A brief survey on user modelling in HCI

Pradipta Biswas and Peter Robinson
{pb400, pr10} @ cam.ac.uk
Computer Laboratory, University of Cambridge, United Kingdom

Abstract— In this paper, we present a brief survey of different user modelling techniques used in human computer interaction. We have investigated about the origin of user modelling and then classified the existing models into different categories. We have discussed a few examples of user models in each category, pointed out their advantages and disadvantages and finally concluded by presenting a few open questions on user modelling. We hope this paper will help system analysts to select the proper type of model for their applications.

Index Terms—User Model, Human Computer Interaction, Cognitive Model.

1 INTRODUCTION

"Our ultimate objective is to make programs that learn from their experience as effectively as humans do. We shall…say that a program has common sense if it automatically deduces for itself a sufficient wide class of immediate consequences of anything it is told and what it already knows. "- John McCarthy, from his paper, "Programs with Common Sense," 1958. Quoted by Daniel Crevier, "The Tumultuous History of the Search for Artificial Intelligence," 1993

Addressing a large variety of users is always a challenge to designers due to diverse range of abilities and differences in task, prior knowledge and situation. A user model is a representation of the knowledge and preferences of users [3]. It is not a mandatory part of a software but it helps to get the system serve the user better. Any information stored about the user or usage pattern (like event log or user log) is not a user model unless it can be used to get some explicit assumption about the user.

1.1 History on modelling user

Research on simulating user behaviour to predict machine performance was originally started during the Second World War. Researchers tried to explore operators' limitations of different military hardware by applying the discoveries of different experiments in psychological laboratories. During the same time, computational psychologists were trying to model the mind by considering it as an ensemble of processes or programs. The early attempts of cognitive modelling [10, 18] include the use of various mathematical models like Bayes' decision model (e.g. Edwards [19] probabilistic information processor) or Shanon's information theory. McCulloch and Pitts' model of neuron and subsequent models of neural networks, Marrs model of vision are a few examples of influential works in this discipline. Boden [10] presents a detailed discussion of such computational mental models. In the late 70s, as interactive computer systems became cheaper and accessible to more people, modelling human computer interaction (HCI) also gained much attention. However, models like Hick's law [29] (used for predicting visual search time) or Fitts' law [21] (used for prediction of movement time) were individually not enough to simulate a whole interaction.

The Command Language Grammar [45] developed by Moran at Xerox Parc could be considered as 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 Model Human Processor (MHP) [15] was an important milestone in modelling HCI since it introduced the concept of simulating HCI from the perspective of users. It gave birth to the GOMS family of models [15, 33] that are still the most popular modelling tools in HCI.

Outside the domain of HCI, recent researches on cognitive modelling address a wide range of topics such as investigating mental processes for new idea generation [65], speech perception [63], bilingualism [43], knowledge representation, learning [26] and so on. However the domain of cognitive modelling is currently overwhelmed by the cognitive architectures and models developed using them. This kind of models does not only work 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 Newell [47] 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 1972 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 in psychological laboratories. Since these models are capable (or at least demanded to be capable) of simulating any type of user behaviour, they are also often used to simulate the behaviour of users while interacting with a computer. Gray and colleagues [25] assert 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 modelling: the GOMS family of models was developed only for HCI while the models involving cognitive architectures took a more detailed view of human cognition. Based on the accuracy, detail and completeness of these models, Kieras classified them as low fidelity and high fidelity models respectively [40]. 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 while cognitive architectures aim to represent the users' mental model [16]. The Sequence/Method representation assumes that all interactions consist of a sequence of operations or generalized methods, while 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 [56], Task Action Grammar [53] etc.). These models represent an interaction by using formal grammar where each action is modelled 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 as extensively as users' behaviour simulator [16].

In the following sub sections, we briefly describe these different types of user models. Then, we present a critical review of the existing models and conclude by highlighting a few open questions in user modelling.

