# LTM$^C$ — An Improved Long-Term Memory for Cognitive Architectures

**Holger Schultheis, Thomas Barkowsky, and Sven Bertel**
{schulth,barkowsky,bertel}@sfbtr8.uni-bremen.de
SFB/TR 8, Universität Bremen, 28334 Bremen, Germany

**Abstract**

As Newell (1990) has argued, every cognitive architecture should include a knowledge base modeling human long-term memory (LTM). A critical analysis of current architectures, however, reveals that none of these seems to implement such an LTM component in a completely satisfactory manner. Based on the problems with existing approaches identified by this analysis, a new LTM component has been developed and fully implemented. This new LTM is able to model a wide range of memory effects and to resolve the problems identified with existing approaches.

## Introduction

In his seminal book, Newell (1990) states 13 constraints on the human mind which characterize human cognition. Due to their characterizing function, these constraints pose requirements for any cognitive architecture trying to truthfully model human information processing: an architecture can only be assumed to be mind-like if it exhibits the same constraints as the human mind.

One of the constraints Newell (1990) enumerates is the ability to persistently store and subsequently retrieve information about the environment. Accordingly, an essential part of every cognitive architecture should be, as Anderson and Lebiere (2003) recently termed it, a knowledge base which corresponds to human *long-term memory* (LTM). Moreover, including a component realizing some knowledge base, though necessary, is not sufficient to build a complete cognitive architecture. Rather, the way the knowledge is stored and accessed should mirror memory structures and processes as observed in humans.

In this contribution, we argue that current cognitive architectures do not realize LTM in a completely satisfactory manner. Thus, we developed and implemented a new LTM component improving on the existing approaches. This LTM will subsequently be termed LTM$^C$ as it is part of the cognitive architecture *Casimir* which is being developed by our group (cf. Schultheis, Bertel, Barkowsky, Freksa, & Seifert, 2005).

The remainder of this article is structured as follows: first, three current cognitive architectures, namely EPIC (Kieras & Meyer, 1997), Soar (Newell, 1990), and ACT-R (Anderson et al., 2004), are analyzed with respect to the way they implement LTM[1]. As a result of this analysis, several problems with each of the current LTM components are identified. Second, LTM$^C$ will be described and validated by showing (a) that it resolves the problems associated with the existing approaches and (b) its ability to accurately model human LTM effects. In concluding, we touch on work in progress aimed at making LTM$^C$ available to the ACT-R architecture.

[1]There are several versions of Soar and ACT-R available which differ in their implementation as well as underlying theoretical assumptions. In this article, we are discussing both architectures in their most recent version, i.e., Soar 8.6 and ACT-R 6.0.

## LTM in Current Architectures

Given the success of all three above mentioned architectures in modeling human cognition, the concerns regarding their LTM might not be immediately comprehensible. Thus, each architectures' LTM will be considered in more detail below.

### LTM in EPIC and Soar

In EPIC, LTM consists of several discrete blocks of information. All production rules are evaluated with respect to these blocks. If the condition of a production requires some information to be present for the production to be applicable and this information is in LTM, the production's actions will be taken. Implicitly, such a representation and use of knowledge assumes that all blocks in LTM are always completely available. Although such permanent availability might be desirable for information processing systems in general, it is not in accord with basic findings regarding human memory. First of all, several studies (e.g., Godden & Baddeley, 1975) have shown that the availability of information in long-term memory is context-dependent: information obtained in a certain situation $s$ is best retrieved in situations which are similar to $s$. Second, beginning with the work of Ebbinghaus (1885), numerous studies indicate that information may be lost from LTM over time (see Wixted & Ebbesen, 1991, for an overview). Finally, both context dependence and forgetting do not occur in an all or none fashion but gradually. That is, forgetting and change of context may make it harder to retrieve information from LTM, but not necessarily render retrieval impossible. Since the availability of knowledge in EPIC's LTM does neither depend on context nor on the time elapsed since information storage, it seems to be a rather inaccurate model of LTM in humans.

LTM in Soar comprises only production rules which are assumed to realize the associative and context-dependent nature of human memory. If the conditions for a certain production match the current context, its actions may retrieve certain information by putting this information into working memory. However, this associative context dependence is not graded: either a current context allows to retrieve certain knowledge (i.e., to apply a certain production) or it does not. Consequently, LTM in Soar is in disaccord with basic findings regarding human LTM.

