Extending ACT-R's Memory Capabilities

Holger Schultheis (schulth@sfbtr8.uni-bremen.de)
SFB/TR 8 Spatial Cognition, Universität Bremen, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

Shane Lile (slile@alumni.calpoly.edu)
Department of Computer Science, California Polytechnic State University, San Luis Obispo CA 93407, USA

Thomas Barkowsky (barkowsky@sfbtr8.uni-bremen.de)
SFB/TR 8 Spatial Cognition, Universität Bremen, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

Abstract

To resolve several problems of ACT-R's *declarative memory* (DM), Schultheis, Barkowsky, and Bertel (2006) developed a new long-term memory (LTM) component, called *LTM*^{C}. In this paper we present two ACT-R interfaces which integrate LTM^C into ACT-R. Such integrating LTM^C makes it easily accessible to ACT-R modelers and allows more thoroughly evaluating it in its interplay with other components of a cognitive architecture. By considering four different memory phenomena we show that ACT-R with LTM^C is superior to ACT-R employing only DM and, thus, (a) LTM^C's benefits are not impaired when integrating it into a cognitive architecture and (b) using the newly developed interfaces improves ACT-R. In particular, integrating LTM^C into ACT-R allows computationally exploring memory conceptions which cannot be modeled with ACT-R utilizing only DM.

Introduction

As a cognitive architecture ACT-R (Anderson et al., 2004) aims at constituting a computational model of all human cognition. And indeed a wide variety of psychological phenomena have successfully been modeled using ACT-R. Despite this success, concerns have recently been raised by Rutledge-Taylor (2005) and by Schultheis et al. (2006) regarding the suitability of ACT-R's *declarative memory* (DM) component as a model of human long-term memory (LTM). In response to these concerns Schultheis et al. (2006) have developed a new LTM component called *LTM*^{C}.

As shown in Schultheis et al. (2006) this new LTM component is able to solve the problems currently associated with ACT-R's DM. However, so far LTM^C was only available as a stand-alone component which is disadvantageous for at least two reasons. First, one might argue that it is not possible to thoroughly judge the suitability of an LTM component without considering it in its interplay with other components of a cognitive architecture. Put differently, it would be possible that shortcomings associated with LTM^C would only become obvious when utilized in the framework of generally accepted assumptions about human cognition. Second, as a stand-alone component, LTM^C would be inaccessible for most cognitive modelers, because considerable programming expertise would be necessary to employ LTM^C for modeling. To eliminate these problems we developed two interfaces, *LTM-DM* and *LTM-Buffer*, to integrate LTM^C into ACT-R. This integration not only allows further evaluating LTM^C in the scope of generally accepted assumptions about the functioning of the human information processing apparatus, but also makes LTM^C and its advantages easily accessible for ACT-R modelers.

In this contribution we present the two interfaces and how they can be employed and illustrate their suitability to model several memory phenomena. Before describing the interfaces and their application we will give a short recap of ACT-R's existing memory structure and its problems as well as the structure and processes of LTM^C and how it is able to solve these problems. Subsequently, the ways LTM-Buffer and LTM-DM integrate LTM^C into ACT-R are explicated and evaluated. In concluding, issues for future work are highlighted.

ACT-R and its Limitations

In ACT-R's DM knowledge is represented by *chunks*. Chunks are data structures that contain one or more *slots*, which may contain values or other chunks. The slots a chunk contains are determined by its chunk type. Chunk types are declared separately for each model and are assumed to be fixed, that is, cannot and will not change during model runs. Thus, the chunk type specifications impose an unalterable structure on declarative knowledge. In particular, the imposed structure is assumed to be identical for knowledge currently being processed and knowledge stored in LTM (see Anderson et al., 2004, for a more detailed description of DM).

This means of representing knowledge is inappropriate to model human LTM, since the way chunks and chunk types structure knowledge renders knowledge situation-specific. Whereas this is unproblematic for knowledge currently being processed, a situation-specific representation of knowledge in LTM—which supposedly is the source of human knowledge in all situations—seems implausible.