2 THE MODELS

2.1 The GOMS family of models

GOMS stands for Goals, Operators, Method and Selection. It was inspired by the GPS system [48] developed by Newell. 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 sub goals. There exist many variations of the original GOMS model. The KLM model [15] simplifies the GOMS model by eliminating the goals, methods, and selection rules, leaving only six primitive operators. They are:

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, and
6) waiting for the computer to execute a command.

The durations for each 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 NGOMSL (Natural GOMS Language) [38]. Originally, it was an attempt to represent the content of a CCT model [34] at a higher level of notation. CCT is a rule-based system developed by Bovair and colleagues [11] 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 also developed a modelling tool, GLEAN (GOMS Language Evaluation and Analysis) [37], to execute NGOMSL. It simulates the interaction between a simulated user with a simulated device for undertaking a task.

John and Kieras [33] 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 developed the Soar architecture [49] as a possible candidate for his unified theories of cognition. According to Newell [49] and Johnson-Laird [35], the vast variety of human response functions for different stimuli in the environment can only be explained by a symbolic system. So the Soar system models human cognition as a production-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)" [49]. 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 and colleagues [41] successfully explained the power law of practice through the chunking mechanism.

However, there are certain aspects of human cognition (like perception, recognition, motor action) that can better be explained by a connectionist approach than a symbolic one [52]. It is believed that initially conscious processes control our responses to any situation while after sufficient practice, automatic processes are in charge for the same set of responses [28]. Lallement and Alexandre [42] have classified all cognitive processes into synthetic or analytical processes. Synthetic operations are concerned with low-level, non-decomposable, unconscious, perceptual tasks. In contrast, analytical operations signify high level, conscious, decomposable, reasoning tasks. From the modelling point of view, synthetic operations can be mapped on to connectionist models while analytic operations correspond to symbolic models. Considering these facts, the ACT-R system [1] does not follow the pure symbolic modelling strategy of the Soar, rather it was developed as a hybrid model, which has both symbolic and sub symbolic 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 behaviour is modelled at the sub-symbolic 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 EPIC (Executive-Process/Interactive Control) [38] architecture pioneers to incorporate separate perception and motor behaviour modules in a cognitive architecture. It mainly concentrates on modelling 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 modelling of menu searching and icon searching tasks.

The CORE system (Constraint-based Optimizing Reasoning Engine) [20, 32, 64] 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 behaviour.

There exist additional cognitive architectures (like Interactive Cognitive Subsystems [2], Apex, DUAL, CLARION [17] etc.), but they are not yet as extensively used as the previously discussed systems.

2.3 Grammar-based models

The grammar based model (like Task action grammar [53] and Task action language[56]) 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 modelling 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 works has been done on user modelling for developing customizable applications. These models have the following generic structure (Figure 1). They maintain a user profile and use different types of AI systems to predict performance.

These type of models are particularly popular in online adaptable systems (like personalized search engines or portals). We discuss a few more examples besides online systems in the next sub-section.

The Generative User Model [46] is developed for personalized information retrieval. In this model user given query words are related to user's mental state and retrieved object using latent probabilistic variables. Norcio [51] 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 users. Norcio and Chen [50] also used an artificial neural network for the same purpose as their previous work [51]. The user's characteristic is stored as user image and neural networks are used as pattern associates or pattern classifiers to get user's knowledge, detect user's goal etc. Lumiere convenience project [31] used influence diagram in modeling users. Lumiere project is the background theory of the Office Assistant shipped with Microsoft Office application. The influence diagram models the relationships among users' needs, goals, user background etc. 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.