## LTM in ACT-R

The LTM of ACT-R not only realizes the above stated aspects of human LTM, but also has been shown to be able to account for a wide variety of memory effects (see Anderson et al., 2004, for an overview). Nevertheless, Rutledge-Taylor (2005) has recently argued that ACT-R's LTM may be inadequate as a model of human LTM, since the representation structures employed in ACT-R's LTM are not general enough. Substantiating the broad concern of Rutledge-Taylor (2005), we identified three problems regarding both ACT-R's capabilities to model task-specific empirical data and the suitability of ACT-R's LTM as a model of human LTM in its entirety. In the rest of this section we will explicate the three identified problems in detail after shortly summarizing the characteristics of ACT-R's LTM.

**Characteristics of ACT-R's LTM** The basic building blocks of ACT-R's LTM are *chunks*. Chunks are structured collections of information: each chunk comprises a type specification, several *slots*, and, potentially, for each slot a value. Which slots a chunk offers depends on its type and a chunk type definition lists all slots a chunk of this type might have. The value of a slot may be a chunk, a string, or a digit. In addition, every chunk has an activation value which is a floating point number determining its availability.

It is further assumed in ACT-R that the chunk types and chunks available to a model—corresponding to one person being modeled—are specific to that model. Accordingly, the LTM of ACT-R is model-specific and comprises all chunks which have been created during the existence of a model.

To access chunks in LTM during a model run, the type of the chunk to be retrieved has to be specified. Only chunks of the given type can be retrieved. Optionally, values for different slots of chunks of this type may also be indicated. In principle, every chunk to be retrieved must match all of the slot values specified[2]. Of all of those chunks which match both the type and the slot values the one with the highest activation will be retrieved. In doing so, the activation value of each matching chunk in the scope of a certain retrieval is computed by summing three sources of activation: base level activation, spreading activation, and noise. The base level activation is determined by the frequency and recency of a chunk's use: the more frequent and more recent a chunk has been used the higher its base level activation. In contrast, spreading activation is not determined by the history of a chunk, but by the current context. Every chunk $c$ which is currently part of the context increases the activation of those chunks in LTM which contain or are equal to $c$. The third contribution to activation is a random number added to the activation value of a chunk as noise. It is assumed that this random number is logistically distributed with mean $0$.

Assigning an activation value to each chunk and using it as just described, nicely allows to model all of the basic memory effects listed in the previous section. Yet, as will be detailed in the next section, several problems are associated with the structure of ACT-R's LTM as well as its use.

[2] An exception to this will be discussed in the problems section.

**Problems with ACT-R's LTM** There are at least three issues raising doubt regarding the suitability of ACT-R's LTM as a model of human LTM:

- **Model specificity**: It is common practice when modeling memory effects (see e.g. Anderson & Reder, 1999), to insert just the immediately task-relevant chunks into LTM. Accordingly, LTM normally consits only of some 10 to 30 chunks. Obviously, it is highly implausible that a human's LTM contains just that few discrete blocks of information. It also remains unclear whether the accuracy of the model would still be given when the task-relevant knowledge would be embedded in additional, potentially interfering information. A further difficulty potentially arising when trying to construct a less task-specific LTM is due to the way knowledge is structured in the different ACT-R models. Similar to knowledge itself, knowledge structure, i.e., the chunk types used, is also model-specific. Therefore 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.
- **Inflexible structure**: Chunks group information and how they group it (e.g., the number of slots) is determined by their corresponding chunk types. Furthermore, chunks are the atomic elements of memory access: if successful, a retrieval request will result in a single, complete chunk. Accordingly, the atomic elements of retrieval are groups of information and, importantly, how the information is grouped depends only on the chunk types. In particular, the available types of grouping do not depend on the context in which retrieval occurs. Put differently, information that will be retrieved together in one context will be retrieved together in every context. Thus, chunks and chunk types impose an inflexible grouping on the knowledge in ACT-R's LTM—a problem which is aggravated by the fact that chunk types have to be preset by the modeler and cannot be changed or newly created during a model run. Such structuring however, does not seem to be in accord with the general idea of context dependence of human LTM.
- **Partial matching**: On the one hand, access of information in human LTM is quite focused. That is, if one tries to remember the solution to a mathematical problem which one has solved previously, the result of the memory access will almost never be anything completely unrelated to this mathematical question. Consequently, by default only those chunks in ACT-R's LTM which match the chunk type as well as all specified values given in the request (see above) may be retrieved. On the other hand, as for instance the investigations by Erickson and Mattson (1981) have shown, in certain situations, retrieval results may differ slightly from what has been requested. Since in such cases the result of the retrieval does not perfectly match the retrieval request, this phenomenon has been termed *partial matching*. The only way to accommodate such memory phenomena in ACT-R is to employ a mechanism also called partial matching. For this mechanism to work, the modeler has to detail values for pairs of chunks giving their respective similarity to each other. Once this is done