Concretizing this general concern, Schultheis et al. (2006) identified the following three problems with ACT-R's knowledge representation: first, chunk types are too specific to their models. Every model defines its own chunk type(s). Consequently, at the moment it is uncertain (a) whether chunk types working well when considered in isolation still do so when considered together or (b) whether one could create a unified chunk structure which both represents all knowledge used in ACT-R models so far and still gives adequate modeling results. Second, the fact that chunk types cannot be altered during model runs makes the knowledge representation rather inflexible: Information retrieved together (i.e., in a chunk) from LTM in one context will be retrieved together in all contexts, an idea that is contrary to the common finding of context dependence of memory access (cf. Godden & Baddeley, 1975).

Finally, several studies (e.g., Erickson & Mattson, 1981; Park & Reder, 2004) have shown that humans in certain situations retrieve information from LTM which only partially matches the information originally requested. Yet, in ACT-R this effect does not arise from the basic architectural mechanisms. Instead, it is necessary to specify the degree to which partial matching is to occur between two chunks by hand.

An Outline[1] of LTM^C

Structure

In LTM^C knowledge is represented as a network of nodes. Every node comprises a name and a unique identifier. The name of a node is a string signifying which entity in the world this node stands for. The unique identifier is an alphanumeric sequence allowing to unambiguously identify and address each node[2]. In addition to its name and identifier, every node contains links to other nodes in LTM. These links represent associations between different entities—if two entities are associated with each other their corresponding nodes are mutually linked.

One noteworthy property of LTM^C is that the links between the nodes generally bear no meaning apart from indicating that the connected nodes are associated. In particular, links do not stand for relations, but relations are also represented as nodes. Thus, the fact that London is north of Paris would be represented by three nodes (London, north-of, and Paris) associatively linked to each other.

Besides knowledge about concrete entities such as "north-of", LTM^C also contains knowledge about categories of entities such as "direction relation" and knowledge about respective subsumption relations such as "north-of" "is a" "direction relation". Different from all other relations, however, subsumption relations are represented as links, since representing them as nodes would lead to infinite regress. By organizing the knowledge in a hierarchy (i.e., an ontology) the knowledge representation in LTM^C roughly takes the form of a tree with the most general entity as the root and concrete instances as the leaves.

This structure bears some resemblance to the knowledge representation proposed by Kokinov (2003) and Kokinov and Petrov (2001). However, due to the dissimilar processes at work in LTM^C, the overall functioning of LTM^C differs from the system proposed by Kokinov and colleagues. These processes will be described in the next section.

Processes

The processes employed to realize retrieval of knowledge from LTM^C are activation-based. Each node has an activation value that determines which nodes are retrieved on a certain request. This activation value is calculated on every new retrieval request as the sum of the node's base-level activation, the activation spread to that node, and some randomly varying activation (i.e., noise). Base-level activation, like in ACT-R, reflects the recency and frequency of a node's retrieval: the more frequently and recently a node has been retrieved previously the higher is its base-level activation.

The activation spread to a node, on the contrary, does not depend on events of the past, but on the current context (i.e., entities in working memory or the environment) in which the retrieval takes place. If, for example, a person is asked which direction relation holds between London and Paris the activation of the corresponding nodes "direction relation", "London", and "Paris" will be increased. Importantly, the nodes receiving some activation directly from the context will spread activation to nodes with which they are associated. Nodes receiving activation from other nodes will again spread activation to the nodes they are associated with and so on. This activation spreading is subject to four constraints: first, only a fraction of the activation just received is spread to other nodes. Second, activation is not spread back to that node from which the to be spread activation has been received. Third, the amount of activation which will be spread to associated nodes will be equally distributed to these nodes. Fourth, the received activation will only be spread if it is high enough (i.e., above a certain threshold).

