24 Hours

Press SELECT to set reminder or Ⓡ to record

FIG. 23. Representative of an actual interface (color figure available online).

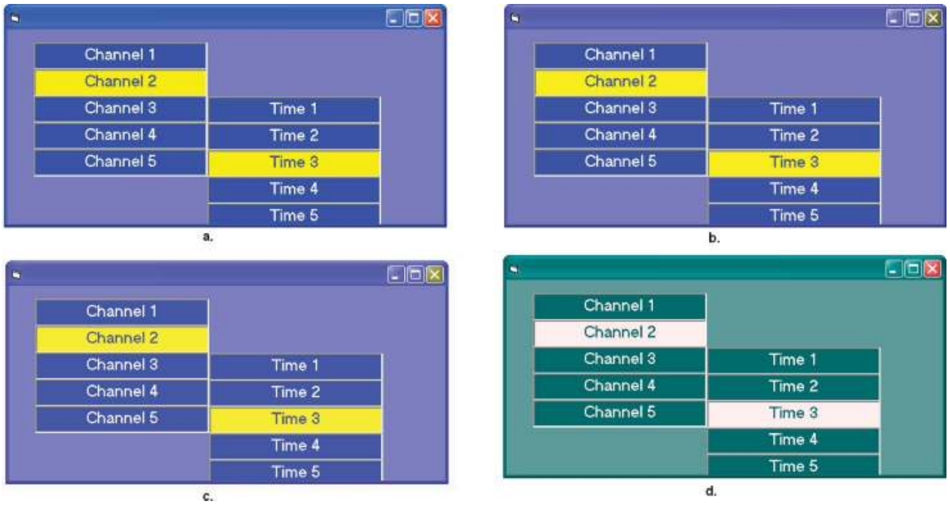


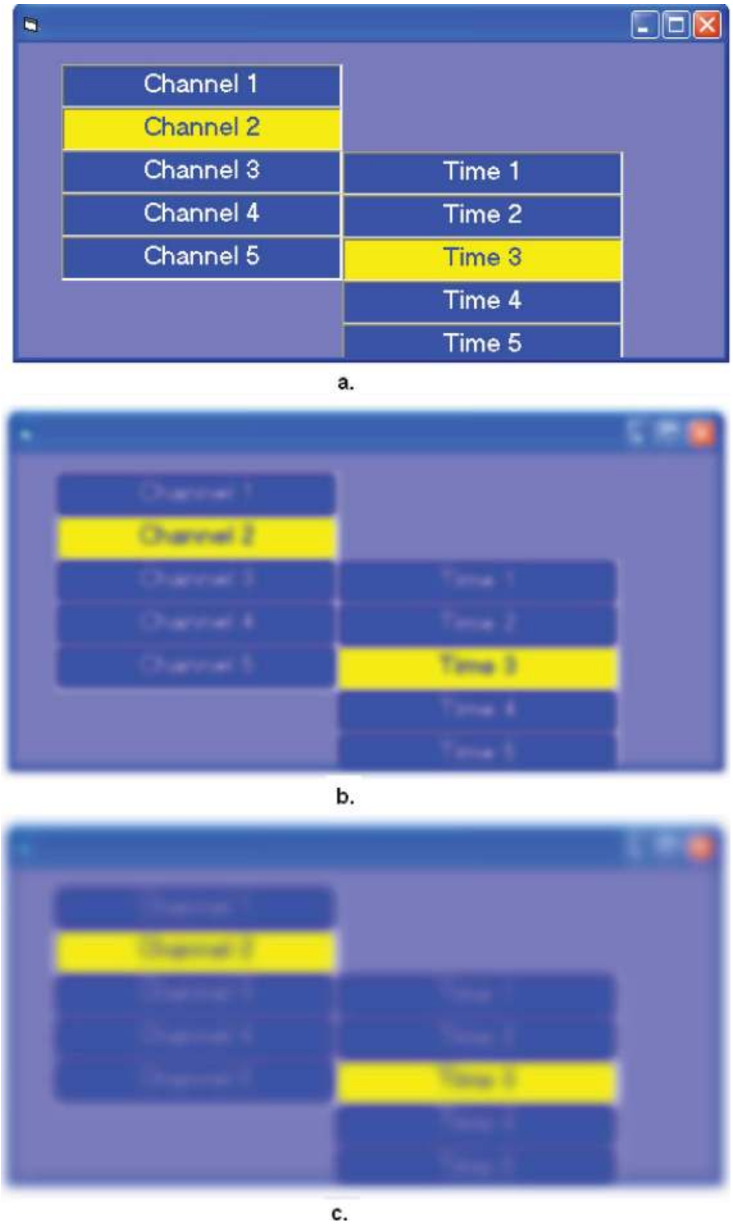
FIG. 24. Perception of people having color blindness. (a) Original interface. (b) Interface perceived by Potanopia. (c) Interface perceived by Deuteranopia. (d) Interface perceived by Tritanopia (color figure available online).

different words as captions and discussed it in detail in a later section.

Problem identification. Initially the output from the simulator is used to identify accessibility problems. Figure 24 shows the perception of the interface for three different types of color blindness. Details about the simulation can be found in Biswas (2010). It can be seen that although the colors look different, the particular color combination of the interface does not reduce the legibility.

Figure 25 shows the perception of the interface of people with mild (less than approximately –3.5 Dioptre) and severe (more than approximately –4 Dioptre) acuity loss. Details about the simulation can be found in Biswas (2010). It can be seen from the figure that the captions (which are in 14 point) become illegible for severe acuity loss.

Figure 26 shows a possible cursor trace of a pointing device (like mouse) operated by a person having motor impairment. The thin purple line shows a sample trajectory of mouse movement of a motor-impaired user. It can be seen that the trajectory contains random movements near the source and the target. The thick black lines encircle the contour of these random movements. The area under the contour has a high probability of missed clicks as the movement is random there and thus lacks control. Details about the simulation can be found in Biswas (2010). It can be seen that as the buttons



a.

b.

c.

FIG. 25. Perception of people having less visual acuity. (a) Original interface. (b) Interface perceived by mild visual acuity loss. (c) Interface perceived by severe visual acuity loss (color figure available online).

are closely spaced, there is a significant probability of missed click in a wrong button, which would surely frustrate any user.

Based on the simulation results we identified the following two accessibility issues

- Legibility of captions
- Spacing between menu items

New interface. Based on the previous discussion, we have redesigned the interfaces. We have increased the font size of captions for users with visual impairment. For people with motor impairment, we have changed the size of the buttons without changing the screen size such that no couple of buttons shares a common boundary. This should reduce chances of missed clicks. Figures 27 and 28 show the new interfaces. We have not designed anything new to cater to color-blind users, as the present interface seems perfect for them.

Figure 29 shows the perception of the new interface for people with mild and severe acuity loss. It can be seen that the modified captions (now at 18 point) has better legibility than the previous case even for severe acuity loss. We have

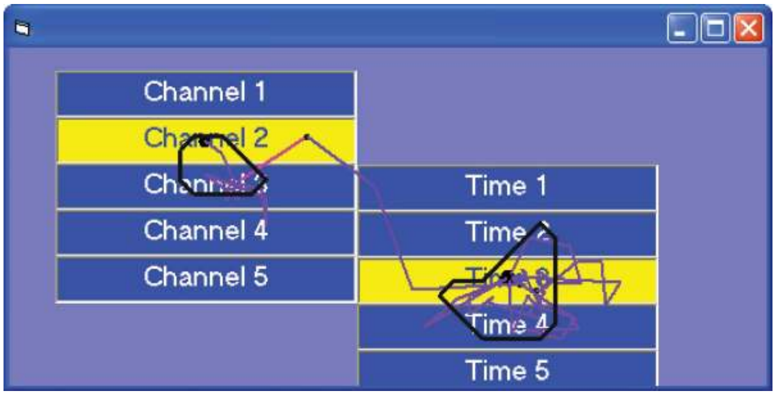


FIG. 26. Possible cursor trace of a pointing device (like mouse) operated by a person having motor impairment (color figure available online).