those similarities are taken into account during retrieval: all chunks of the specified chunk type are considered and the activation of those which do not perfectly match the requested slot values will be decreased inversely proportional to the similarity values between the differing slot values. Although this mechanism allows ACT-R to account for partial matching phenomena, it seems to be rather adhoc and is not arising from the structure and general mechanisms of ACT-R's LTM (cf. Altmann, 2000).

In summary, none of the three considered architectures seems to realize a knowledge base modeling human LTM in a completely satisfactory manner. Therefore, LTM$^C$ which improves on the existing realizations has been conceptualized and implemented. LTM$^C$ and its evaluation will be presented in the next section.

## An Improved LTM

The objective in developing LTM$^C$ was twofold: first, LTM$^C$ should be capable of representing human LTM in its entirety and not just with respect to specific tasks. Second, LTM$^C$ should improve on the existing models regarding cognitive plausibility. To achieve both goals it seemed reasonable to start from those aspects of existing LTM conceptions that have proven to be valuable and replace just those aspects of previous conceptions which the preceding analysis has identified as being problematic.

When comparing the three discussed cognitive architectures, ACT-R's LTM not only seems to be the most elaborate one, but also has been most successfully applied to modeling human memory phenomena (cf. Anderson et al., 2004). Moreover, the problems identified with this approach seem to be mainly structural. The mechanisms realizing context dependence and memory decay, namely base level activation, spreading activation, and noise, seem to satisfactorily mirror processes in human LTM. It is merely the chunk structure of knowledge representation which seems to cause problems. Consequently, LTM$^C$ realizes an advanced structure while at the same time largely adopting the activation related mechanisms of ACT-R. In doing so, LTM$^C$ is able to resolve all of the three problems associated with ACT-R's LTM without compromising those aspects of ACT-R's LTM which seem to be in accord with human LTM.

### Structure

The basic building blocks of the new structure are nodes and connections between them. Each node comprises a name, a unique identifier, and one or more connections to other nodes. The name of a node is a string and its main purpose is to indicate which entity in the world a certain node stands for. The unique identifier on the other hand serves rather a technical function in allowing the system to address every single node. As a third component of each node, connections establish links to other nodes. Some of these links ensure the efficiency of the technical realization by implementing a binary search tree on all nodes in LTM whereas the other links represent associations between nodes in LTM. Since only connections of the second type are of interest regarding psychological phenomena, the following description will concentrate on those.

The links in LTM$^C$ generally bear no meaning other than to establish associative connections between nodes. In particular, links do not constitute relations, but relations are also represented as nodes. As a result, there are essentially two kinds of nodes in LTM$^C$: object nodes representing objects (e.g., persons, buildings, countries, etc.) and relation nodes representing relations between entities (e.g., north-of, has-color, between, etc.). It is by linking these two types of nodes that knowledge is represented in LTM$^C$. For instance, the knowledge that London is north of Paris would be represented as depicted in Figure 1.


Figure 1: Three nodes with two links representing the fact that London is north of Paris.

Representing relations as nodes has at least three advantages: first, relations of differing arity can easily be accommodated in the same framework. Consider, for example, the relation "between" which has arity three. If relations were represented by connections, representing "between" would be rather difficult, since every connection links just two nodes. With relations as nodes however, "between" can be easily represented by linking the corresponding relation node to the three entities to be related. Second, relations as such can be primed, i.e., the process of activation spreading takes into account not only the entities which are related, but also the relations themselves. Third, by representing relations as nodes, categories and subsumptions of categories for relations can be built. For example, the knowledge that "north-of" is a "direction relation" can be explicitly encoded.

Apart from knowledge about concrete entities LTM$^C$ also contains information about classes or categories of entities. One important aspect of the knowledge about categories is their subsumption relations as in the above example of "north-of" and "direction relation". Since representing subsumption relations as nodes would lead to infinite regress they are represented by connections, called *isa-connections*.