Once spreading has stopped, the amount of activation accumulated in a node during spreading is added to its base-level activation. By furthermore adding noise—computed as in DM (see Anderson et al., 2004)—the final activation of each node is computed. On the basis of these final activation values the nodes to be retrieved are determined: only those nodes having an activation which is higher than the average activation of nodes in LTM will be retrieved.

This spreading process requires setting four parameters when using LTM^C: The amount of activation a node receives from the context at the beginning of the spreading (cA), the fraction of activation to be distributed to associated nodes (f), the threshold to terminate spreading (t), and the strength of the noise (n). The latter three were set to $f = 0.6$, $t = 0.01667$, and $n = 0.1$ for all simulations reported below, whereas cA was allowed to vary across the different models.

Given its structure and processes, LTM^C can easily account for basic human memory phenomena such as context and time dependence of knowledge availability or the fan effect (cf. Anderson, 1974). More importantly, as Schultheis et al. (2006) have shown LTM^C solves the problems which have been identified with ACT-R's LTM (see above): due to its more flexible structure LTM^C is better able to model the context-dependent grouping of knowledge in the scope of a retrieval. In addition, LTM^C allows modeling the effect of partial matching (cf. Park & Reder, 2004) more plausibly and parsimoniously. Accordingly, LTM^C keeps the advantages of ACT-R's LTM (i.e., being able to account for basic human memory effects) while at the same time avoiding some of its weaknesses.

Interfacing LTM^C and ACT-R

One main aim in developing the two interfaces LTM-DM and LTM-Buffer—besides making LTM^C available in ACT-R—was to check whether these advantages persist for LTM^C as a part of a general cognitive architecture. Both ways of inter-

[1]A more detailed description of the structure and processes of LTM^C can be found in Schultheis et al. (2006)

[2]Using strings and alpha-numeric sequences for identifying nodes are arbitrary representational conventions. The strings are meant to help the modeler to quickly see what is represented by a node. The alphanumerical sequences are just one way of assigning a unique key to each node which is necessary to realize a working implementation. Thus, using strings and alphanumeric sequences is not meant to suggest that conceptual knowledge is language based.

facing and their evaluation will be explicated in the following sections. The first module, called LTM-DM, is an extension to ACT-R's existing DM module. The second is called LTM-Buffer and is intended to be an ACT-R module to be used independently of DM. The two modules are intended to be used exclusively of one another and represent two different approaches to accessing knowledge from LTM^C and using it in ACT-R. In describing the interfaces we will first explicate those aspects of using LTM^C in ACT-R which both interfaces have in common and then go into more detail on the particulars of LTM-DM and LTM-Buffer, respectively.

Retrieving Knowledge from LTM^C

As argued above, the advantages of LTM^C compared to DM mainly arise from LTM^C's more flexible representation structure. Instead of imposing a fixed grouping of information onto the contents of LTM, the node network employed in LTM^C allows for a context-dependent and context-appropriate grouping of information. Put differently, the improvements realized by LTM^C are achieved mainly by avoiding the use of chunks and chunk types for representing long-term knowledge. Nothing, however, speaks against using chunks as a representation for the knowledge currently being processed (see above). Accordingly, the use of chunks for representing declarative knowledge in ACT-R is only problematic regarding LTM and thus the best way to integrate LTM^C into ACT-R seems to be to keep chunks as representations for processing while using LTM^C instead of DM as the LTM component.

