A Survey on User Modeling in HCI

Mohammad Sajib Al- Seraj
Department of Computer Science
Michigan Technological University
Houghton, MI 49931, USA
malseraj@mtu.edu

Abstract

In the last few years user modeling has become an important research area in Human Computer Interaction. A large amount of research has been conducted in this field where different approaches on user modeling are shown. In this paper, we provide an overview of the field of user modeling and describes the different user model namely, GOMS family of models, cognitive architecture, grammar based model, and application specific models. We have discussed a few examples of user models in each category. This paper also discusses the future challenges of this research area.

*Index Terms***—**User Model, Human Computer Interaction (HCI), GOMS model, Cognitive model, Grammar based model.

Introduction

From user experience and meticulous research, we find that computer systems are not easy to learn and once learnt, may be easily forgotten. Software industry updating their software in regularity basis with different interface functionality which sometimes creates difficulties for even for the learned users to cope with. Users changing their perception of and proficiency with different software systems which is another problem. The range of skills, knowledge and preferences required of users' means that any computer system which offers a fixed interface will be better suited to some users than to others.

Different types of user use the computer system in different way and their perspective is also different. An effective way of dealing with system complexity for the novice user is to provide a functionally simple system. Intermittent or discretionary computer users have to master different application packages as the need arises, and seldom have any real choice in the purchase, selection or use conditions of the software which their management decides upon. Discretionary users are a particularly important class as they often have to be encouraged to make use of a system and will generally benefit from easy to use, intuitive and functionally simple interfaces. Another familiar class of user is the committed or so called 'expert' user who may choose to employ different packages for different activities.

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 [Benyon & Murray, 1993]. It is not a mandatory part of the software but it helps to get the system serve the user better. Any information stored about the user or usage pattern is not a user model unless it can be used to get some explicit assumption about the user.

In the following section, the history of the user model will be presented. Different types of user model will be presented in the next section and the future challenge of the user model will be presented in the last section.

History of user modeling

User modeling is traced back to the works of Allen, Cohen and Perrault [Perrault et al., 1978; Cohen and Perrault, 1979] and Elaine Rich [Rich, 1979a, 1979b]. For a ten-year period following this seminal research, numerous application systems were developed that collected different types of information about, and exhibited different kinds of adaptation to, their current users. In these early modeling systems, the modeling components were not distinct from the rest of the application, but as the field grew user modeling systems were made more modular.
The Command Language Grammar [Moran, 1981] 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) [Card & Newell, 1983] was an important milestone in modeling HCI since it introduced the concept of simulating HCI from the perspective of users. It gave birth to the GOMS family of models [Card & Newell, 1983] that are still the most popular modeling tools in HCI.

Outside the domain of HCI, recent researches on cognitive modeling address a wide range of topics such as investigating mental processes for new idea generation. However, the domain of cognitive modeling 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 two main approaches of user modeling: 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 [Kieras, 2005]. 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. 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. 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' behavior simulator.

Different Models

- ***The GOMS family of models***

GOMS (Goals, Operators, Method and Selection) was in-spired by the GPS system developed by Newell. The method is composed of methods that are used to achieve specific goals. It 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 four different GOMS model.

1. KLM

Keystroke-Level Model (KLM) [Card & Newell, 1983] is the first and simplest GOMS technique by eliminating the goals, methods, and selection rules, leaving only six primitive operators. The KLM model [Card & Newell, 1983] simplifies the GOMS model by eliminating the goals, methods, and selection rules, leaving only six primitive operators. They are:

1) **K** keystroking/ keypressing
2) **P** pointing with a mouse to a target
3) **H** homing the hand on the keyboard
4) **M** performing mental preparation,
5) **D** drawing a line segment on a grid
6) **R** 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.

2. CMN-GOMS

CMN-GOMS is the original GOMS model proposed by Stuart Card, Thomas P. Moran and Allen Newell. It takes the KLM as its bases and adds sub goals and selection rules. This model can predict operator sequence as well as execution time. A CMN-GOMS model can be represented in pro-gram form, making it amenable to analysis as well as execution. CMN-GOMS has been used to model word processors and CAD systems for ergonomic design. The CMN method can predict the operator sequence and the execution time of a task on a quantitative level and can focus its attention on methods to accomplish goals on a qualitative level.

3. NGOMSL

A structured language representation of GOMS model, called NGOMSL (Natural GOMS Language) [Kieras, 1994] developed by Kieras. NGOMSL builds on CMN-GOMS by providing a natural-language notion for representing GOMS models, as well as a procedure for constructing the models. Under NGOMSL, methods are represented in terms of an underlying cognitive theory known as cognitive complexity theory, or CCT model at a higher level of notation. CCT is a rule-based system developed by Bovair and colleagues to model the knowledge of users of an interactive computer system. Kieras also developed a modeling tool, GLEAN (GOMS Language Evaluation and Analysis), to execute NGOMSL. It simulates the interaction between a simulated users with a simulated device for undertaking a task.

4. CPMGOMS

John and Kieras [John & Kieras, 1996] proposed a new version of the GOMS model, called CPMGOMS, 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.

- ***Cognitive Architectures***

Allen Newell developed the Soar architecture [Newell, 1990] as a possible candidate for his unified theories of cognition. According to Newell [Newell, 1990] and Johnson-Laird [Johnson-Laird, 1988], 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)" [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.

However, there are certain aspects of human cognition that can be better explained by a connectionist approach than a symbolic one. 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. Lallement and Alexandre 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 modeling 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 does not follow the pure symbolic modeling 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 behavior 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) 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.

The CORE system (Constraint-based Optimizing Reaso ing Engine) [Lewis et al., 2006; Howes et al., 2004] 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, but they are not yet as extensively used as the previously discussed systems.

