

Simplifying the Development and the Analysis of Cognitive Models

Marcus Heinath, Jeronimo Dzaack, Andre Wiesner ({mhe, jdz, awi}@zmms.tu-berlin.de)

Center of Human-Machine-Systems - Technische Universität Berlin
Franklinstraße 28-29, FR 2-7/2, 10587 Berlin, Germany

Leon Urbas (leon.urbas@tu-dresden.de)

Institute of Automation - Technische Universität Dresden
01062 Dresden, Germany

Abstract

Usability of complex dynamic human computer interfaces can be evaluated by cognitive modeling to investigate cognitive processes and their underlying structures. Even though the prediction of human behavior can help to detect errors in the interaction design and cognitive demands of the future user the method is not widely applied. The time-consuming transformation of a problem “in the world” into a “computational model” and the lack of fine-grained analysis of simulation data are mainly responsible for this. Having realized these drawbacks we developed HTAmap and SimTrA to simplify the development and analysis of cognitive models. HTAmap, a high-level framework for cognitive modeling, aims to reduce the modeling effort. Within HTAmap the process of building cognitive models is transformed into a pattern-oriented task, based on “cognitive activity patterns”. SimTrA supports the analysis of cognitive model data on an overall and microstructure level and enables the user to automatically compare simulated data with empirical data. This paper describes both concepts and first implementations. The practicability of both tools is shown using an example in the domain of process control.

Introduction

Recent introductions of new information technologies in the range of dynamic human-machine systems (e.g. process control systems in the chemical industry or airplane cockpits) have led to increasing cognitive requirements caused by a shift from manual processes of operation to the management of complex automated processes. This leads to user interfaces which are characterized by a high complexity and a high degree of dynamics. Because of the integrated functionality and the complex data structures, these interfaces require more cognitive information processing. The aim is to design systems which support the cognitive demands of users. Cognitive modeling seems to be a good candidate for this purpose because it allows to gain insight into cognitive aspects of human behavior in a more specific way than empirical or heuristic methods. But despite the promising potential, this method is still rarely used in industrial research departments. The main obstacles are time and cost efforts for building and analyzing cognitive models, caused by a lack of support tools and by sophisticated knowledge in both cognitive psychology as well as artificial intelligence programming (Heffernan, Koedinger & Alevan, 2003). Having realized this drawback we present *HTAmap* (Hierarchical Task Mapper) and

SimTrA (Simulation Trace Analyzer), two new approaches to simplify the development and analysis of cognitive models and thereby reducing costs and time. HTAmap provides two key features: firstly, it uses a plain high-level description based on appropriate task analysis methods. Secondly, it supports the reuse of cognitive model components based on cognitive activity pattern. SimTrA provides applications to extract and process cognitive model data on an overall and microstructure level. It allows to analyze cognitive model data and to compare the data with empirical data afterwards. Both are part of a series of software tools within an integrated modeling environment for cognitive models.

Potentials and Constraints of Cognitive Modeling as Method for System Evaluation

Cognitive architectures incorporate psychological theories (e.g. visual information processing, decision making, motor commands) and empirically based representations about aspects of human cognition. There is general agreement that these cognitive aspects are relatively constant over time and relatively task-independent (Howes & Young, 1997). Therefore, cognitive architectures present these aspects in a software framework to explain and predict human behavior in a detailed manner. In this context, a cognitive model can be seen as an application of a cognitive architecture to a specific problem domain with a particular knowledge set. Building a cognitive model, the modeler must describe cognitive mechanisms in a highly-detailed and human-like way. Two levels of cognitive architectures can be differentiated: High-level architectures (e.g. GOMS, CTT, HTA, see Limbourg & Vanderdonckt, 2004 for an overview) describe behavior on a basic level and define interactions as a static sequence of human actions. Low-level architectures (e.g. ACT-R, SOAR or EPIC, see Byrne, 2003 for an overview) describe human behavior on an atomic level. They allow a more detailed insight into cognitive processes than high-level architectures. Most low-level architectures use production systems to simulate human processing and cognition. The use of independent production rules allows cognitive models to react on external stimuli (bottom-up processes) and to model interruption and resumption of cognitive processes in contrast to high-level architectures which are usually controlled top-down. The research presented in this paper uses the cognitive architecture ACT-R (Anderson et al., 2004).