Taking this approach the result of a retrieval request to LTM^C cannot directly be processed by the other ACT-R components, since a retrieval result is given in the form of a subnet of the overall net representing long-term knowledge and not in the form of chunks. Consequently, to interface LTM^C with ACT-R, retrieved subnets have to be recast as chunks. To achieve this we introduced a new construct called *mapping* which accompanies chunk type definitions. A mapping specifies how the slots of a chunk type relate to the nodes in LTM^C. If the newly developed LTM modules are employed in ACT-R, support for mapping definitions is automatically enabled. On a retrieval request the resulting subnet will be recast as chunks according to the defined mappings. This approach has the following advantages: first, ACT-R modelers can still use any types of chunks they like, that is, using LTM-DM or LTM-Buffer does not restrict the freedom of modelers to define and use particular chunk types. Second, by means of the mapping definitions chunk types are anchored in the general ontological structure of LTM^C. This potentially allows to compare and relate the knowledge used in different ACT-R models and thus alleviates one of the above identified problems of ACT-R's DM, namely the model specificity of chunks and chunk types.

By using the mapping mechanism retrieving knowledge from LTM^C is very similar to the process of obtaining chunks from DM. Figure 1 outlines the retrieval process. The first step of a retrieval is to issue a normal ACT-R request to the buffer of the corresponding module. The ACT-R LTM modules are able to use—employing the mapping—such an ACT-R request to activate nodes in LTM^C and spread that activation to those nodes' neighbors. Once this is complete, the

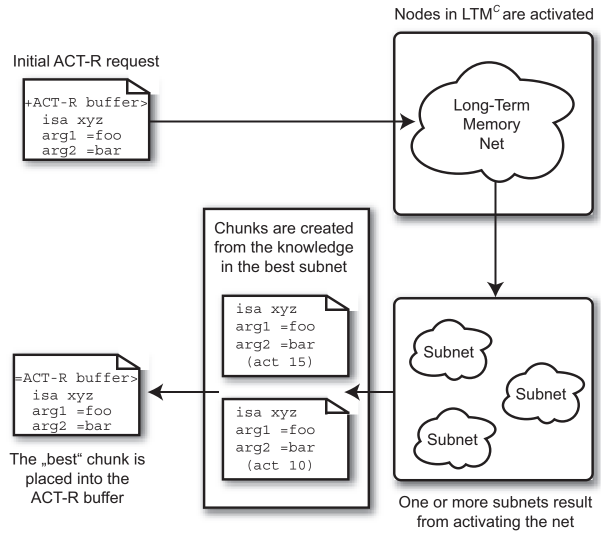

Figure 1: Retrieval process for LTM^C in ACT-R

nodes with an activation smaller than the average activation of the entire semantic net are discarded to create one or more disjoint subnets. The modules find the best subnet (i.e., the subnet with the highest total activation) and—again relying on the mapping—use it to create chunks of the requested type for ACT-R to use. The chunk with the highest activation is selected and placed in the buffer that requested the knowledge. The retrieval time is computed from the activation of the chunk finally put into the buffer using the same formula as in ACT-R.

Apart from this common procedure employed to retrieve information from LTM^C, the two interfaces differ with respect to certain aspects. Most notably, the overall memory conception realized in ACT-R changes depending on whether LTM-DM or LTM-Buffer is employed. The next two sections illustrate the differences between the two interfaces.

LTM-DM

LTM-DM is an extension to the existing DM in ACT-R. It offers the capabilities of LTM^C as an additional source of information, but if information is already present in DM then this may be used instead. Chunks retrieved from LTM^C are given a base-activation equal to their activation in LTM^C and then placed directly in DM to allow it to determine which chunk to place in its buffer. This might include any subsymbolic processes (e.g., spreading activation, noise, etc.) of DM. Thus, using LTM-DM establishes a two-component memory structure with a strict separation between the two components.

At first sight, using both DM and LTM^C might seem unreasonable, since—in light of the usual interpretation of DM as LTM—this might look like creating a system with two distinct LTM stores. Yet, as stated above, there are strong arguments for the stance that DM with its chunk structure is insufficient as a model of human LTM and is better viewed as holding the knowledge in a representation format used for the information currently processed. Thus, the memory structure resulting in ACT-R when using LTM-DM can be taken to implement not two types of LTM, but one long-term store

(LTM^C) and one store transiently holding the knowledge which is currently (or has recently been) processed (DM).