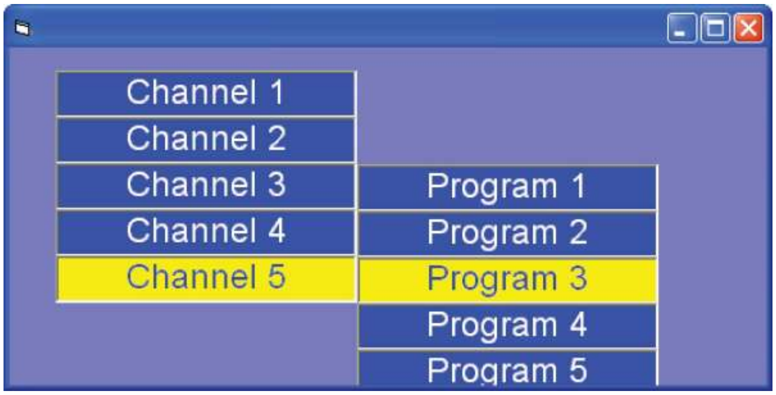


FIG. 27. Interface for people having less visual acuity (color figure available online).

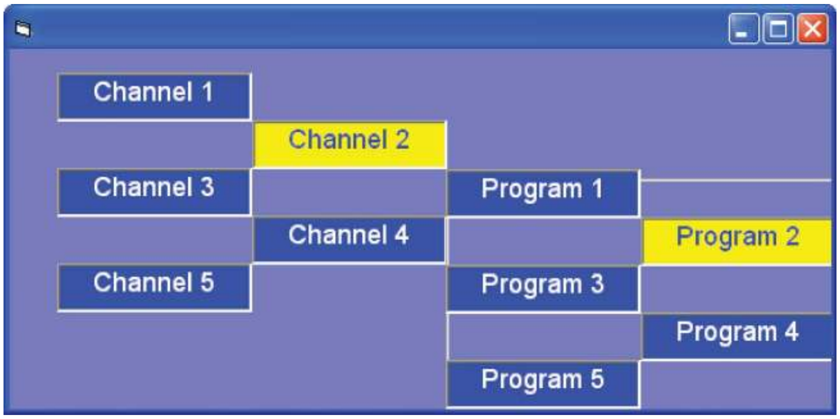


FIG. 28. Interface for people with motor impairment (color figure available online).

also investigated the effect of severe visual acuity loss for the following six font types (Figure 30):

- Microsoft Sans Serif
- Sabon
- Verdana
- Times New Roman
- Arial
- Georgia

It can be seen in Figure 30 that the legibility is not much different for different font types and nearly same for all.

Figure 31 shows the possible cursor trace of a person with motor impairment for the new interface. It can be seen that the

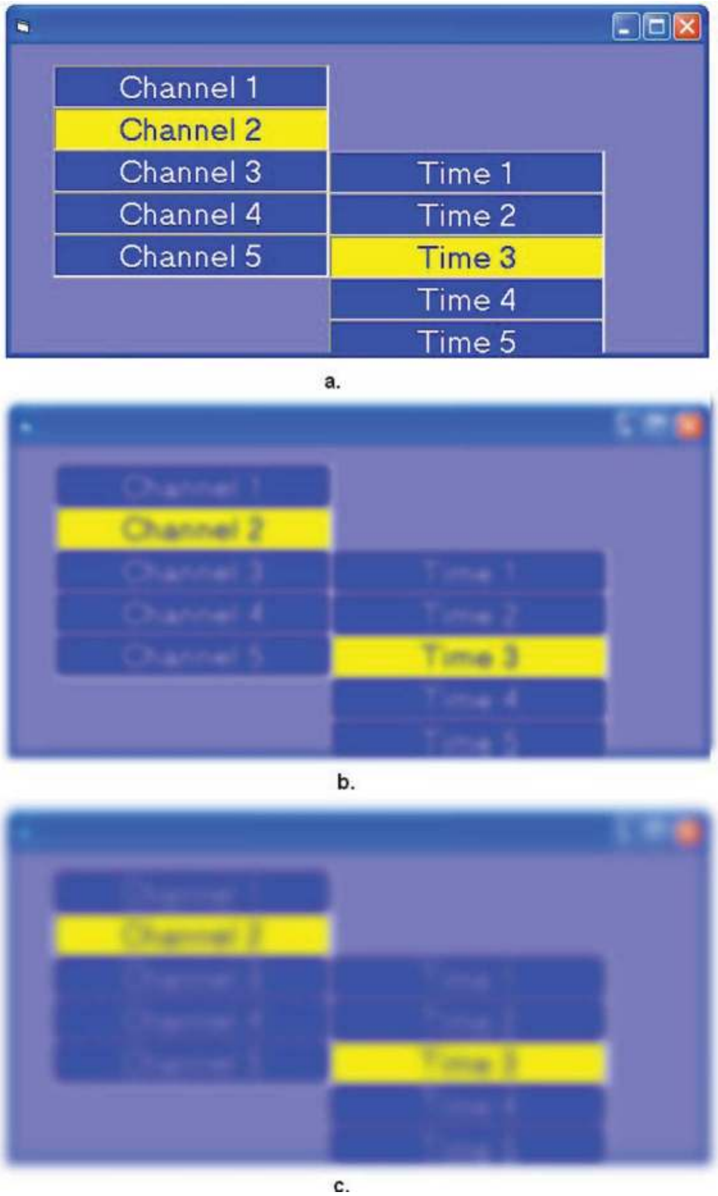


FIG. 29. Perception of the new interface of people having less visual acuity. (a) Original interface. (b) Interface perceived by mild visual acuity loss. (c) Interface perceived by severe visual acuity loss (color figure available online).

contour covering the area of missed click does not contain more than one menu item now. Finally we have evaluated the new interfaces with the following controlled experiment.

5.2. Validation

We validated the new interface through a user study. In this study it is hypothesized that

- People with visual acuity loss and motor impairment will perform a task faster and with less number of errors in the new interface (Figures 32 and 33) than the unchanged version (Figure 33).
- People with color blindness will perform a task equally well with respect to people with no impairment (control group) in the unchanged version of the interface (Figure 33).

We measured the task completion time as a measure of performance and the number of missed clicks as a measure of errors.