Determining Factors for Practical Application

In a practical application cognitive models can be used to evaluate the usability of prototypes. This helps to detect errors in the interaction design of interfaces and gives indications about the cognitive demands of the future user. Cognitive models extend classical usability methods and expand the repertoire by cognitive aspects. However, this method is seldom employed in usability research and development because of a lack of support tools for creating and analyzing cognitive models.

Development Effort for Cognitive Models

Various authors (Heffernan, Koedinger & Alevan, 2003; Crossman et al., 2004; Tollinger et al., 2005) analyzed the cognitive modeling process in detail, together with the necessary subtasks and requirements. Transformation of task knowledge into the computational description of the cognitive architecture is challenging and requires extensive programming experience. The resulting high cost/benefit ratio is an important constraint on the practical application. The real bottleneck lies in the transformation-gap after the preliminary task analysis process (see Figure 1) and is caused by the different levels of task decomposition and formalization. Task analysis methods formalize knowledge about cognitive processes with a greater degree of abstraction and less formalization compared to low-level modeling approaches such as ACT-R. For example, “read button” which is atomic at the level of task analysis corresponds to a complex sequence of production rules in ACT-R (e.g. retrieve-position, find-position, attend-position, read and store results) and a set of data elements for flow control and internal representation. It is up to the cognitive modeler to fill this gap between the high-level task description and the low-level implementation by the means of a cognitive architecture. A current topic in the cognitive research community is to minimize this gap with the development of high-level languages to model human cognition based on low-level cognitive architectures (for an overview of current approaches see Ritter et al., 2006). The main objectives are to simplify the model-building process and to improve concepts for sharing and reusing model components (Crossman et al., 2004).

Analyzing Effort for Simulation Data

Most low-level simulation experiments use global information to analyze the model and its fit to empirical data (e.g. errors or times). A problem is connected with this procedure. In order to validate that a current cognitive model acts like a human, not only the behavior of a cognitive model and the human have to be highly comparable, but also the kind of computations and sub-processes to achieve the results have to be equivalent. With the psychological theories implemented in low-level cognitive architectures a more detailed analysis of cognitive model data is possible to enrich the explanatory power of cognitive models. This makes it possible to analyze the computations that lead to the cognitive model behavior and thereby to determine the level of correctness of the implemented cognitive model. For this purpose fine-grained patterns can be detected in the simulation data to enrich the

explanatory power of cognitive models (e.g. sequence of actions). For example, the arrangement and the appearance of elements of an interface can be evaluated with respect to theories of eye-movement or signal detection. Nevertheless using cognitive models reveals some problems. Cognitive architectures and models are incomplete and describe only a small part of the processes that make up human cognition. Reasons for this are the partial knowledge of internal cognitive processes in cognitive science and the insufficient implementation of all cognitive aspects that are needed to handle a task. Aspects like esthetics, boredom, fun or personal preferences which can be observed in empirical settings are not implemented (Byrne, 2003). When analyzing and comparing cognitive model data with empirical data, these differences have to be taken into account. So far, no tools exist for the extraction of fine-grained information from model data for the evaluation of user interfaces.

Integrated Environment for Cognitive Modeling and Analysis

To reduce the effort for developing cognitive models in ACT-R and to support the analysis of simulation and empirical data in a systematic way, *HTAmap* (Hierarchical Task Mapper) and *SimTrA* (Simulation Trace Analyzer) were developed. The placement of HTAmap and SimTrA within the general cognitive modeling process (consisting of the four steps: task analysis, empirical data collection, model implementation and validation) is shown in Figure 1. On the following pages the concepts, paradigms and first implementations of HTAmap and SimTrA are described in detail.

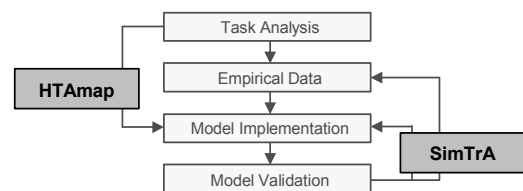


Figure 1: Placement of HTAmap and SimTrA within the general cognitive modeling process.

HTAmap

Building cognitive models is not easy and involves a strong process of synthesis, i.e. building a new solution by putting parts together in a logical way. For developing cognitive models in ACT-R this implies programming in a “cognitive assembly language”. Behavior is expressed in terms of production-rules that manipulate knowledge expressed in declarative memory elements. To open cognitive modeling for a wider user group and make the developing task easier and more accessible, HTAmap addresses (1) a structured formalization method to minimize the “transformation-gap” between high-level task analysis and low-level approaches of cognitive modeling, and (2) programming paradigms with more immediate results. These can be achieved by model reuse and domain-oriented paradigms.