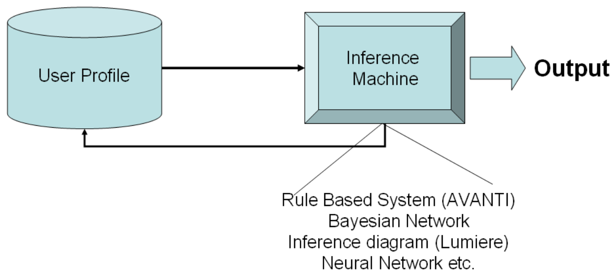

Figure 1. Application Specific User Models

3 REVIEW

The GOMS family of models is mainly suitable for modelling the optimal behaviour (skilled-behaviour) of users [33]. 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 or 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 do 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 sub-optimal behaviour of users. As it fails to model the sub-optimal behaviour, it cannot be used to predict the occurrences of different errors during interaction. These problems are common for any

Sequence/Method representations since these ways of representations overlook the underlying mental models of users [16].

On the other hand, cognitive architectures model the uncertainty of human behaviour in detail but they are not easily accessible to non-psychologists 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 [64] for the CORE architecture. Additionally, Kieras has shown that a high fidelity model cannot always outperform a low fidelity one though it is expected to do so [40].

Researchers have already attempted to combine the GOMS family of models and cognitive architectures to develop more usable and accurate models. Salvucci and Lee [59] have developed the ACT-Simple model by translating basic GOMS operations (like 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 and colleagues [9] implemented the Programmable User Model (PUM) [66] by using the Soar architecture. They developed a program, STILE (Soar Translation from Instruction Language made Easy), to convert the PUM Instruction Language into Soar productions. However, this approach also demands good knowledge of Soar on the part of an interface designer. Later, the PUM team identified additional problems with runnable user models and they are now investigating abstract mathematical models [12]. There also exist some application-specific models that combine GOMS models with a cognitive architecture. As for example, Gray and Sabnani [24] combined GOMS with ACT-R to model a VCR programming task, while Peck and John [54] used Soar to model interaction with a help-browser, which ultimately turned out to be a GOMS model.

Another problem of existing modelling approaches stems from issues related to disability. There is not much reported work on systematic modelling of assistive interfaces. McMillan [44] 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 [61, 62] modelled an assistive interface for a web browser based on static and dynamic characteristics of users. The interface is initialised according to static characteristics (e.g. age, expertise etc.) of the user. During interaction, the interface adapts itself based on dynamic characteristics (e.g. idle time, error rate etc.) of the user. This model does not address the basic perceptual, cognitive and motor behaviour of users and so it is hard to generalize to other applications. Keates and colleagues [36] measured the difference between able-bodied and motor-impaired users with respect to the Model Human Processor (MHP) [15] 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 use. Our previous user model [8] also took a more generalized approach than the AVANTI project. It broke down the task of user modelling 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.

Our work

In our current research, we address some of the current problems of user modelling by developing a simulator inspired by Model Human Processor [15]. The simulator embodies both the internal state of a computer application and also the perceptual, cognitive and motor processes of its user. It takes a task definition and locations of different objects in an interface as input. It then predicts possible eye movements and cursor paths on the screen and uses these to predict task completion times. It consists of a perception model, a cognitive model and a motor-behaviour model. The models are parameterized to represent different physical abilities, levels of skill and input devices.

The perception model [6] simulates the phenomena of visual perception (like focussing and shifting attention). It can also simulate the effects of different visual impairments (e.g.: wet and dry Maccular Degeneration, Diabetic Retinopathy, Tunnel Vision etc.) on interaction. We have investigated eye gaze patterns of able-bodied users as well as people with visual impairment and our model can predict the visual search time and eye gaze pattern of able-bodied people and a few types of visually-impaired users with statistically significant accuracy.

The cognitive model [5] simulates expert performance by using CPM-GOMS model [33]. It can also simulate performance of novices by using a dual-space model [57].

The motor-behaviour model [7] is developed by statistical analysis of cursor traces from motor-impaired users. As part of the model, we also develop a new scale of characterizing the extent of disability of users by measuring their grip strength, which was not earlier possible by using clinical scales [60].