To sum up, the representation structure employed in LTM$^C$ consists of object and relation nodes which are associatively linked. Knowledge about entities is represented by associative links between the corresponding nodes. Moreover, the representation structure supports organizing knowledge in a subsumption hierarchy (i.e., an ontology) of categories and concrete instances. Thus, the representation of knowledge in LTM$^C$ roughly takes the form of a tree with the most general entity as the root and concrete instances as the leaves (see Figure 2).

### Processes

Retrieval of information from LTM$^C$ is activation-based. Like in ACT-R, every node has an activation value which is the sum of the base level activation of the node, the activation spread to that node, and some randomly varying activation (i.e., noise). On every retrieval request, the activation of each node, starting out at $0$, is computed in the following way: first, elements currently in context (i.e., in working memory or in the environment) increase the activation of corresponding nodes in LTM. If, for example, a person is asked

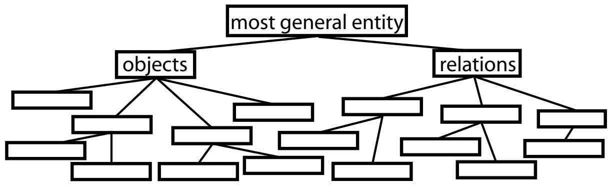


Figure 2: Structure of the knowledge representation.

which direction relation holds between London and Paris, the activation of the nodes "direction relation", "London", and "Paris" will be increased. The amount of activation which enters LTM is fixed and this amount will be equally distributed to all nodes which receive activation.

Second, the activation inserted into LTM by the previous step will be spread from node to node via their associative links. The activation that a node $N$ receives—either from the context or from another node—is added to its current activation. Furthermore, a certain fraction of the just received activation $f_{act}^N$ is spread to all nodes associatively linked to $N$ except for the node from which $N$ initially received the activation. The amount of activation any neighbor of $N$ receives is inversely proportional to the number of neighbors of $N$. More precisely, if $N$ has $m$ neighbors to which activation is spread, each of these neighbors' will be increased by $f_{act}^N/m$.

Spreading stops when the amount of activation to be spread from a node to its neighbors falls below a certain threshold. This mechanism not only avoids infinitely spreading activation, but also is in accord with the way the human nervous system works: a neuron will only transmit a signal (i.e., activation) to its neighbors if the activation it receives is above a certain threshold (cf. Kandel & Schwartz, 1985). However at the same time, using such a threshold introduces a free parameter into the system. Since too many free parameters potentially reduce the explanatory value of the model, we decided to make the threshold dependent on the amount of overall stimulation from the context $S_{context}$ by setting it to $S_{context} * 10^{-4}$. Dependence of the threshold on the amount of stimulation has its analogue in neural processing where neurons are less sensitive to signals after high stimulation (cf. Kandel & Schwartz, 1985).

There is one further mechanism constraining the direction of activation spreading. As described in the previous section, information in LTM$^C$ comprises both knowledge about categories and knowledge about instances. With the spreading process as introduced so far it would be possible that activation stemming from some instance node would spread to the category node the instance is subsumed by and from there back to some other instance node. For example, if the node "Kofi Annan" spreads activation to the category "Person" this activation may then spread from the "Person" node to every person of which there is knowledge in LTM. Put differently, when trying to remember something about Kofi Annan all the persons one knows would eventually come to one's mind. Since this seems highly implausible, any activation emanating from an instance node is constrained to never spread downwards in the subsumption hierarchy. Following a similar argument, activation emanating from a category node is constrained to never spread upwards in the hierarchy.

Once spreading has stopped, as the third step of the retrieval request, both base level activation and noise are added to each node's activation. The formula to compute the former is identical to the one used in ACT-R (Anderson et al., 2004). That is, the base level activation $B_i$ of a node at time $t_c$ is determined by the formula

$$B_i = \ln(\sum_1^n (t_c - t_j)^{-d})$$

where $n$ is the number of times this node has been used and $t_c - t_j$ is the time elapsed since the $j$th use of this node. A node is assumed to be used when it is part of the result of a retrieval request or when it is stored in LTM. Also identical to ACT-R are the formulas for computing (a) the noise to be added to each node's activation and (b) the retrieval latency.