Pattern-oriented Cognitive Modeling

A number of task analysis methods have been developed for different purposes (see Limbourg & Vanderdonck, 2004; for an overview), not many of them being a good starting point for cognitive modeling at all. To be suitable as base for a pattern oriented model engineering approach a task analysis method needs at least a minimal formal structure, clear rules for reuse and an appropriate level of detail. This is true for Ormerod & Shepherd's (2004) "sub-goal template" (SGT) method. The SGT method extends the "hierarchical task analysis" (HTA) method (Shepherd, 2001) by providing a nomenclature for stereotypical operator tasks and defining four essential steps (see Figure 2): the (1) initial task/subtask decomposition to the point where (2) "information-handling operations" (IHO) are recognized, followed by the (3) strategic decomposition and the (4) redescription in terms of sub-goal templates. Information handling operations are divided into three classes: receiving (IHO_R), evaluating (IHO_E) and acting on information (IHO_A). The identified IHOs are redescrbed as operator tasks regarding one of the four sub-goal templates: (A)ct, (E)xchange, (N)avigate and (M)onitor. In addition, the SGT method defines a plan in which IHOs are sequenced relative to each other (i.e. fixed, free, parallel or contingent sequence). It specifies information requirements needed by an operator to carry out tasks during operation of a technical system.

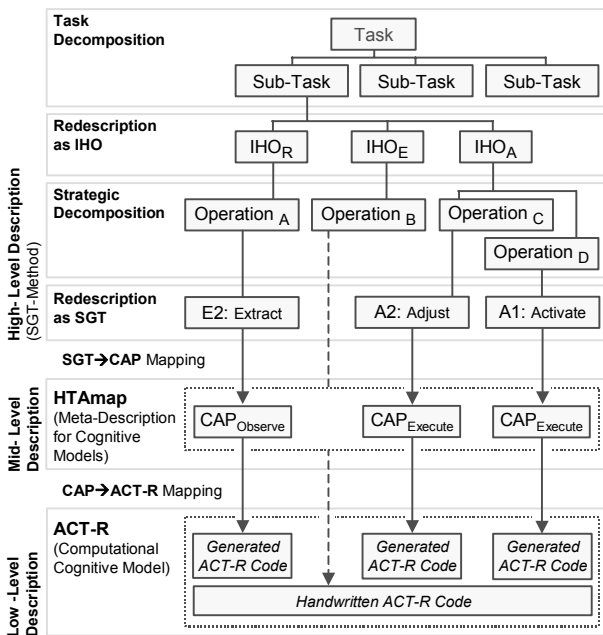


Figure 2: The mapping process from SGT to HTAmap and ACT-R code/models.

The redescription to the level of SGTs is the starting point for the HTAmap approach (see Figure 2). "Cognitive activity patterns" (CAP) are used to solve the high to low level mapping-problem. Within HTAmap, CAPs add a layer between a higher-level behavior specification in terms of SGTs and the lower-level behavior specification in terms of

the target architecture. Compound CAPs (cCAP) are composed by setting relations between elementary CAPs (eCAP). In general, the presented CAP approach allows the transition of IHOs into the less abstract level of the target architecture. In detail, a CAP comprises the necessary templates to generate declarative and procedural structures. While the general structure of the templates may be suitable for an arbitrary production system, the internals are highly specific for the target architecture (in our case ACT-R), because they need to be parameterized towards various control strategies, task environments and tasks. Table 1 shows a couple of implemented CAPs.

Table 1: Selection of CAPs and their definition.

CAP	Definition (after Hollnagel, 1998)
scan	Quick review of displays [...] to obtain a general but preferably complete impression of the state of a system or sub-system.
observe	Look for or read one or more specific measurement values or system indications.
monitor	Follow the development of one or of a set of specific parameters [...] over time.
execute	Perform a previously specified action.