These models do not need knowledge of psychology or programming to operate. They have graphical user interfaces to provide input parameters and showing output of simulation. The main contributions of our work are

1. Identification and calibration of two image processing algorithms to predict points of eye-gaze fixations and the corresponding fixation durations during visual search in a computer screen undertaken by people with and without visual-impairment.

2. Analysis of eye movement trajectories during visual search in a computer screen and identification of the most probable strategies to predict the actual trajectory.
3. Investigation of the effect of hand strength on human-computer interaction.
4. Development a statistical model to predict pointing times of motor-impaired computer users based on their hand strength.

Our studies are already being used to design and develop inclusive computer interfaces (e.g. accessible game [55], new assistive interaction technique [4]).

4 ISSUES WITH MODELLING

Now we are in a position to discuss modelling from a broader perspective than criticizing individual techniques. In particular we like to discuss the following issues.

4.1 Optimum fidelity

In which level of detail, a model should work? At one end, the application specific models simulate component-tasks of an application directly. But they are hard to generalize to another application. The cognitive architectures models basic rules of cognition, but it also made them hard to use. Additionally, each level of modelling introduces an error in prediction and if we drill down many such levels to model an interaction, the errors will only accumulate and make the prediction worse. In contrast, modelling at a higher level may miss detail of interactions between underlying components. Figure 2 presents a classification of human actions based on the duration of their completion. Simulations of each individual band have their own implications and limitations. However the cognitive band is particularly important since 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.

TIME SCALE OF HUMAN ACTION		
SCALE (sec)	SYSTEM	STRATUM
10^{7} 10^{6} 10^{5}		SOCIAL
10^{4} 10^{3} 10^{2}	Task Task Task	RATIONAL
10^{1} 10^{0} 10^{-1}	Unit Task Operations Deliberate Act	COGNITIVE
10^{-2} 10^{-3} 10^{-4}	Neural Circuit Neuron Organelle	BIOLOGICAL

Figure2. Timescale of human action (adapted from [49])

In our opinion, the fidelity of a model should depend upon the application – do we need the model to explain the underlying mental processes of an interaction or to predict performances of users for comparing two interfaces? In the first case we should go for a high fidelity model while the second case demands an engineering system which is easy to use and comprehend.

4.2 What to model

Most user models concentrate on modelling performance, however TAL and TAG challenge this view by modelling knowledge and competence of users. The cognitive architectures can also be used to estimate the amount of information (knowledge) a user needs to remember for an interaction (e.g. number of chunks in working memory). It is particularly important while working with novice or dyslexic users or new interfaces. In those cases, prediction about the amount of knowledge and learning required, effects of previous experience (positive / negative learning) are more important than performance simulation.

4.3 When to model

The final issue is rather philosophical – when should we use models. In other disciplines of science, modelling remains the only opportunity when it is impossible to take direct measurement of an event (like in astronomy or neurology). However, in HCI, a user trial is always theoretically possible and it will give us ‘real’ data in contrast to the prediction from a model. However 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 [13] has 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*". Additionally, user trials are not representative in certain cases, especially for designing inclusive interfaces for people with special needs. A good simulation with profound theoretical foundation can be more useful than a user trial in such cases. Exploratory use of modelling can also help designers to understand the problems and requirement of users, which may not always apparent through user trials or controlled experiments.

5 CONCLUSIONS

In this paper, we present a survey on the existing modelling techniques used in HCI. We have classified them into four categories and reviewed their applications. We have also discussed user modelling from a broader perspective. It should be evident that the use of modelling and the type of model to be used depend on many factors like the application, the designers, availability of time and cost for design etc. However, we hope this paper will give system analysts an understanding of different modelling paradigms, which in turn may help them to select the proper type of model for their applications.

REFERENCES

[1] Anderson J. R. and Lebiere C. "The Atomic Components of Thought." Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1998.

