High-Level ACT-R Modeling based on SGT Task Models

Marcus Heinath (mhe@zmms.tu-berlin.de)
Berlin University of Technology, Center of Human-Machine Systems,
Franklinstraße 28-29, FR 2-7/2, D-10587 Berlin

Introduction

Today's most important constraint on practical application of cognitive modeling as evaluation method is the high cost/low benefit ratio caused by a lack of support tools for modeling and by high requirements on sophisticated knowledge in cognitive psychology as well as artificial intelligence programming. The development of high-level languages to model human cognition based on low-level cognitive architecture is a current matter in the cognitive research community (see Ritter et al., 2006).

To open cognitive modeling to a wider user group and make the developing task easier and more accessible, the _Hierarchical Task Analysis Mapper_ approach (HTAmap) has been developed. The main idea behind HTAmap is modeling on a higher level of abstraction. The foundation of the HTAmap approach is determined by:

- _HTAmap modeling process_: A formalized modeling process to minimize the transformation-gap between semi-formal, high-level sub-goal template task models (SGT; Ormerod & Shepherd, 2004) and formal, low-level ACT-R models (Anderson et al., 2004).
- _HTAmap modeling language_: An XML-based representation of task- and device-related knowledge within an integrated ACT-R model as well as strategies for model reuse and adaptation with regard to dynamic task environments and user skills.
- _HTAmap editor_: A Java-based software tool for systematic and semi-automated ACT-R modeling based on predefined model fragments.

The HTAmap modeling approach delivers cognitive modeling functionality based on predefined and modifiable _cognitive activity patterns_ (CAP). Options for integrating separately defined device models of complex dynamic task environments (ACT-R graphical interface, AGI) as well as interfaces for embedded perception and action-models (AGI strategies) are also possible.

HTAmap modeling process

A large part of the cognitive model building process is transformed into a more simple pattern-oriented modification task. Figure 1 shows an overview of the different modeling steps. The output of the SGT method is the formal decomposition of the task and its re-description in terms of sub-goal templates, which represent a nomenclature for stereotypical operator tasks (TM_{SGT}). This re-description is the starting point for building the XML based HTAmap model (TM_{HTAmap}) and its automated transformation into ACT-R code (TM_{ACT-R}). In comparison to existing macro compilation approaches like ACT-Simple, G2A or ACT-Stitch (see Ritter et al., 2006 for an overview), that are widely limited to model user behavior in static task environments, HTAmap focuses strongly on simulations of dynamic task environments and their simplified coupling with ACT-R ($IM_{AGI} \rightarrow IM_{AGImap} \rightarrow IM_{ACT-R}$).

HTAmap modeling language

The transformation gap between both levels of description is closed by the _HTAmap modeling language_. CAPs are being used to solve the mapping-problem from high-level to low-level description (SGT and ACT-R models). A CAP represents a generalized solution for the execution of an operator task (e.g. observe, scan, monitor, de-/activate, adjust, evaluate) using resources (i.e., declarative and
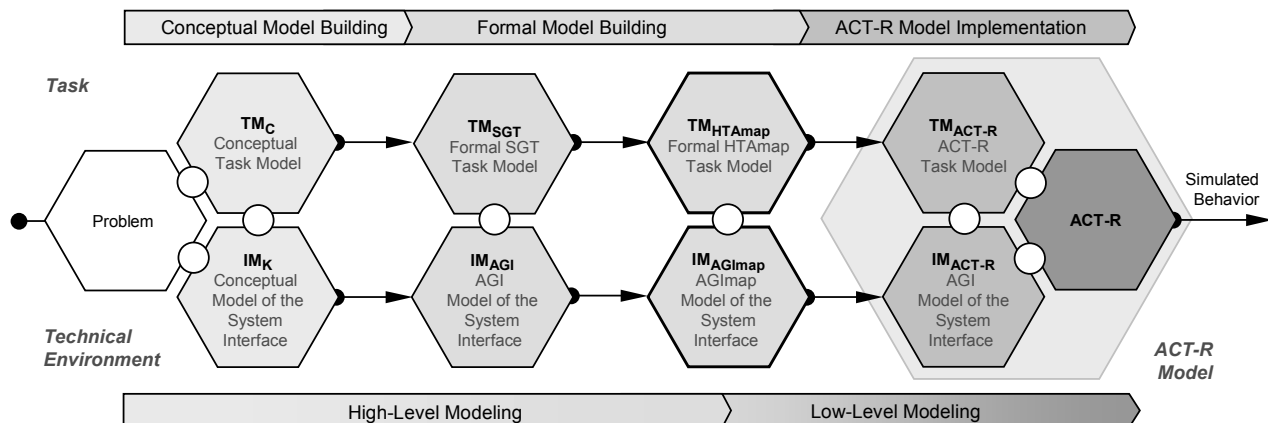