Formalization and Implementation of the Cognitive Activity Pattern and the HTAmap-Model

The CAPs are implemented using a specific schema based on the "Extensible Markup Language" (XML) standard. The notation specifies semantic information about the pattern, descriptions of functionality in terms of ACT-R primitives, the relations to other CAPs and a structured documentation. The CAP's implementation concept provides reusable and task independent components, i.e. generic cognitive behavior blocks in the form of associated production rules and the specification of domain dependent components within one structure. The latter offers possibilities to parameterize the CAPs regarding particular task situations.

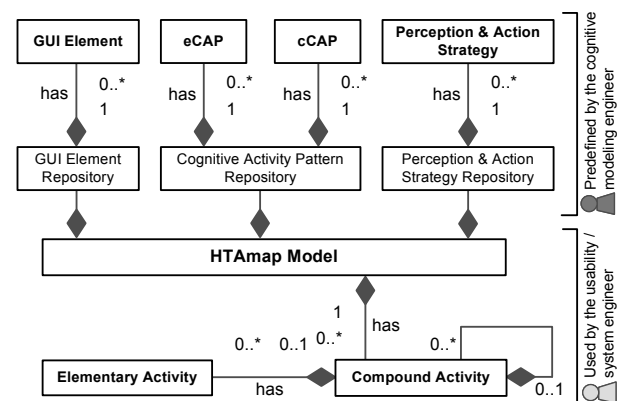


Figure 3: UML-diagram of the components of the HTAmap-model.

The HTAmap-model represents the meta-description of an ACT-R model concerning its associated high-level task model defined by elementary and compound cognitive

activity patterns (eCAP/cCAP), a description of the used tasks interface elements (GUI element) and required strategies that handle the perception and action of models (see Figure 3). To build a cognitive model within HTAmap, the modeler selects one of the predefined CAPs stored in the CAP repository. In addition, compound CAPs are composed of their associated elementary CAPs and their relations to each other (e.g., sequencing information). Afterwards, the CAPs are parameterized on the basis of the predefined GUI elements and their associated strategies using the AGImap framework (Urbas & Leuchter, 2005). The high-level task is now specified as HTAmap-model in terms of the low-level cognitive architecture ACT-R relating to a specific task environment. The resulting HTAmap-model is transformed into specific ACT-R constructs and is executable within the cognitive framework.

SimTrA

To analyze cognitive model data for the evaluation of human computer interaction the simulation data has to be preprocessed and provided conveniently. For this reason the *Simulation Trace Analyzer* (SimTrA) was developed. SimTrA is independent of the cognitive model and allows to process simulated and empirical data. For the analysis the data is transferred into a general-purpose format for complex, hierarchically structured data (XML). This forms the basis for a general algorithm-based analysis of the interaction processes. Two levels of abstraction have to be observed in order to analyze the model's performance on the basis of the integrated psychological theories (e.g. visual perception and processing): the global structure and the microstructure level to identify the underlying processes (Gray, 2006). For the global structure, aspects of the model's overall performance are analyzed (e.g. times, errors, and transition-matrices of areas of interest). The microstructure can be characterized by the sub-processes of the cognitive model, i.e. repeated short sequences of action such as control-loops, or scanpaths. For the automated simulation data analysis, algorithms are implemented and integrated into SimTrA. The results are plotted as in classical usability-evaluation methods.

Analyzing Cognitive Model Data

The important role of eye-movement studies in psychological and cognitive research shows that eye movement data is able to give an insight into human behavior and its underlying cognitive processes (Just & Carpenter, 1984; Rayner, 1990; Rötting, 2001). Cognitive models with ACT-R can process visual information, providing spatial and temporal information of the simulated eye movement as in empirical studies. The extraction of this information provides a way to analyze and compare cognitive model data with empirical data on an global (e.g., number of fixations) and microstructure level (e.g., scanpaths). That is the reason for the implementation of SimTrA for eye-movement data (empirical and simulated). The implemented process is divided into three components: the (1) preprocessing of the raw data, the (2) analysis of the preprocessed data and the (3) comparison with further models or empirical data (see Figure 4). This allows the

independent development of each module and an easy extension of its functionality in the future. SimTrA enables the user to apply basal applications regarding the process of analyzing and comparing. After each step the processed data is stored in a general-purpose format and can be processed by external tools (e.g. MatLab, R, SPSS). It is possible to import empirical data into SimTrA for comparison with cognitive model data.