[2] Barnard P. "The Emotion Research Group Website, MRC Cognition and Brain Sciences Unit." Available at: http://www.mrc-cbu.cam.ac.uk/~philb , Accessed on 1st July, 2007

[3] Benyon D., Murray D., Applying User Modeling to Human Computer Interaction Design, Artificial Intelligence Review, Volume 7, Numbers 3-4 August 1993, pp. 199 – 225

[4] Biswas P. and Robinson P., A New Screen Scanning System based on Clustering Screen Objects, Journal of Assistive Technologies, Vol. 2 Issue 3 September 2008, ISSN: 1754-9450

[5] Biswas P. and Robinson P., Automatic Evaluation of Assistive Interfaces, ACM International Conference on Intelligent User Interfaces (IUI) 2008

[6] Biswas P. and Robinson P., Modelling Perception using Image Processing Algorithms, Pradipta Biswas, Peter Robinson, 23rd British Computer Society Conference on Human-Computer Interaction (HCI 09)

[7] Biswas P. and Robinson P., Predicting pointing time from hand strength, Usability & HCI for e-Inclusion, 5th Symposium of the Austrian Computer Society (USAB 2009), LNCS 5889

[8] Biswas P., Bhattacharyya S. and Samanta D. "User Model To Design Adaptable Interfaces For Motor-Impaired Users." Tencon '05 - IEEE Region 10 Conferences 2005. 1801-1844.

[9] Blandford A., Butterworthb R. and Curzonb P. "Models of interactive systems: a case study on programmable user modelling." International Journal of Human-Computer Studies 60 (2004): 149-200.

[10] Boden M. A., Computer Models of Mind: Computational Approaches in Theoretical Psychology, Cambridge University Press 1985

[11] Bovair S., Kieras D. E., and Polson P. G. "The acquisition and performance of text-editing skill: A cognitive complexity analysis." Human-Computer Interaction 5 (1990): 1-48.

[12] Butterworth R. and Blandford A. "Programmable user models: The story so far." Available at: http://www.cs.mdx.ac.uk/puma/wp8.pdf, Accessed on 30th June, 2007

[13] Buxton W., Human Input to Computer Systems: Theories, Techniques and Technology, Available at: http://www.billbuxton.com/inputManuscript.html, Accessed on 27th October, 2009

[14] Byrne M. D. "ACT-R/PM And Menu Selection: Applying A Cognitive Architecture To HCI." International Journal of Human Computer Studies 55 (2001): 41-84.

[15] Card S., Moran T. and Newell A. "The Psychology of Human-Computer Interaction." Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.

[16] Carroll J. M. and Olson J.M. "Mental Models In Human-Computer Interaction." Handbook of Human-Computer Interaction Ed. Helander M. Amsterdam, Netherlands: Elsevier Ltd., 1990. 135-158.

[17] "Cognitive Architectures." Available at: http://en.wikipedia.org/wiki/Cognitive_ architecture, Accessed on 1st July, 2007

[18] Duffy V. G. "Handbook of Digital Human Modeling: Research for Applied Ergonomics and Human Factors Engineering." Boca Raton, FL, USA: CRC Press, 2008.

[19] Edwards W. D. "Dynamic decision theory and probabilistic information processing." Human Factors 4.2 (1962): 59-73.

[20] Eng K., Lewis R. L., Tollinger I., Chu A., Howes A. and Vera A. "Generating Automated Predictions of Behavior Strategically Adapted To Specific Performance Objectives." ACM/SIGCHI Conference on Human Factors in Computing Systems 2006. 621-630.

[21] Fitts P.M. "The Information Capacity of The Human Motor System In Controlling The Amplitude of Movement." Journal of Experimental Psychology 47 (1954): 381-391.

[22] Gajos K. Z., Wobbrock J. O. and Weld D. S. "Automatically generating user interfaces adapted to users' motor and vision capabilities." ACM symposium on User interface software and technology 2007. 231-240.

[23] Gan K. C. and Hoffmann E. R. "Geometrical conditions for ballistic and visually controlled movements." Ergonomics 31 (1988): 829-839.