Procedure. The procedure mimics the process of selecting a channel from a list followed by selecting a program from a drop-down menu. Initially, the participants were shown a channel name and a program name. Then they made two

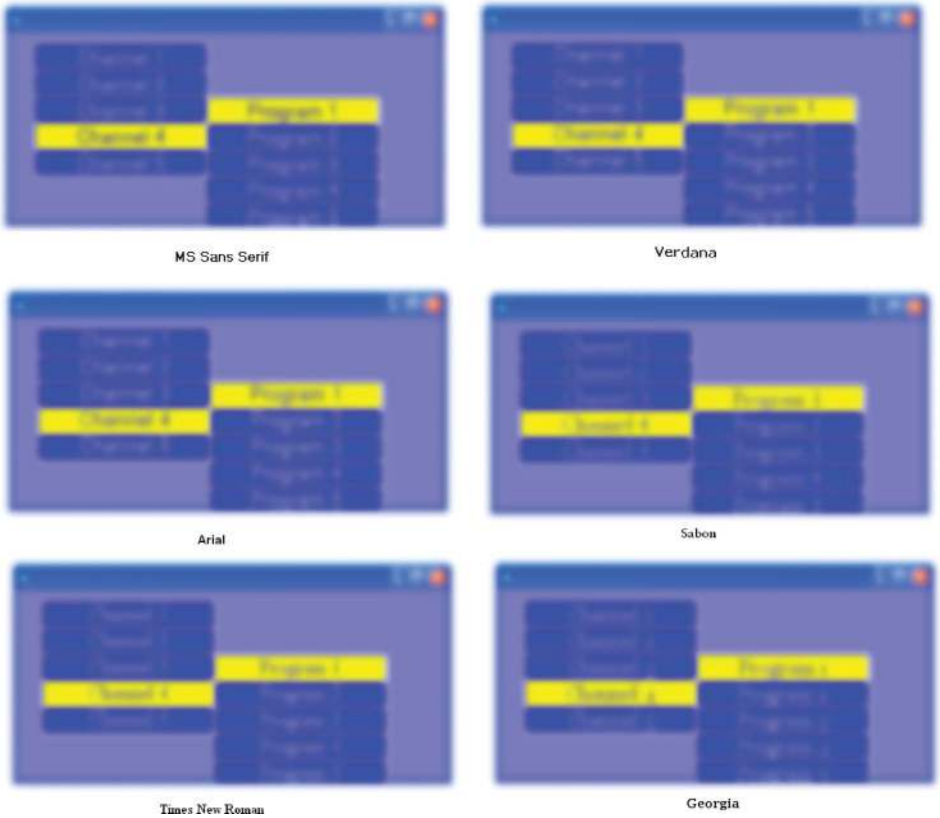


FIG. 30. Comparing different font types (color figure available online).

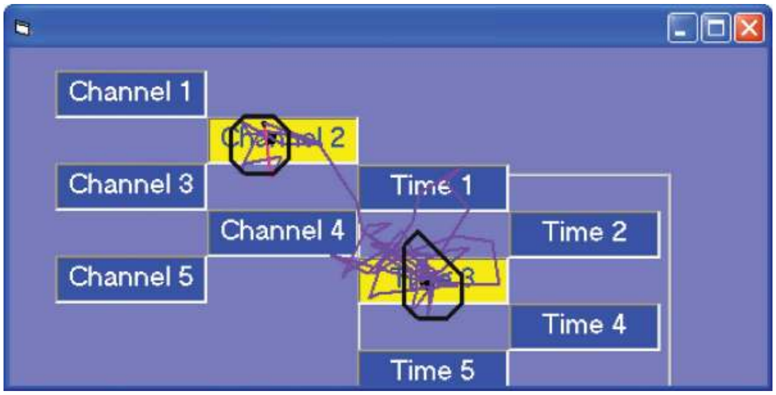


FIG. 31. Possible cursor trace of a mouse operated by a motor impaired person for the new interface (color figure available online).

selections matching the previously shown channel and program names. We did not use real channel and program names to avoid any biasness of users. The first two letters and length of all the captions were kept nearly same to avoid any pop-out effect (Treisman & Gelade, 1980) of the captions during visual search. The Verdana font type is used due to its bigger *x*-height and character spacing than other conventional fonts. Each participant repeated the task 10 times. All participants were trained before undertaking the study.

Material. We used a standard optical mouse and an Acer Aspire 1640 laptop with a 15.5-in. monitor with 1280 × 800 pixel resolution. We also used the same seating arrangement (same table height and distance from table) for all participants.

Participants. We collected data from two institutes, National Institute of Orthopedically Handicapped at Kolkata, India, and Papworth Trust at Cambridge, United Kingdom. All participants (Table 7) have some experience of using computers; either they were learning or using computers regularly. All of them volunteered for the study.

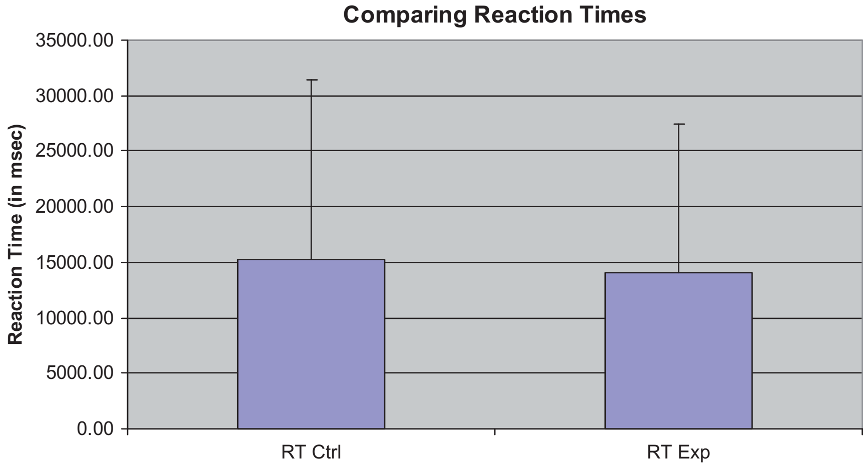


FIG. 32. Comparing reaction times for all participants (color figure available online).

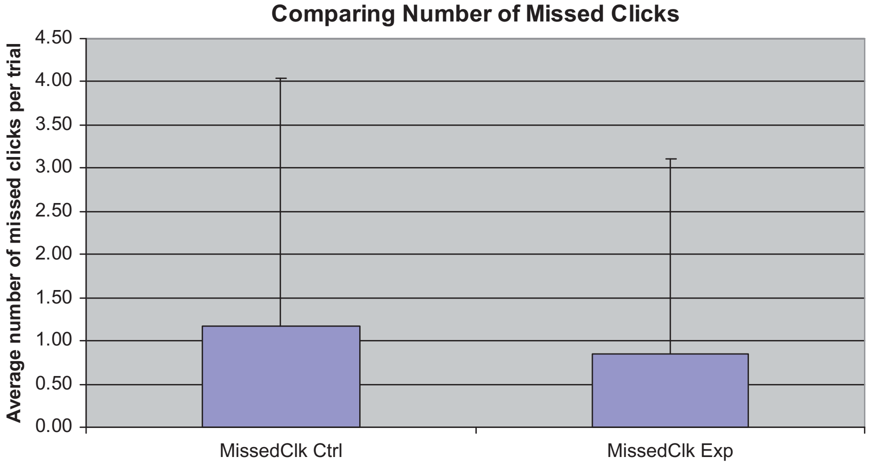


FIG. 33. Comparing number of missed clicks times for all participants (color figure available online).

Results. The average reaction time (total time needed to select the channel and program) was less in the new design than the control design (Figure 34) though the difference was not statistically significant in an independent sample two-tailed *t* test, $t(120) = 0.64$, $p > .05$. The average number of missed clicks were also less (Figure 33) in the new design than the control design, though the difference tends to statistical significance in a Wilcoxon ranked test, $W(120) = 163$, $p = .1$. In the experimental condition (new design), missed clicks occurred in 21 trials, whereas missed clicks occurred 31 times in control condition.