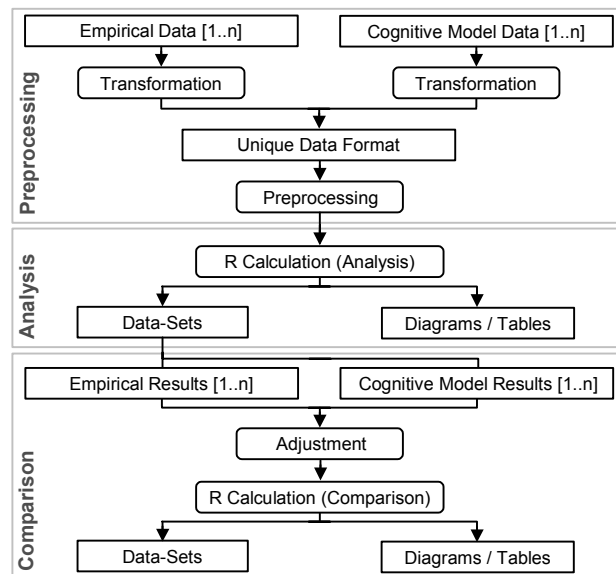


Figure 4: Schematic overview of the implementation concept of SimTrA.

Preprocessing

For the preprocessing of the raw data the empirical data and the cognitive model data are transferred into a general-purpose format (see Figure 4). The transformation allows processing of the data with different quality and origin (e.g. different cognitive architectures, empirical study) with the same algorithms afterwards. While preprocessing each dataset, the stored data is augmented by information on the origin of the data (e.g. cognitive model or empirical data), the subject and the trial identification (i.e. data-Id, subject-Id and trial-Id). In this component, the data of different origins can be scaled to a similar resolution by the user to ensure that the data is comparable. Each kind of information can be altered by the user in the preprocessing component of SimTrA. In this step the data is scaled to a similar resolution to ensure that it is comparable. After the transformation, the data is enriched by additional information that is needed for the analysis. The preprocessing of the raw data is finished by choosing the desired analysis methods.

Analysis

This component enables the analysis of the preprocessed data (see Figure 4). The algorithms for the analysis are implemented in R, a free tool for statistical computing. The results are saved in tables and as graphical plots. In this step algorithms are implemented to analyze the transition frequencies, the fixations on AOIs, the spatial density, the

statistical dependency in visual scanning, the local scanpaths and a mean for each of these algorithms (for an overview see Rötting, 2001). In the result tables the data is distinguished by the data-Id, subject-Id and the trial-Id. The absolute and the cumulative number of fixations, fixation durations and general information are stored in a general table. Each algorithm accesses the general table and processes the information needed for the calculation of the respective parameter. To support the user, plots are generated for each analysis. All input-files (i.e. empirical and cognitive model data), results, general information and the distribution of eye movement data are stored in either tables or plots.

Comparison

This component enables the comparison between the analyzed data from cognitive models or empirical studies (see Figure 4). This allows the rating of the simulation experiment and the simulated behavior with respect to empirical findings and psychological theories. This last step enables the iterative adjustment of cognitive models (see Figure 1). Therefore the analyzed data is revised by the user in the provided interface (e.g. missing data, insufficient data points) and compared by algorithms implemented in the software R. To this point of the development of SimTrA the comparison is done by descriptive methods (e.g. mean, standard error and standard deviation) because today's cognitive model data do not have a high variance and statistical inference methods are not applicable.

Practical Application: Process Control System

The use of HTAmap and SimTrA can be illustrated by considering a user's interaction with a complex dynamic interface of a process control system (see Figure 5). The aim is to stabilize the level of liquid in a container which is moderated by inflow, outflow and evaporation. The level can be regulated by adjusting a valve and a heater. For the model based investigation we developed an ACT-R 6.0 model representing the internal knowledge of an experienced user who is familiar with the technical process (as found in a survey of experts). It also describes the graphical user interface (GUI) in terms of AGImap (Urbas & Leuchter, 2005), an interaction abstraction layer above the low-level ACT-R graphical interface (AGI).

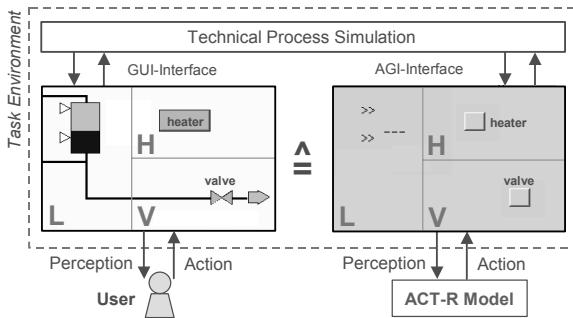


Figure 5: Schematic representation of the empirical study and the simulation experiment with the areas of interest (AOIs) for the human and the models interface.