[24] Gray W. D. and Sabnani H. "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.

[25] Gray W., Young R.M. and Kirschenbaum S. "Introduction to this special issue on cognitive architectures and human-computer interaction." Human-Computer Interaction 12 (1997): 301-309.

[26] Griffiths T. L., Kemp C. and Tenenbaum J. B. "Bayesian Models of Inductive Learning." Tutorial at the Annual Meeting of the Cognitive Science Society 2008. 2665

[27] Gump A., Legare M. and Hunt D. L. "Application of Fitts' Law to individuals with cerebral palsy." Perceptual and Motor Skills 94 (2002): 883-895.

[28] Hampson P. J. and Moris P. E. "Understanding Cognition." Oxford, UK: Blackwell Publishers Ltd., 1996.

[29] Hick W.E. "On the rate of gain of information." Journal of Experimental Psychology 4 (1952): 11-26.

[30] Hornof A. J. and Kieras D. E. "Cognitive Modeling Reveals Menu Search Is Both Random And Systematic." ACM/SIGCHI Conference on Human Factors in Computing Systems 1997. 107-114.

[31] Horovitz, E. and colleagues, Microsoft Research, The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users, available at: http://research.microsoft.com/adapt//~horvitz/lumiere.htm

[32] Howes A., Vera A., Lewis R.L. and Mccurdy, M. "Cognitive Constraint Modeling: A Formal Approach To Reasoning About Behavior." Annual meeting of the Cognitive Science Society Lawrence Erlbaum Associates, 2004.

[33] John B. E. and Kieras D. "The GOMS Family of User Interface Analysis Techniques: Comparison And Contrast." ACM Transactions On Computer Human Interaction 3 (1996): 320-351.

[34] Johnson P. "Human Computer Interaction: psychology, task analysis and software engineering." Maidenhead.Maidenhead: Mcgraw Hill Book Company, 1992.

[35] Johnson-Laird P.A. "The Computer and The Mind." Cambridge, MA, USA: Harvard University Press, 1988.

[36] Keates S., Clarkson J. and Robinson P. "Investigating The Applicability of User Models For Motion Impaired Users." ACM/SIGACCESS Conference On Computers And Accessibility 2000. 129-136.