Seen this way LTM-DM is nicely in accord with a number of recent suggestions regarding the nature of the human memory system. For example, Baddeley (2000) proposed the *episodic buffer* as that part of working memory which (a) is a multi-dimensional store, that is, a store combining information from different modalities and sources, (b) can hold information for a longer duration than other parts of working memory, and (c) mediates between working memory and LTM. Since DM as used in LTM-DM also has these three features, it can be viewed as instantiating the episodic buffer instead of a second LTM.

Other areas of research that parallel LTM-DM are the sensory-perceptual episodic memory proposed by Conway (2001). In the case of Conway's research, LTM^C takes the place of autobiographical memory, "a type of memory that persists over weeks, months, years, decades, and lifetimes," and "retains knowledge at different levels of abstraction." Accessing the memories in LTM^C follows Conway's description of access to autobiographical memory. Conway describes these memories as "patterns of activation over the indices of autobiographical memory knowledge structures." His explanations of autobiographical memory describe a system not unlike LTM^C's semantic net, in which specific cues stimulate specific memories and spread activation throughout memory in order to retrieve larger collections of memories. DM is Conway's episodic memory. These structures "keep track of progress on active goals as plans are executed." Importantly, Conway (2001) describes episodic memory as having its own retrieval process, which is consistent with the strict separation of LTM^C and DM kept by LTM-DM.

Thus, LTM-DM allows modeling these and similar working memory conceptions; an advantage which will be explored in more detail below.

LTM-Buffer

The second module for interfacing LTM^C with ACT-R, LTM-Buffer, was implemented as a complete alternative to using DM. As such, LTM-Buffer is a complete stand-alone ACT-R module and has its own buffer, *pretrieval*, which needs to be used to access LTM^C. Instead of passing chunks to DM after traversing the knowledge in LTM^C, LTM-Buffer handles selecting which chunk to place in its buffer. This entails that the activation of the chunk put into the pretrieval-buffer stems directly from the processes working in LTM^C and is not influenced by subsymbolic computations in ACT-R.

Where LTM-DM suggests separate stores for long-term and working memory, LTM-Buffer's approach follows the view that working memory is not a separate memory store, but rather that it is highly-activated portions of LTM (see, e.g., Cowan, 1999). Thus, the memory structure realized in ACT-R when using LTM-Buffer is the same as in ACT-R proper. The only differences is that LTM is implemented by LTM^C instead of DM.

Evaluation

To evaluate the suitability of LTM^C in the scope of generally accepted assumptions about the functioning of the human information processing apparatus, we modeled several memory phenomena with ACT-R using only DM, ACT-R using LTM-DM, and ACT-R using LTM-Buffer. In doing so we had three aims: The first was to check the cognitive plausibility of LTM^C as part of a cognitive architecture. Second, we compared the performance of LTM-DM and LTM-Buffer to that of ACT-R using only DM to investigate possible advantages or disadvantages of using LTM^C in ACT-R. In particular, comparing the modeling abilities of ACT-R employing only DM and ACT-R using LTM-Buffer allowed attributing any differences between model performances directly to the respective LTM component employed. Put differently, any differences in model performance cannot be due to other parts of the architecture, since ACT-R using DM and ACT-R using LTM-Buffer differed only in the LTM component utilized. Finally, since the theoretical conceptions of memory underlying the two interfaces are quite different, it seemed interesting to compare LTM-DM and LTM-Buffer to determine which of the two might be more suitable as a model of human LTM.