- ***Grammar-based models***

The grammar based model simulates an interaction in the form of grammatical rules. As for example, Task Action Language models

i. Operations by Terminal symbols
ii. Interaction by a Set of rules
iii. 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.

- ***Application Specific Models***

A lot of works has been done on user modeling for developing customizable applications. These models have the following generic structure. 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 [Motomura, 2000] 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 [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 users. Norcio and Chen also used an artificial neural network for the same purpose as their previous work [Norcio, 1989]. 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 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.

Future Challenges

- ***Support different modeling techniques***

Most of the user modeling approaches failed because of relying on one specific technique. A substantial leverage can be gained by integrating modeling with implicit modeling. This enriched synthesis can be further complemented by asking users (in otherwise open environments, such as HFAs and design environments) to solve specific problems in which the selection of the problem is driven by specific needs of the user modeling component.

- ***Payoff of user modeling***

There is little to be gained if expensive mechanisms against minimal improvements in usability and usefulness. The payoff or utility of cognitive artifacts can be characterized by the quotient of "value / effort." To increase the payoff, we have two options: (1) increase the value by showing that future systems relying on user models are more usable and more useful, or (2) decrease the effort associated with creating a user model.

- ***Wrong, outdated, and inadequate information***

User models represent a world that is outside the computational environment. The mapping of external information to the internal model may be wrong to start with, but even under the assumption that it is an adequate representation at some point of time, it may become outdated by external changes of which the model is unaware [Allen, 1997]. How, when, and by whom can a wrong user model be identified? Who will have the authority and the knowledge to change the model, and which modification mechanisms will be available to do so?

- ***Criteria for different domains***

Assuming that user modeling is useful in some domains but not in others, which criteria do we have to distinguish these domains?

- ***Control***

A consequence of any smart behavior of systems is that agents (humans or computers) can guess wrong and per-form hidden changes that users do not like. Current systems often lack the possibility or at least the transparency for users to turn off these "smart" features, which can get more in the way than help. As argued above, systems, even smart ones, are aware of only a fraction of the total problem solving process their human partners undergo [Hollan, 1990], and they cannot share an understanding of the situation or state of problem-solving of a human. Whereas these drawbacks of smart systems may be only annoying in HFAs such as word processors, they are unacceptable in other collaborative human-computer systems, such as airline cockpit computers serving as intelligent agents. Billings [Billings, 1991] argues convincingly that in computerized cockpit design each intelligent agent in a human-computer system must have knowledge of the intent and the rationale of the actions of the other agents. To avoid these drawbacks, intelligent systems should pro-vide malleable tools that empower rather than diminish users, giving them control over tasks necessary for everyday life. There are situations in which we desire automation and intelligence (for example, few people will have the desire to compile their programs themselves) — but the decision as to what should be automated and what not should be under the control of the people affected by the system.

- ***Privacy***

Security is a major concern in this computational era. Numerous organizations compile user models of our behavior and actions — and there is the great danger that this in-formation can be misused. It will be a major challenge to find ways to avoid misuse, either by not allowing companies to collect this information at all or by finding ways that the individual users have control over these user models.

Conclusion

In this paper, different types of user models has been elaborately discussed. We have classified user model into four categories. We have also analyzed user modeling from a broader perspective. 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 etc. However, we hope this paper 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

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

Perrault C. R., Allen J. F., & Cohen P. R. (1978). *Speech Acts As a Basis for Understanding Dialogue Coherence*, In Proceedings of the 1978 Workshop on Theoretical Issues in Natural Language Processing (pp. 125–132). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/980262.980282.

Cohen P. R., & Perrault C. R. (1979). *Elements of a plan-based theory of speech acts*, Cognitive Science, 3(3), 177–212. doi:10.1016/S0364-0213(79)80006-3

Rich, E. (1979a). Building and exploiting user models. In Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2, pp. 720–722.

Rich, E. (1979b). *User modeling via stereotypes**. Cognitive Science, 3(4), 329–354.

Moran T.P. (1981) *Command Language Grammar: A Representation For The User Interface of Interactive Computer Systems*, International Journal of Man-Machine Studies 15.1, pp. 3-50.

Card S., Moran T. and Newell A.(1983), *the Psychology of Human-Computer Interaction*, Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

Kieras D. E.(2005), *Fidelity Issues In Cognitive Architectures For HCI Modelling: Be Careful What You Wish For*, International Conference On Human Computer Interaction.

Kieras D.(1994), *GOMS Modeling of User Interfaces Using NGOMSL*, Conference Companion On Human Factors In Computing Systems,Boston, Massachusetts, USA, pp.371-372.

Newell A.(1990), *Unified Theories of Cognition*, Cambridge, MA, USA: Harvard University Press.

Johnson-Laird P.A.(1988), *The Computer and The Mind*, Cambridge, MA, USA: Harvard University Press.

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, pp. 621-630

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

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

Motomura Y., Yoshida K., Fujimoto K.(200), *Generative user models for Adaptive Information Retrieval*, Proc. IEEE Conf. on Systems.

Norcio F.(1989), Adaptive Interfaces: Modelling Tasks and Users, IEEE Trans. Systems, Man, Cybernetics, 19(2) pp. 399-408.

Allen, R. B. (1997), *Mental Models and User Models* In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), Handbook of Human-Computer Interaction, Volume 1, Elsevier Science B.V., Amsterdam, pp. 49-63.

Hollan, J. D. (1990), *User Models and User Interfaces: A Case for Domain Models, Task Models, and Tailorability*, In Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence, AAAI Press/The MIT Press, Cambridge, MA, pp. 1137.

Billings, C. E. (1991) *Human-Centered Aircraft Automation: A Concept and Guidelines,* NASA Technical Memorandum, Report No. 103885, NASA Ames Research Center.