We have also analyzed the reaction times and missed clicks for each individual participant. Table 8 and Figures 34 and 35 show the average reaction time and total number of missed clicks for each participant. It can be seen that only four of 12 participants (P4, P5, P8, and P9) have an average reaction time greater for the experimental condition and only two of 12 participants (P8 and P12) missed clicked more in the experimental condition than the control condition.

Unfortunately we did not get any participants with color blindness. So we collected data from two able-bodied users with no visual impairment using a color-blindness filter (from Cambridge Research Systems, http://www.crsltd.com) to simulate the effect of dichromatic color blindness. In this case as well we did not find any significant difference in reaction times (Figure 36) in an independent sample two-tailed *t* test, $t(20) = 0.81$, $p > .05$, and did not record any missed clicks.

Discussion. The reaction time and number of missed clicks were both less in the new design, though we failed to find any statistical significance of the difference. Most participants did not have any problem in moving hands, and thus they could control the mouse movement pretty well. Except participant P1, the visual acuity loss was also not severe. In addition, in the present experimental setup, a missed click did not waste time, whereas in a real interface a missed click will take the user to an undesired channel and getting back to the previous screen will incur additional time. So a higher number of missed clicks in the

TABLE 7
List of participants

Participants	Age	Sex	Impairment
P1	>45	M	No mobility impairment. Age-related Hypermetropia (+3.75 / +3.25 Dioptre)
P2	25–45	M	Difficulty in walking, right leg is shorter than left leg. Mild myopia (–2.75 / –2 Dioptre)
P3	25–45	M	Right hand was cut in accident, no impairment in left hand. No visual impairment.
P4	25–45	M	No mobility impairment. Lost vision in right eye, left eye is perfect.
P5	25–45	M	Left arm is affected by polio, no impairment in right hand. No visual impairment.
P6	<25	F	Lower body is affected by polio from birth, no impairment in hands, wheelchair user. No visual impairment.
P7	<25	M	Difficulty in walking from birth. Slight Myopia (–0.7 / –0.7 Dioptre)
P8	44	M	Cerebral palsy reduced manual dexterity also some tremor in hand wheelchair user. Slight loss of visual acuity.
P9	63	M	Left side (nondominant) paralysed after a stroke in 1973 also has tremor
P10	31	M	Cerebral palsy reduced manual dexterity wheelchair user.
P11	>45	M	Reduced manual dexterity in limbs due to neurological problem, wheelchair user.
P12	44	F	Did not mention disease restricted hand movement no tremor. Slight loss of visual acuity.

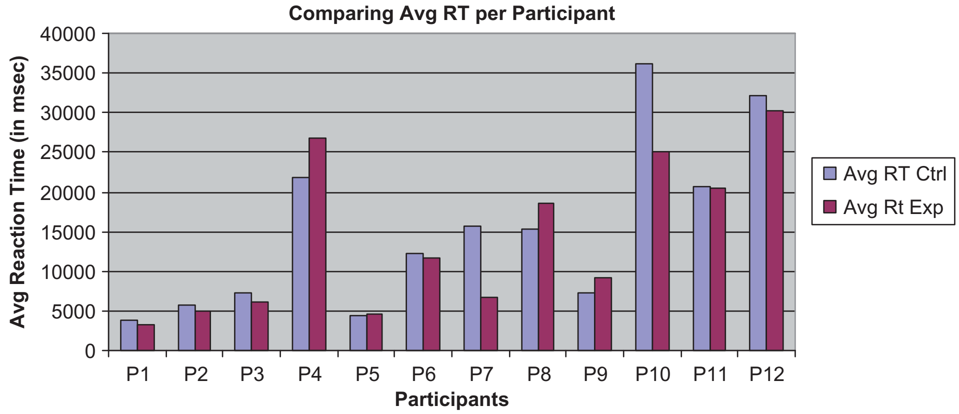


FIG. 34. Comparing reaction times for each participant (color figure available online).

control condition will also increase the channel selection time further in an actual scenario. However, in the future we plan to run the study with more a cautious selection of participants. All of the visually impaired participants preferred the bigger font size. However, a few participants reported difficulty in reading the zigzag presentations of the captions of the new interface. In the future we also plan to use an eye tracker to compare the visual search time for both types (linear and zigzag) of organizations of menu captions.

This study addresses a small segment of accessibility issues related to digital TV interfaces. Future studies will include more interaction modalities (like keypad or gesture-based interaction), devices (like remote control, set-top box, etc.), and impairments (like cognitive impairments). However, the results

TABLE 8
Result per participant

	AvgRT Ctrl (in msec)	AvgRT Exp (in msec)	TotalMC Ctrl	TotalMC Exp
P1	3,886	3,259	0	0
P2	5,755	5,033	0	0
P3	7,230	6,149	0	0
P4	21,777	26,838	72	56
P5	4,481	4,611	0	0
P6	12,195	11,739	11	4
P7	15,628	6,747	13	0
P8	15,394	18,628	20	28
P9	7,213	9,184	0	0
P10	36,160	25,084	11	0
P11	20,752	20,550	14	8
P12	32,228	30,223	0	6
Avg	**15,225**	**14,004**	**11.8**	**8.5**

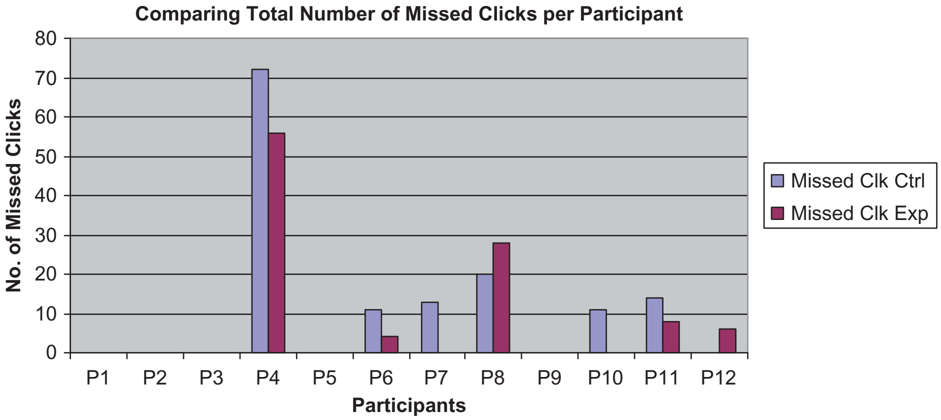


FIG. 35. Comparing number of missed clicks for each participant (color figure available online).

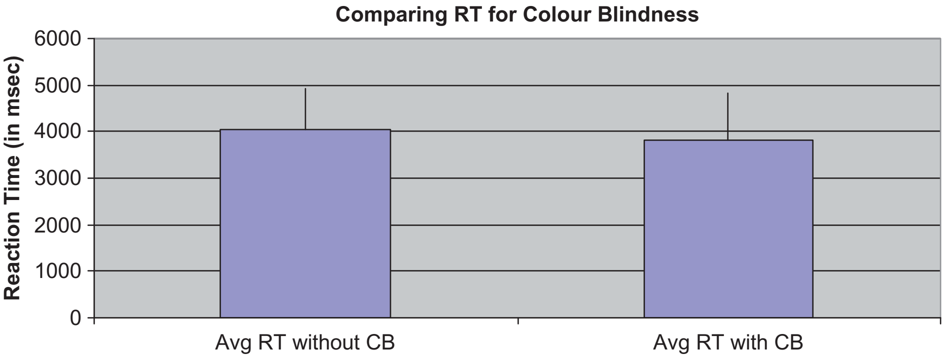


FIG. 36. Comparing reaction times for effect of color blindness (color figure available online).

of this study can be extended beyond program menu interfaces of digital televisions. For example, the font size of captions in absolute terms (*x*-height ≈ 0.5 cm.) indicates the minimum font size required for any text in an interface for serving people with severe visual acuity loss. Similarly the particular color combination of the screen (white text on a blue background) can be used in any other interface as well to cater people with color blindness. Finally, the modified menu structure can be used in computers or other digital devices to make the menus accessible to people with mobility impairment.