In order to achieve these aims we considered four memory phenomena which seemed to be most informative regarding the above evaluation questions. The first two, namely the fan effect (Anderson, 1974) and a phenomenon related to the hierarchical structure of the knowledge representation in LTM (Sharifian & Samani, 1997), illustrate properties of human LTM for which established ACT-R models already exist. If our interfaces can perform at least as good as the ACT-R models this constitutes a proof of their cognitive plausibility, since the ACT-R models we used are generally thought to realize fine models of the two phenomena. The second two phenomena we considered were the *Moses Illusion* (Erickson & Mattson, 1981) and a case study reported in Baddeley (2000). These effects were chosen because they were assumed to illustrate more clearly possible difference between the two interfaces and ACT-R proper as well as between the two interfaces as such.

Model Setup

As just mentioned, established models were used for simulation of ACT-R proper whenever they were available. More precisely, for the fan effect and the hierarchical memory effect we employed the models presented in tutorial 5 and tutorial 1, respectively, of the ACT-R 6 distribution (see http://act-r.psy.cmu.edu/actr6/). For the Moses Illusion we wrote a new model which basically retrieves chunks using the partial matching mechanism of ACT-R, where the similarity values were set by hand to obtain optimal results. Importantly, for all of the standard ACT-R models DM consists of precisely those chunks needed for the task. In the models for ACT-R extended by LTM^C, on the contrary, DM was not equipped with any prior chunks. Any knowledge used during the task had to be retrieved from LTM^C. The information for the fan effect was represented by, for example, the nodes "hippie" and "park" which are associated with a node "in". For the hierarchy experiment the relevant knowledge was represented by pairs of subsumptions relations connecting three nodes as, for instance, "rose" `isa` "flower" `isa` "plant". For the Moses Illusion the represented knowledge consisted of facts about the involved objects as, e.g., "Moses" `isa` "Person", "Noah" `isa` "Person", "Noah" and "Ark" are both linked to "sailed", etc. In particular, LTM^C also contained the knowledge neces-

Table 1: Results for modeling Anderson's fan experiment ("Fan"), the Sharifian experiment ("Hierarchy") and the Moses Illusion experiment of Erickson and Mattson (1981). The values shown are correlations of model and experimental data.

Module	Fan	Hierarchy	Illusion
DM	0.864	0.948	0.988
LTM-Buffer	0.869	0.999	0.992
LTM-DM	0.817	0.999	0.980

sary for the fan effect and the hierarchy effect when modeling the Moses Illusion and vice versa, that is, was not taylored specifically for each modeled phenomenon.

All models together with the newly developed interfaces are available online[3]. The memory phenomena together with modeling results are reported in more detail below.

In the Realm of ACT-R ...

The Fan Experiment. The fan experiment of Anderson (1974) explored the hypothesis that the more knowledge a person has regarding a target concept, the longer it will take them to retrieve specific corresponding knowledge from LTM. In the experiment, participants had to memorize several facts such as "a hippie is in the park" and "a captain is in the cave." Knowing more facts about a specific person or place (e.g., the hippie is in the park and cave, versus the captain who is only in the cave) increased the time it took to recognize whether or not a target sentence was a known fact. Participants were also given foil sentences, in which a known person and place were paired up in a way the participant had not seen previously (e.g., the hippie is in the church; participants knew facts about the hippie and the church, but were never told "the hippie is in the church"), and were expected to respond that they did not recognize the target sentence. In accord with the hypothesis the main dependent measure in the experiment was the time the participants needed to verify the target sentences and, thus, in modeling this experiment we also concentrated on reaction times.

The column "Fan" in Table 1 shows how the response times of the ACT-R models using the different modules correlate to participants' response times in the actual experiment. As can be seen from the table, the data from the model runs employing LTM-Buffer and the model run employing LTM-DM strongly correlate to the experimental results, and are very close to the results obtained when using ACT-R's DM to model the effect.

The Hierarchical Spreading Experiment. The second memory effect we considered is related to the hierarchical organization of knowledge in LTM, illustrated by, for example, the results of Sharifian and Samani (1997). In their experiment the authors presented subjects with pairs of words, such as "flower" and "plant", asking them to identify whether or not the pairs were related. These pairs of words were based off of triads of words where the first and second words were directly related and the second and third words were directly related, thus making the first and third words indirectly related via the second word. For example, one such triad used in the experiment of Sharifian and Samani (1997) was plant-flower-rose. Sharifian and colleagues measured the time it took for subjects to correctly identify whether or not two words presented in a pair were related. Since, again, response times are the major focus of the original experiment, we used those as the dependent measure to evaluate our models.