Pattern Based Modeling of the Task

In dynamic human-machine systems, the human-machine dialogue is normally based on different levels of manual and supervisory control actions. Using the SGT method the regulation task is broken down into the IHOs: get a general idea of the system state (IHO_R), evaluate the system state (especially the level of the container) (IHO_E) and, as necessary, make adjustments to control the level (IHO_A). The identified IHO_R is mapped within the HTAmap-model to the compound CAP "scan" (see Table 1). The compound CAP "scan" environment is composed of a sequence of elementary "observe" CAPs, executed in turn (see Figure 6 left). Required task information for the "observe" CAPs are the interface elements and their values. Within the HTAmap-model the predefined "observe" CAPs are parameterized in the task environment by choosing the relevant GUI elements (i.e. valve, heater and level) and their associated perception strategies (i.e. read-button and read-level) from the particular repositories (see Figure 3). With the compound CAP "scan" two types of sequencing the "observe" CAPs have been implemented: a predefined order (fixed sequence: state-based/top-down control) and a non-defined order (free sequence: reactive/bottom-up control).

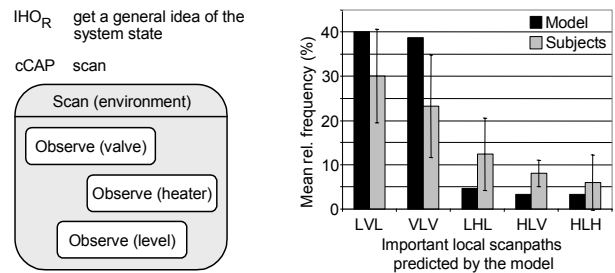


Figure 6: Schematic representation of the cCAP "scan" (right) and relative frequencies of the 5 important scanpaths for model and subject with one standard deviation (left).

Analysis of the Simulation Trace

After simulating the user behavior with the constructed model the simulation data was analyzed by the tool SimTrA. The aim of this analysis was to show that SimTrA is able to provide data that helps to classify the model and its fit to empirical data on a microstructure level. To analyze the microstructure behavior, the local scanpaths (consistent patterns of consecutive fixations; Groner et al., 1984) of actual perceptions were analyzed. They represent stimuli-driven bottom-up processes. Therefore the stimuli were divided in suitable areas of interest (AOI) in the first step (preprocessing) - heater: H, valve: V and level: L (see Figure 5). All theoretical triples of AOIs were determined by the algorithm in the second step (analysis) whereas any sequence of fixations falling into the same AOI is treated as a single gaze fixation and the number of occurrences in the whole sequence of AOIs is assigned to them (e.g. AOIs: 1, 2 - Sequence: 2122112 - triple: 121: 1, 212: 2). Ordering these by frequency shows the most important local scanpaths. Comparing the results with empirical data of a previous experiment in the last step

(comparison) reveals that the important local scanpaths in model and empirical data are almost congruent (Dzaack & Urbas, 2006). Therefore local scanpaths with a frequency of 3% were excluded and the remaining five important scanpaths that cover 90% of the model data were used for the comparison human – model (see Figure 6 right). These scanpaths are found to be important for the subjects as well (all five scanpaths cover 80% of the subject's data) and the rank order of the first 3 empirical and predicted scanpaths is the same. The mean relative frequencies of 3 model's scanpaths are within one standard deviation of the subject's data. The mean deviation between the frequency of all scanpaths in the model and human data is about 4%. The comparative analysis of cognitive model data and the empirical data with SimTrA shows that SimTrA is able to process cognitive model data.

Outlook

We designed HTAmap, a pattern-oriented approach for high-level description of cognitive models. Within this approach the cognitive activity patterns (CAP) are the central elements that specify a generic solution for a stereotypical operator task on the description level of ACT-R. Currently, only a selection of CAPs is specified. To transfer more "associated" production rules into CAPs, further work on verification and validation is required. An editor for building HTAmap-models is being implemented and an usability evaluation will be conducted. Further work is needed on the extension of the analysis tool SimTrA. It has to be evaluated which additional usability related analysis algorithms can be integrated for this purpose. Subsequently a second experiment is planned where different interface designs for the task described above are tested with humans and with a cognitive model. We believe that building cognitive models with the help of HTAmap makes the modeling process more accessible for a wider user group, simplifies the reuse of model fragments and improves the model communication. SimTrA enables the comparison of cognitive model behavior with human behavior on a global and a microstructure level. Both together could lead to an increased application of cognitive models in the usability evaluation.