6. THE GUIDE SYSTEM

Currently the simulator is being used in the GUIDE Project. The GUIDE Project (http://www.guide-project.eu/) is a medium-scale, focused research project on an EU FP7, Accessible and Assistive Information and Communication Technologies grant, "GUIDE–Gentle User Interfaces for Disabled and Elderly Citizens." It aims to develop a toolbox of adaptive, multimodal user interfaces that target the accessibility requirements of elderly and impaired users in their home environment, making use of TV set-top boxes as processing and connectivity platform. The objectives of this project are as follows:

- Developing an intelligent software layer for multimodal adaptation of user interfaces.
- Developing novel user interface technologies (gesture, multitouch, adaptive user interface rendering, virtual avatars) suitable for different impairments.
- Conducting extensive tests with impaired users to better understand their requirements.
- Developing a set of reference applications for home automation, video conferencing, media navigation and tele-learning based on GUIDE.
- Developing a multimodal hardware interface that integrates a set of input modalities (visual gesture recognition and microphone beamforming).

Figure 37 shows a schematic overview of the GUIDE user model and hub. The use of the user models and user profiles will help it to address more varieties of users than the existing similar system (Epelde et al., 2009).

The development of GUIDE adaptive toolbox will have two main parts:

- Development of the user model
- Development of the adaptation algorithms

The GUIDE user model will be developed based on empirical data collected from users with a wide range of abilities (Figure 37). It will cluster users' data into user profiles and predict interaction patterns for a different user profile. The GUIDE Hub will use the model and the user profiles to adapt interfaces (Figure 37). The adaptation algorithms can broadly be classified into

- Static adaptation
- Dynamic adaptation

The static adaptation algorithm will take part before and at the end of an interaction. It will select appropriate modalities of interaction for users and update individual user profiles after the interaction. The dynamic adaptation algorithm will run while the user is interacting, and it will adapt the interface in run time to suit users' capabilities and context of use.

The GUIDE user model will accept help from the simulator to select the appropriate modality of interaction for each user.

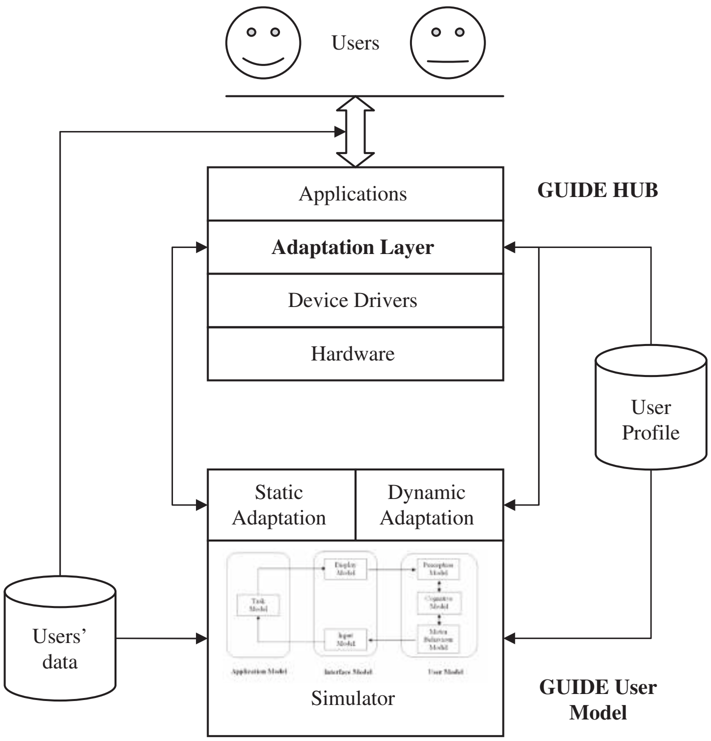


FIG. 37. Schematic diagram of the GUIDE toolbox and Hub.

The simulator will also change the interface layout based on the context of use and the interaction patterns of users. For example, if a user has less visual acuity, the font size of the screen will automatically be increased. If someone has a spasm or tremor in a finger, as with cerebral palsy, one may find the remote control hard to use and the simulator will invoke alternative interaction devices for that person. The personalization will also be invoked by change in context of use. For example, if the simulator detects a higher level of background noise, it may automatically increase the volume. The following two subsections elaborate the ideas of static and dynamic adaptation.

6.1. Static Adaptation

This example demonstrates a case study of using the simulator within the context of the GUIDE Project to select input and output modalities of interaction and personalizing interfaces from user profile. Initially we selected a set of user characteristics and modalities of interaction. Then we presented conditions for selecting different modalities for different types of users. In the last part of the document, we extended the idea of modality selection to interface personalization in general.

User profile

- Visual Acuity after correction (VA)
- Presence and type of Colour Blindness (CB)
- Grip Strength (GS)
- Range of Motion of Forearm (ROM)
- Capacity of Short Term Memory (STM)

Input modalities

- Physical Keys like Remote Control (K)
- Pointing Device (P)
- Pointing with Haptic feedback (PH)
- Gesture based (G)
- Voice based (V)

Output modalities

- Screen (S)
- Audio (A)

Assumptions

We have considered the following assumptions based on the results obtained from our simulator (Biswas, 2010).

- People having less than –5.5 Dioptre Visual Acuity will not prefer to use screen and thus pointing devices.
- People having less than 20 kg Grip Strength will prefer to use the pointing device with haptic feedback more than the ordinary one.
- People with color blindness (especially red–green color blindness) will prefer white text on a blue background on the screen.
- People having less capacity of short-term memory will prefer to use a reduced command set for gesture-based and voice-based input.

Flow chart

These assumptions are diagrammatically represented in Figure 38. The Yes/No options of the flowchart can be replaced with probability values after collecting enough data to calculate the prior probabilities. Then the flow chart can be implemented using a probabilistic rule-based system (like CLIPS) or can be converted to a Bayesian Network (Griffiths, Kemp, & Tenenbaum, 2008).