As can be seen from column "Hierarchy" in Table 1, all three ACT-R modules show a very high correlation to the experimental results found by Sharifian and Samani (1997). In particular, the two newly defined interfaces are again as good as the available ACT-R model.

... and Beyond

Moses Illusion. If people are asked "How many animals of each kind did Moses take on the ark?" most of them will answer "two", although the correct answer would be "none": it was not Moses who sailed the ark, but Noah. This effect is called Moses Illusion and appears in a number of situations similar to the Moses Question. As Park and Reder (2004) show, this effect is most likely the result of partial matching processes working on LTM. Erickson and Mattson (1981) were the first to investigate this effect. They presented their participants with the Moses question and three additional questions of the same kind. The dependent measure was the relative frequency with which the participants answered the questions as if they were correct.

In ACT-R proper the only way of modeling partial matching and, thus, the Moses Illusion is by (a) enabling a special feature and (b) specifying similarity values between those concepts which are supposed to partially match (e.g., "Moses" and "Noah" in the above example)[4]. Consequently, our model for ACT-R proper realizes the Moses Illusion by using partial matching.

On the contrary, when using LTM-Buffer or LTM-DM, it is not necessary to specify any special retrieval mode or hand-picked similarity values. By relying on LTM^C the two interfaces can account for partial matching in human LTM employing the same memory structure and processes as in modeling the other memory effects (see Schultheis et al., 2006).

Column three of Table 1 shows the modeling results for the three modules. Like with the above memory phenomena the model accuracies are quite similar across the different modules. However, their more parsimonious approach to modeling this effect makes LTM-Buffer and LTM-DM superior to the model using only DM.

Episodic Buffer and Related Conceptions. As already explained when describing LTM-DM, this interface establishes a new overall memory conception in the ACT-R architecture. In particular, by creating such a memory LTM-DM allows modeling memory phenomena completely out of the

[3] http://www.sfbtr8.uni-bremen.de/project/r1/models/

[4] The ACT-R model presented in Budiu and Anderson (2004) realizes the Moses Illusion without using the partial matching mechanism of ACT-R. However, this is achieved by (a) using retrieval processes not available in the standard ACT-R distribution and (b) also relying on similarity values defined between different concepts. Thus, the Moses Illusion effect is essentially reduced to similarity values without explaining from which structures or processes these similarities might arise.

scope of ACT-R using only DM. Consider, for example, the memory performance of patient PV as reported by Baddeley (2000): PV showed normal LTM performance while at the same time having a reduced word (one item) and sentence span (5 items). Using standard ACT-R it is not possible to model such a pattern of memory performances since it is not possible to change short-term memory capabilities without affecting LTM capabilities, since both are instantiated by DM. Employing LTM-DM, on the contrary, easily allows modeling also such memory performance patterns as PV exhibits: LTM^C would serve as the (intact) LTM and DM could be modified, such that it implements the reduced short-term capabilities of PV.

Evaluation Summary

By considering the discussions and model results from the above sections it is possible to answer the three evaluation questions posed at the beginning. First, the suitability of LTM^C as a model of human LTM has been corroborated. LTM^C yields accurate modeling results of human LTM effects not only as a stand-alone component, but also when integrated into a general cognitive architecture: The correlations of model data and empirical data are very high for each modeled phenomenon (the lowest correlation was 0.817), and, in particular, the two LTM^C modules are at least as good as the standard ACT-R model in every of the modeled tasks. Second, LTM-Buffer and LTM-DM are superior to ACT-R using only DM, because they allow modeling memory phenomena with the same accuracy, but more parsimoniously than standard ACT-R. LTM-DM furthermore enables modeling memory conceptions and related phenomena previously not available in ACT-R. Third, LTM-DM seems to be slightly superior to LTM-Buffer. Besides modeling several memory phenomena as accurately as LTM-Buffer it additionally allows to model memory phenomena out of the scope of LTM-Buffer.