Acknowledgments

This project has been funded by Deutsche Forschungsgemeinschaft (Research Training Group 1013 prometei) and VolkswagenStiftung.

References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*, 1036-1060.
- Byrne, M. D. (2003). Cognitive Architectures. In J. Jacko, & A. Sears (Eds.), *Handbook of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Crossman, J., Wray, R. E., Jones, R. M., & Lebiere C. A. (2004) High Level Symbolic Representation for Behaviour Modeling. *Proc. Conference on Behavior Representation*. Arlington, VA.
- Limbourg, Q., & Vanderdonckt, J. (2004). Comparing Task Models for User Interface Design. In D. Diaper & N. Stanton (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction*. NJ: LEA.
- Dzaack, J., & Urbas, L. (2006). Kognitive Modelle zur Evaluation der Gebrauchstauglichkeit von Mensch-Maschine Systemen. In M. Grandt & A. Bauch (Eds.), *Cognitive Systems Engineering in der Fahrzeug- und Prozessführung*. Bonn: DGLR.
- Ellis, S.R., & Smith, J.D. (1985). Patterns of Statistical Dependency in Visual Scanning. In R. Groner, G.W. McConkie & C. Menz (Eds.), *Eye Movement and Human Information Processing*. Amsterdam: Elsevier.
- Gray, W. D., Sims, C. R., & Schoelles, M. J. (2006). Musings on Models of Integrated Cognition: What Are They? What Do They Tell Us that Simpler Approaches Cannot? How Do We Evaluate Them? *Proc. of the 7th ICCM* (pp. 12-13). NJ: LEA.
- Groner, R., Walder, F., & Groner, M. (1984). Looking at faces: local and global aspects of scanpaths. In A. G. Gale & F. Johnson (Eds.), *Theoretical and Applied Aspects of Eye Movement Research*. North-Holland: Elsevier.
- Heffernan, N. T., Koedinger, K. R., & Alevan, V. A. (2003). Tools Towards Reducing the Costs of Designing, Building, and Testing Cognitive Models. *Proc. BRIMS*.
- Hollnagel, E. (1998). *Cognitive Reliability and Error Analysis Method (CREAM)*. Oxford: Elsevier.
- Howes, A., & Young, R. M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human-Computer Interaction* *12*, 4, 311-343.
- Just, M.A., & Carpenter, P.A. (1984). Using eye fixations to study reading comprehension: Individual differences in working memory. *Psychological Review* *99*, 122-149.
- Ormerod, T. C., & Shepherd, A. (2004). Using Task Analysis for Information Requirements Specification: The Sub-Goal Template (SGT) Method. In D. Diaper, & N., A. Stanton, (Eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: LEA.
- Rayner, K. (1999). Do eye movements reflect higher order processes in reading? In R. Groner, G. d'Ydewalle & R. Parham (Eds.), *From Eye to Mind: Information Acquisition in Perception, Search, and Reading*. NY: Elsevier.
- Ritter, F. R., Haynes, S. R., Cohen, M., Howes, A., John, B., Best, B., et al. (2006). High-level Behavior Representation Languages Revisited. *Proc. ICCM '06* (pp. 404-407). Edizioni Goliardiche.
- Rötting, M. (2001): *Parametersystematik der Augen- und Blickbewegungen für arbeitswissenschaftliche Untersuchungen*. Shaker, Aachen
- Shepherd, A. (2001). *Hierarchical task analysis*. New York: Taylor & friends.
- Tollinger, I., Lewis, R. L., McCurdy, M., Tollinger, P., Vera, A., Howes, A., et al. (2005). Supporting efficient development of cognitive models at multiple skill levels: exploring recent advances in constraint-based modeling. *Proc. CHI '05* (pp. 411-420). ACM Press.
- Urbas, L., & Leuchter, S. (2005). Model Based Analysis and Design of Human-Machine Dialogues through Displays. *KI – Künstliche Intelligenz*, *4*, 45-51.