6.2. Dynamic Adaptation

The predicted cursor trace (Figure 8) from the simulator helps in developing dynamic adaptation algorithms. Figure 39 shows cursor traces on three different buttons in an interface. It can be seen that if the user aims to click at the middle of the button, then in spite of the random movements, there is no chance of miss-clicking neighboring buttons. However, if the user aims to click at the edge of the button, as in the case of

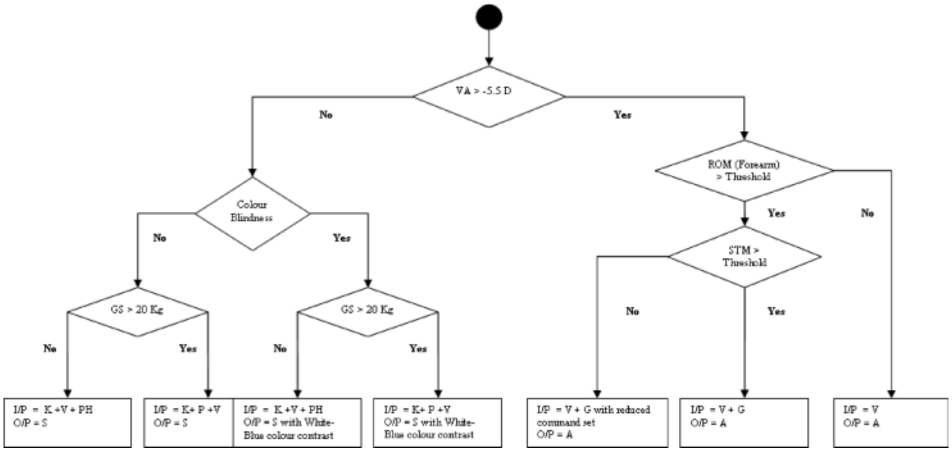
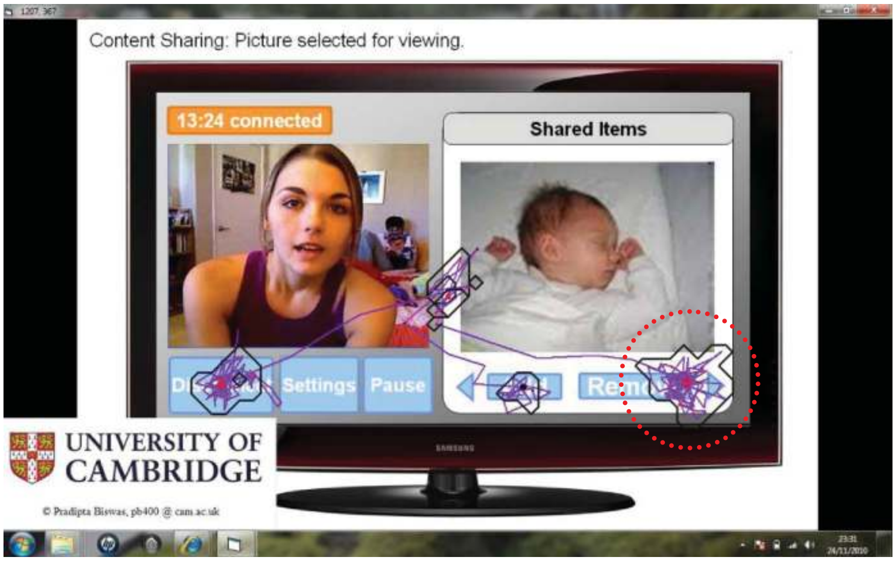
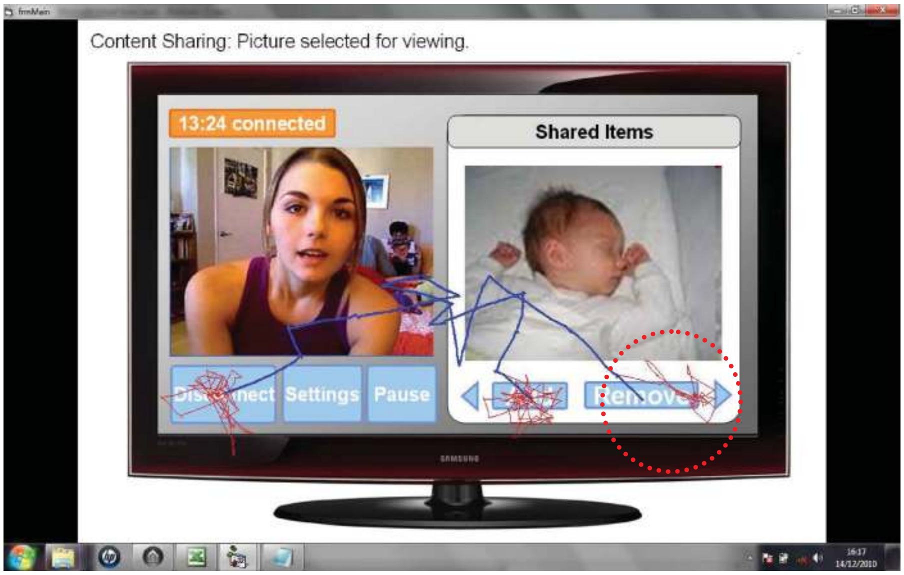


FIG. 38. Flowchart to select interaction modalities.



Content Sharing: Picture selected for viewing.

FIG. 39. An example of missed clicking (color figure available online).



Content Sharing: Picture selected for viewing.

FIG. 40. An example of the gravity well (color figure available online).

the rightmost button labeled "Disconnect," the user may end up clicking on a neighboring button—in this case the right arrow key (marked with dotted red circle). We can alleviate this problem by employing an adaptation principle called gravity well (Hwang, Langdon, Keates, Clarkson, & Robinson, 2002). The gravity well will attract the pointer in the middle of a button, if it is in vicinity of the button. So even if the user points toward the edge of a button, the pointer will automatically move to the center of the button. The thick blue line in Figure 40 shows the modified cursor traces after employing the gravity well, and the dotted red circle highlights how the cursor has been attracted to the middle of the Disconnect button influenced by the gravity well. Similar analysis can also be done for other adaptation algorithms like cursor path averaging, damping (Hwang et al., 2002), and setting parameters for those algorithms. In fact, we have already used the simulator to design and evaluate accessible interfaces and interaction techniques (Biswas, 2010).

7. IMPLICATIONS AND LIMITATIONS OF USER MODELING

User trials are always expensive in terms of both time and cost. A design evolves through an iteration of prototypes, and if each prototype is to be evaluated by a user trial, the whole design process will be slowed down. Buxton (1994) also noted that "while we believe strongly in user testing and iterative design. However, each iteration of a design is expensive. The effective use of such models means that we get the most out of each iteration that we do implement." In addition, user trials are not representative in certain cases, especially for designing inclusive interfaces for people with special needs. A good simulation with a principled theoretical foundation can be more useful than a user trial in such cases. Exploratory use of modeling can also help designers to understand the problems and requirements of users, which may not always easily be found through user trials or controlled experiments.

This work shows that it is possible to develop engineering models to simulate HCI of people with a wide range of abilities and that the prediction is useful in designing and evaluating interfaces. According to Allen Newell's time scale of human action, our model works in the cognitive band and predicts activity in the millisecond to second range. It cannot model activities outside the cognitive band like microsaccadic eye gaze movements, response characteristics of different brain regions (in biological band; Newell, 1990), affective state, social interaction, consciousness (in rational and social band; Newell, 1990), and so on. Simulations of each individual band have their own implications and limitations. However, the cognitive band is particularly important because models working in this band are technically feasible, experimentally verifiable, and practically usable. Research in computational psychology and more recently in cognitive architectures supports this claim. We have added a new dimension in cognitive modeling by including users with special needs. Thus it helps to implement Universal Design or Inclusive Design processes by helping to visualize, understand, and measure the effect of different impairments on interaction.

However, this system should not be used in isolation. It is not designed to capture subjective choice or preference of users; the simulator should complement existing qualitative techniques (Springett, 2008) or participatory designs (Ellis & Kurniawan, 2000) for assessing users' experience.