Conclusions

This paper presents two new interfaces for the ACT-R architecture, called LTM-DM and LTM-Buffer which integrate the LTM component LTM^C developed by Schultheis et al. (2006) into ACT-R. One major aim of integrating LTM^C into ACT-R was to more thoroughly validate its advantages by employing LTM^C as one part of a general cognitive architecture. Through modeling several memory phenomena we evaluated LTM^C as part of ACT-R and compared ACT-R using LTM-DM and LTM-Buffer with ACT-R using only DM. The results of this evaluation clearly show that LTM-Buffer and LTM-DM are at least as good as DM and in some cases considerably better in modeling human LTM. Particularly powerful seems the LTM-DM module, since it opens up a completely new field of memory phenomena to model with ACT-R.

Thus, our future work will concentrate on further exploring the possibilities and the power of LTM-DM as an extension to ACT-R. In addition, to complete the integration of LTM^C we will devise mechanisms for storing knowledge in LTM^C based on the mappings described above. If a mapping exists for every chunk type used in ACT-R, correspondence between any chunk and the knowledge stored in LTM^C can be established. Roughly speaking this allows transforming any chunk to a node and integrating it into the ontology given by LTM^C.

Acknowledgments

In this paper work done in the project R1-[ImageSpace] of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition is presented. Funding by the German Research Foundation (DFG) is gratefully acknowledged. We also thank Eduard Krieger for his help with the implementation and the anonymous reviewers for their valuable suggestions.

References

Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology*, *6*, 451 - 474.

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.

Baddeley, A. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Science*, *4*, 417-423.

Budiu, R., & Anderson, J. R. (2004). Interpretation-based processing: a unified theory of semantic sentence comprehension. *Cognitive Science*, *28*, 1 - 44.

Conway, M. (2001). Sensory-perceptual episodic memory and its context: autobiographical memory. In *Episodic memory.* Oxford University Press.

Cowan, N. (1999). An embedded-process model of working memory. In P. Shah & A. Miyake (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control.* Cambridge University Press.

Erickson, T. D., & Mattson, M. E. (1981). From words to meaning: a semantic illusion. *Journal of Verbal Learning and Verbal Behavior*, *20*, 540 - 551.

Godden, D. R., & Baddeley, A. D. (1975). Context-dependent memory in two natural environments: On land and under water. *British Journal of Psychology*, *66*, 325 - 331.

Kokinov, B. (2003). The mechanisms of episode construction and blending in DUAL and AMBR: Interaction between memory and analogy. In B. Kokinov & W. Hirst (Eds.), *Constructive memory.* NBU Press.

Kokinov, B., & Petrov, A. (2001). Integration of memory and reasoning in analogy-making: The AMBR model. In D. Gentner, K. Holyoak, & B. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science.* Cambridge, MA: MIT Press.

Park, H., & Reder, L. M. (2004). Moses illusion: Implication for human cognition. In R. F. Pohl (Ed.), *Cognitive illusions.* Hove: Psychology Press.

Rutledge-Taylor, M. (2005). Can ACT-R realize "Newell's dream"? In *Proceedings of the 27th annual meeting of the Cognitive Science Society.*

Schultheis, H., Barkowsky, T., & Bertel, S. (2006). LTM^c — an improved long-term memory for cognitive architectures. In *Proceedings of the 7th international conference on cognitive modeling, 2006, Trieste.*

Sharifian, F., & Samani, R. (1997). Hierarchical spreading of activation. *Proc. of the Conference on Language, Cognition, and Interpretation.*