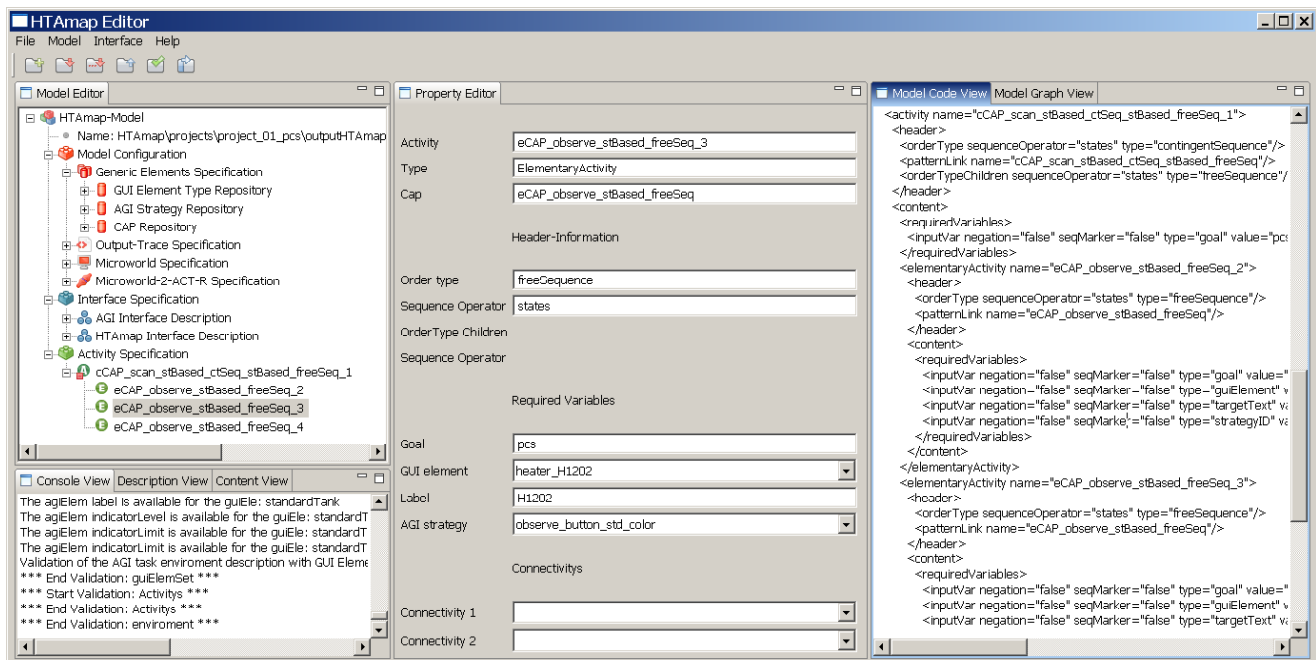Figure 1: Overview of the HTAmap Modeling Process.

Figure 2: Parts of the HTAmap Editor: *Model Editor* (left), *Property Editor* (middle) and *Model Code View* (right).

procedural memory as well as perception and action modules) to tackle a recurrent problem in a specific context. In general, CAPs describe the specific applications of identified SGT information handling operation at a less abstract level of ACT-R. In detail, a CAP comprises the necessary ACT-R declarative and procedural structures (i.e., production and chunks) and provides interfaces for parameterization regarding various tasks and environments. *Elementary cognitive activity patterns* (eCAP) are the basic building blocks for tasks. *Compound cognitive activity patterns* (cCAP) are used to represent the hierarchical structure of tasks. Complex task structures are represented via nesting of cCAPs: a cCAP can contain several eCAPs and/or other cCAPs (see Heinath & Urbas, 2007 for an overview). An activity is the combination of a CAP and task relevant knowledge. The HTAmap model itself is the combination of activities and task control knowledge (see Figure 2, right side). The kind of execution of activities is specified by their *order-types* which strongly refer to the plan concept of the SGT method: fixed, free or contingent sequence and simple choice.

HTAmap modeling editor

To proof the concept and reduce the entry level in cognitive modeling**,** a *HTAmap editor* has been implemented. Figure 2 shows a screenshot. The editor consists of three main parts: A *Model Editor* is used to define the repositories with the predefined model fragments (e.g., CAPs, AGI strategies, etc.), the AGI interface specification of the technical environment and the activity structure as result of the SGT task analysis. Secondly, task specific information are set in the *Property Editor*. Thirdly, the resulting HTAmap model - in XML syntax - is shown in the *Model Code View*. The transformation of HTAmap models into ACT-R code is realized automatically via the software tool *HTAtrans* that has been also integrated in the HTAmap modeling editor.

Conclusion and Outlook

HTAmap extends the modeling process to a wider user group. Main scope is the simplification of ACT-R modeling, the reuse of model fragments and consequently an increase of model application in the context of usability evaluation of Human-Machine Systems. The HTAmap model can be seen as a first approximation to model human behavior in ACT-R and provides a good starting point for further refinements. Currently, only a subset of CAPs is specified. The transfer of more 'associated' production rules into CAPs requires further verification and validation steps in the future.

Acknowledgments

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 (4), 1036-1060.

Ormerod, T. C. & Shepherd, A. (2004). Using Task Analysis for Information Requirements Specification: The Sub-Goal Template (SGT) Method. In D. Diaper & N. Stanton (Hg.), *The handbook of task analysis for human-computer interaction* (p. 347–365). Mahwah, NJ: LEA.

Heinath, M. & Urbas, L. (2007). Simplifying the Development of Cognitive Models using a Pattern-based Modeling. In *Proc. of 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. Seoul, Korea.

Ritter, F. E., Haynes, S. R., Cohen, M., Howes, A., John, B. & Best, B. (2006). High-level Behavior Representation Languages Revisited. In *Proc. of the 7th ICCM* (p. 404–407). Trieste, Italy: Edizioni Goliardiche.