8. CONCLUSION

In this article we have presented a literature survey on human behavior simulation and their applications on modeling users in HCI. The review of the current state-of-the-art work shows a deficiency of modeling tools for users with disabilities. We have developed a simulator to address the problems in existing modeling techniques. It should be evident that the use of modeling and the type of model to be used depend on many factors like the application, the designers, availability of time and cost for design, and so on. However, we hope this article will give system analysts an understanding of different modeling paradigms, which in turn may help them to select the proper type of model for their applications.

REFERENCES

Anderson, J. R., & Lebiere C. (1998). *The atomic components of thought*. Hillsdale, NJ: Erlbaum.

Barnard, P. (2011). The Emotion Research Group website, MRC Cognition and Brain Sciences Unit. Retrieved from http://www.mrc-cbu.cam.ac.uk/research/emotion/

Benyon, D., & Murray D. (1993). Applying user modeling to human computer interaction design. *Artificial Intelligence Review*, *7*,199–225.

Bernard, M., Liao, C. H., & Mills, M. (2001). The effects of font type and size on the legibility and reading time of online text by older adults. *ACM/SIGCHI Conference on Human Factors in Computing Systems 2001*, 175–176.

Bernard, M., Mills, M., Peterson, M., & Storrer, K. A. (2001, July). A comparison of popular online fonts: Which is best and when? Usability News, 3(2). Retrieved from http://www.surl.org/usabilitynews/32/font.asp

Beymer, D., Russell, D. M., & Orton, P. Z. (2007). An eye tracking study of how font size, font type, and pictures influence online reading. *Proceedings of the 22nd British Computer Society Conference on Human–Computer Interaction 2007*, 15–18.

Biswas, P. (2010). *Inclusive user modelling.* Unpublished doctoral dissertation, University of Cambridge, England.

Biswas, P., Bhattacharyya, S., & Samanta, D. (2005). User model to design adaptable interfaces for motor-impaired users. *Tencon '05 - IEEE Region 10 Conferences 2005.* 1801–1844.

Biswas, P., & Robinson, P. (2008). Automatic evaluation of assistive interfaces. *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI) 2008*, 247–256.

Biswas, P., & Robinson, P. (2009a). Modelling perception using image processing algorithms. *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, pp. 494–503.

Biswas, P., & Robinson, P. (2009b). Predicting pointing time from hand strength. *Proceedings of the Usability & HCI for e-Inclusion, 5th Symposium of the Austrian Computer Society (USAB 2009), LNCS 5889*, 428–447.

Blandford, A., Butterworth, R., & Curzon, P. (2004). Models of interactive systems: A case study on programmable user modelling. *International Journal of Human-Computer Studies*, *60*, 149–200.

Boden, M. A. (1985). Computer models of mind: Computational approaches in theoretical psychology. Cambridge, UK: Cambridge University Press.

Bovaria, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction*, *5*, 1–48.

Boyarski, D., Neuwirth, C., Forlizzi, J., & Regli, S. H. (1998). A study of fonts designed for screen display. *ACM/SIGCHI Conference on Human Factors in Computing Systems 1998*, 87–94.

Butterworth, R., & Blandford, A. (1997). *Programmable user models: The story so far*. Retrieved from http://www.cs.mdx.ac.uk/puma/wp8.pdf

Buxton, W., with Billinghurst, M., Guiard, Y., Sellen, A., & Zhai, S. (1994). *Human input to computer systems: Theories, techniques and technology*. Unpublished manuscript. Retrieved October 27, 2009, from http://www.billbuxton.com/inputManuscript.html

Byrne M. D. (2001). ACT–R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human–Computer Studies*, *55*, 41–84.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human–computer interaction*. Hillsdale, NJ: Erlbaum.

Carroll, J. M., & Olson, J. M. (1990). Mental models in human–computer interaction. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 135–158). Amsterdam, the Netherlands: Elsevier.

Cognitive architecture. (2007). In *Wikipedia*. Retrieved July 1, 2007, from http://en.wikipedia.org/wiki/Cognitive_ architecture

Ellis, R. D., & Kurniawan, S. (2000). Increasing the usability of online information for older users: A case study in participatory design. *International Journal of Human–Computer Interaction*, *12*, 263–276.

Eng, K., Lewis, R. L., Tollinger, I., Chu, A., Howes, A., & Vera, A. (2006). Generating automated predictions of behavior strategically adapted to specific performance objectives. *ACM/SIGCHI Conference on Human Factors in Computing Systems 2006*, 621–630.

Epelde, G., Carrasco, E., Zimmermann, G., Bund, J., Dubielzig, M., & Alexandersson, J. (2009). URC based accessible TV. *Proceedings of the Seventh European Conference on European Interactive Television Conference*, 111–114.

Field A. (2009). *Discovering statistics using SPSS*. Thousand Oaks, CA: Sage.

Fitts P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*, 381–391.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2007). Automatically generating user interfaces adapted to users' motor and vision capabilities. *ACM Symposium on User Interface Software and Technology 2007*, 231–240.

Gray, W. D., & Sabnani, H. (1994). Why you can't program your VCR, or, predicting errors and performance with production system models of display-based action. *Conference Companion on Human Factors in Computing Systems in ACM/SIGCHI Conference on Human Factors in Computing Systems 1994*, 79–80.

Gray, W., Young, R. M., & Kirschenbaum, S. (1997). Introduction to this special issue on cognitive architectures and human-computer interaction. *Human-Computer Interaction*, *12*, 301–309.

Griffiths, T. L., Kemp, C., & Tenenbaum, J. B. (2008, July). *Bayesian models of inductive learning*. Tutorial at the Annual Meeting of the Cognitive Science Society, Washington, DC.

Hampson, P. J., & Morris, P. E. (1996). *Understanding cognition*. Oxford, UK: Blackwell.

Hick, W. E. (1952). On the rate of gain of information. *Journal of Experimental Psychology*, *4*, 11–26.

Hornof, A. J., & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. *ACM/SIGCHI Conference on Human Factors in Computing Systems 1997*, 107–114.

Horovitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumiere Project: Bayesian user modeling for inferring the goals and needs of software users. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 256–265. Available from http://research.microsoft.com/en-us/um/People/horvitz/lumiere.htm

Howes, A., Vera, A., Lewis, R. L., & McCurdy, M. (2004). Cognitive constraint modeling: A formal approach to reasoning about behavior. In K. D. Forbus, D. Gentner & T. Regier (Eds.), *26th Annual Meeting of the Cognitive Science Society, CogSci'04* (pp. 595–600). Mahwah, NJ: Erlbaum.

Hurst, A., Hudson, S. E., & Mankoff, J. (2010). Automatically identifying targets users interact with during real world tasks. *ACM International Conference on Intelligent User Interfaces*. pp. 11–20.

Hwang, F., Langdon, P. M., Keates, S., Clarkson, P. J., & Robinson, P. (2002). Cursor characteristics and haptic interfaces for motor-impaired users. *Cambridge Workshop on Universal Access and Assistive Technology*, 87–96.

John, B. E. (2010, March). *Reducing the variability between novice modelers: Results of a tool for human performance modeling produced through human-centered design*. Paper presented at the 19th Conference on Behavior Representation in Modeling and Simulation, Charleston, SC.

John, B. E., & Kieras, D. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer Human Interaction 3*, 320–351.

John, B. E., Prevas, K., Salvucci, D., & Koedinger, K. (2004). Predictive human performance modeling made easy. *ACM/SIGCHI Conference on Human Factors in Computing Systems*. pp. 455–462.