[37] Kieras D. E., Wood S. D., Abotel K. and Hornof A. "GLEAN: A Computer-Based Tool For Rapid GOMS Model Usability Evaluation of User Interface Designs." ACM Symposium On User Interface And Software Technology 1995. 91-100.
[38] Kieras D., GOMS Modeling of User Interfaces Using NGOMSL, Conference Companion On Human Factors In Computing Systems,Boston, Massachusetts, USA, pp.371-372, 1994
[39] Kieras D. and Meyer D. E. "An Overview of The EPIC Architecture For Cognition And Performance With Application to Human-Computer Interaction." Human-Computer Interaction 12 (1990): 391-438.
[40] Kieras D. E. "Fidelity Issues In Cognitive Architectures For HCI Modelling: Be Careful What You Wish For." International Conference On Human Computer Interaction 2005.
[41] Laird J.E., Rosenbloom P.S. and Newell A. "Towards chunking as a general learning mechanism." National Conference on Artificial Intelligence at Austin, TX: Morgan 1984. 188-192.
[42] Lallement Y. and Alexandre F. "Cognitive Aspects of Neurosymbolic Integration." Connectionist-Symbolic Integration Ed. Sun R. and Alexandre F. London, UK: Lawrence Erlbaum Associates, 1997.
[43] Li P. "Computational Modeling of Bilingualism." Workshop at the Annual. Meeting of the Cognitive Science Society 2006. 2659
[44] Mcmillan W. W. "Computing For Users with Special Needs And Models of Computer-Human Interaction." ACM/SIGCHI Conference on Human Factors In Computing Systems 1992. 143-148.
[45] Moran T.P. "Command Language Grammar: A Representation For The User Interface of Interactive Computer Systems." International Journal of Man-Machine Studies 15.1 (1981): 3-50.
[46] Motomura Y., Yoshida K., Fujimoto K., Generative user models for Adaptive Information Retrieval, Proc. IEEE Conf. on Systems, 2000
[47] Newell A. "You can’t play 20 questions with nature and win." Projective comments on the papers of this symposium. Pittsburgh, Carnegie Mellon University, Department of Computer Science. 1973.
[48] Newell A. and Simon H. A. "GPS, A Program That Simulates Human Thought." Cambridge, MA, USA: MIT Press, 1995.
[49] Newell A. "Unified Theories of Cognition." Cambridge, MA, USA: Harvard University Press, 1990.
[50] Norcio F. and Chen Q., Modeling User’s with Neural Architecture, Proc. Intl. Joint Conf. on Neural Networks, 1992. pp. 547-552.
[51] Norcio F., Adaptive Interfaces: Modelling Tasks and Users, IEEE Trans. Systems, Man, Cybernetics, 19(2) pp. 399-408, 1989.
[52] Oka N. "Hybrid cognitive model of conscious level processing and unconscious level processing." IEEE International Joint Conference on Neural Networks 1991. 485-490.
[53] Payne S.J. and Green T.R.G. "Task-Action Grammars: A Model of Mental Representation of Task Languages." Human-Computer Interaction 2 (1986): 93-133.
[54] Peck V. A. and John B. E. "Browser-Soar: a computational model of a highly interactive task." ACM/SIGCHI conference on Human factors in computing systems 1992. 165-172.
[55] Phillips N. "Graphical modification for partially sighted gamer accessibility." Computer Laboratory, Tripos Part II project University of Cambridge, 2009.
[56] Reisner P. "Formal Grammar And Human Factors Design of An Interactive Graphics System." IEEE Transactions On Software Engineering 7 (1981): 229-240.
[57] Rieman J. and Young R. M. "A dual-space model of iteratively deepening exploratory learning." International Journal of Human-Computer Studies 44 (1996): 743-775.
[58] Salvucci D. D. "An integrated model of eye movements and visual encoding." Cognitive Systems Research (2001):
[59] Salvucci D. D. and Lee F. J. "Simple cognitive Modelling in a complex cognitive architecture." ACM/SIGCHI Conference on Human Factors in Computing Systems 2003. 265-272.
[60] Scholtes V.A. B., Becher J. G., Beelen A. and Lankhorst G. J. "Clinical assessment of spasticity in children with cerebral palsy: a critical review of available instruments." Developmental Medicine and Child Neurology 48 (2006): 64-73.
[61] Stephanidis C., Paramythis A., Sfyrakis M., Stergiou A., Maou N., Leventis A.,Paparoulis G. and Karagiannidis C. "Adaptable And Adaptive User Interfaces for Disabled Users in the AVANTI Project." Intelligence In Services And Networks, LNCS-1430, Springer-Verlag 1998. 153-166.
[62] Stephanidis C. and Constantinou P. "Designing Human Computer Interfaces For Quadriplegic People." ACM Transactions On Computer-Human Interaction 10.2 (2003): 87-118.
[63] Strauss T. J. and Mirman D. and Magnuson J. S. "Speech Perception: Linking Computational Models and Human Data." Tutorial at the Annual Meeting of the Cognitive Science Society 2006. 2669

[64] Tollinger I., Lewis R. L., McCurdy M., Tollinger P., Vera A., Howes A. and Pelton L. "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 2005. 411-420.

[65] Wang H. C. "Modeling Idea Generation Sequences Using Hidden Markov Models." The Annual Meeting of the Cognitive Science Society 2008. 107-112.

[66] Young R.M., Green T.R.G. and Simon T. "Programmable User Models For Predictive Evaluation of Interface Designs." ACM/SIGCHI Conference on Human Factors in Computing Systems 1989. 15-19.