Johnson, P. (1992). *Human–computer interaction: Psychology, task analysis and software engineering*. New York, NY: McGraw-Hill.

Johnson-Laird, P. A. (1988). *The computer and the mind*. Cambridge, MA: Harvard University Press.

Keates, S., Clarkson, J., & Robinson, P. (2000). Investigating the applicability of user models for motion-impaired users. *ACM/SIGACCESS Conference on Computers and Accessibility*, 129–136.

Kieras, D. (1994). GOMS modeling of user interfaces using NGOMSL. *Conference Companion on Human Factors in Computing Systems,*371–372.

Kieras, D. E. (2005, July). *Fidelity issues in cognitive architectures for HCI modelling: Be careful what you wish for*. Paper presented at the 11th International Conference on Human Computer Interaction,Las Vegas, NV.

Kieras, D., & Meyer, D. E. (1990). An overview of the EPIC architecture for cognition and performance with application to human–computer interaction. *Human-Computer Interaction*, *12*,391–438.

Kieras, D. E., Wood, S. D., Abotel, K., & Hornof, A. (1995). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. *ACM Symposium on User Interface and Software Technology*, 91–100.

Laird, J. E., Rosenbloom, P. S., & Newell, A. Towards chunking as a general learning mechanism. *Proccedings from the National Conference on Artificial Intelligence*, 188–192.

Lallement, Y., & Alexandre, F. (1997). Cognitive aspects of neurosymbolic integration. In R. Sun & F. Alexandre (Eds.), *Connectionist-symbolic integration*. London, UK: Erlbaum.

Mankoff, J., Fait, H., & Juang, R. (2005). Evaluating accessibility through simulating the experiences of users with vision or motor impairments. *IBM Systems Journal*, *44*, 505–518.

Marr, D. C. (1980). Visual information processing: The structure and creation of visual representations. *Philosophical Transactions of the Royal Society of London*, *290*, 199–218.

Marsden, G., & Jones, M. (2001). Ubiquitous computing and cellular handset interfaces—Are menus the best way forward? *Proceedings of the SAICSIT Annual Conference: Hardware, Software and Peopleware: The Reality in the Real Millennium*, 111–119.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *7*,115–133.

McGuffin, M., & Balakrishnan, R. (2005). Fitts' Law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer–Human Interaction*, *12*, 388–422.

Mcmillan, W. W. (1992). Computing for users with special needs and models of computer-human interaction. *ACM/SIGCHI Conference on Human Factors in Computing Systems*, 143–148.

Moran, T. P. (1981). Command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies*, *15*, 3–50.

Motomura, Y., Yoshida, K., & Fujimoto, K. (2000). Generative user models for adaptive information retrieval. *Proceedings of the IEEE Conference on Systems.* pp. 665–670.

Newell, A. (1973). *You can't play 20 questions with nature and win. Projective comments on the papers of this symposium*. Pittsburgh, PA: Carnegie Mellon University, Department of Computer Science.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1995). *GPS, a program that simulates human thought*. Cambridge, MA: MIT Press.

Nilsen E. L. (1992). *Perceptual-motor control in human-computer interaction*. Ann Arbor: The Cognitive Science and Machine Intelligence Laboratory, the University of Michigan.

Norcio, F., & Chen, Q. (1992). Modeling user's with neural architecture. *Proceedings of the International Joint Conference on Neural Networks*, 547–552.

Norcio F. (1989). Adaptive interfaces: Modelling tasks and users. *IEEE Transactions on Systems, Man, and Cybernetics*, *19*,399–408.

Oka, N. (1991). Hybrid cognitive model of conscious level processing and unconscious level processing. *IEEE International Joint Conference on Neural Networks*, 485–490.

Parush, A., Nadir, R., & Shtub, A. (1998). Evaluating the layout of graphical user interface screens: Validation of a numerical computerized model. *International Journal of Human-Computer Interaction*, *10*, 343–360.

Payne, S. J., & Green, T. R. G. (1986). Task-action grammars: A model of mental representation of task languages. *Human-Computer Interaction*, *2*, 93–133.

Peck, V. A., & John, B. E. (1992). Browser-soar: A computational model of a highly interactive task. *ACM/SIGCHI Conference on Human Factors in Computing Systems*, 165–172.

Reisner, P. (1981). Formal grammar and human factors design of an interactive graphics system. *IEEE Transactions on Software Engineering*, *7*,229–240.

Rieman, J., & Young, R. M. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies*, *44*, 743–775.

Ruiz, J., & Lank, E. (2010). Speeding pointing in tiled widgets: Understanding the effects of target expansion and misprediction. *ACM International Conference on Intelligent User Interfaces*, 229–238.

Salvucci, D. D., & Lee, F. J. (2003). Simple cognitive modelling in a complex cognitive architecture. *ACM/SIGCHI Conference on Human Factors in Computing Systems*, 265–272.

Serna, A., Pigot, H., & Rialle, V. (2007). Modeling the progression of Alzheimer's disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction*, *17*, 415–438.

Springett, M. (2008). Assessing user experiences within interaction: Experience as a qualitative state and experience as a causal event. *Proceedings of Meaningful Measures: Valid Useful User Experience Measurement (VUUM).*

Stephanidis, C., & Constantinou, P. (2003). Designing human–computer interfaces for quadriplegic people. *ACM Transactions on Computer-Human Interaction*, *10*, 87–118.

Stephanidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Maou, N., Leventis, A., . . . Karagiannidis, C. (1998). Adaptable and adaptive user interfaces for disabled users in the AVANTI Project. *Intelligence in Services and Networks, LNCS-1430*, 153–166.

Tollinger, I., Lewis, R. L., McCurdy, M., Tollinger, P., Vera, A., Howes, A., & Pelton, L. (2005). Supporting efficient development of cognitive models at multiple skill levels: Exploring recent advances in constraint-based modeling. *ACM/SIGCHI Conference on Human Factors in Computing Systems*, 411–420.

Treisman, A., & Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, *12*, 97–136.

Wendell, S. (1996). *The rejected body: Feminist philosophical reflections on disability*. New York, NY: Routledge.

World Bank. (2011). Retrieved from http://web.worldbank.org

World Health Organization. (2011). Our ageing world [Web page]. Retrieved from http://www.who.int/ageing/en

Yoshida, K., & Motoda, H. (1996). Automated user modeling for intelligent interface. *International Journal of Human-Computer Interaction*, *8*, 237–258.

Young, R. M., Green, T. R. G., & Simon, T. (1989). Programmable user models for predictive evaluation of interface designs. *ACM/SIGCHI Conference on Human Factors in Computing Systems*, 15–19.

ABOUT THE AUTHORS

Pradipta Biswas is a Research Associate at the Engineering Design Centre and Trinity Hall of University of Cambridge. He is member of the British Computer Society, Royal Society of Medicine, the EU Task Force on Standardization of User Models and a working group coordinator of the ITU focus group on Audiovisual Media Accessibility.

Peter Robinson is Professor of Computer Technology and Deputy Head of Department at the University of Cambridge Computer Laboratory in England and a Fellow, Praelector and Director of Studies in Computer Science at Gonville & Caius College.

Pat Langdon is a Senior Research Associate at the Engineering Design Centre of University of Cambridge. He is currently External Examiner for the Kings College London and Guy's Hospital Intercollegiate MSc in Assistive Technology.