The amount of activation a node has after adding all three sources of activation is the basis for determining which nodes to retrieve. Only those nodes can be retrieved which have an activation above a certain threshold. This threshold is defined as the average activation of the nodes in LTM. Choosing the threshold in this way has two advantages: first of all, it reduces the number of free parameters, since the threshold has not to be set by the modeler. Second, such a relative threshold is not susceptible to overall variation of activation (e.g., due to changes in global arousal). When using an absolute threshold, on the contrary, the threshold would have to be reset by the modeler with every change in global arousal.

It is further assumed that the result of a retrieval request will not be the set of all nodes with an activation above threshold, but just one connected subset of those nodes. A connected subset is defined as being a set of nodes $S_N$ such that for every pair of nodes $N_i, N_j$ in $S_N$ there is at least one sequence of nodes from $S_N$ $N_i, N_{i+1}, \ldots, N_{j-1}, N_j$ for which holds that for all $N_m$ with $i \leq m < j$ $N_m$ is associatively linked to $N_{m+1}$. Because a retrieval request may give rise to more than one connected subset, an additional criterion is needed to select one of those subsets as the result of the request. Like with selecting the nodes, the selected subnet should be the one most prominent compared to the others. Different from selecting the nodes however, average activation of each subnet does not seem to be sufficiently informative to determine prominence of subnets, since the number of nodes in a subnet can be assumed to have an impact on its prominence as well. To take into account both the activation and the number of nodes of a subnet we define prominence of a subnet as being the sum of the activation of all nodes in that subnet. That is, the result of the retrieval will be the subnet with the highest overall activation.

Given its structure and the just described processes, LTM$^C$ can easily account for all basic memory phenomena. Since (a) the activation inserted into LTM at the beginning of a retrieval request is added to those nodes which are related to the context and (b) the height of the nodes' activation determines whether they are retrieved, nodes related to the context and nodes associatively linked to these will have a higher activation and thus a higher chance of being retrieved. This shows that the availability of knowledge in LTM$^C$, like in human LTM, is context-dependent. Moreover, due to the use of base level activation, the time elapsed since the last use of the node does influence its activation value and thereby its chance of being retrieved. Hence, also the second basic memory phenomenon is taken into account: information may be lost from memory with time. Owing to the real valued nature of activation both context dependence and time dependence are modeled as occurring in a graded fashion which is in accord with human memory phenomena. Furthermore, the modeling capabilities of LTM$^C$ go beyond those basic memory effects. For example the *fan effect* (e.g., Anderson, 1974) arises straightforwardly from the way activation spreads from node to node: the more associative links a node has to other nodes the less activation will be spread to those nodes. By accounting for both basic memory phenomena and additional memory effects, LTM$^C$ clearly improves on the LTM components of EPIC and Soar. More importantly, LTM$^C$ also improves on the LTM of ACT-R in avoiding the problems which we have identified with the latter.

Consider first the problem of the inflexibility of the chunk structure used by ACT-R's LTM. This chunk structure predefines how information is grouped together and thus which information will be retrieved together. In contrast, our approach does not impose such a predefined structure on the knowledge in LTM. Which information is retrieved together is determined entirely by the processes of context activation, spreading activation, base-level activation, and noise. Of course, any subnet resulting from a retrieval request to LTM$^C$ is also grouped information and could be recoded as a chunk given a suitable chunk type. The important difference is that in our approach the grouping is only transient, i.e., context-dependent. Given another context, the information as it is retrieved from LTM$^C$ may well be grouped differently. Consequently, LTM$^C$ is more flexible than ACT-R's LTM component in providing information for different reasoning situations (i.e., in different contexts). Therefore, LTM$^C$ seems more plausible as a model of multi-purpose human LTM.

A second advantage of LTM$^C$ compared to ACT-R's LTM is the way that partial matching is realized. In ACT-R, by default, only chunks matching all of the specified characteristics of the requested chunk can be retrieved. This amounts to a deterministic, ungraded selection of knowledge from LTM. The only way to avoid such deterministic selection is by hand-coding the grading into the model using similarity values. In LTM$^C$, on the other hand, retrieval request specifications just determine which nodes are initially activated. Due to the activation-related mechanisms, this initial activation placement only influences the chances of some information being retrieved, but does not prevent any information from being retrieved. Thus, graded selection, i.e., partial matching, in LTM$^C$ directly arises from its structure and processes. To illustrate this point, a model of an experimental task pertaining to partial matching is presented in the next section.

## The Moses Illusion

Consider the question "How many animals of each kind did Moses take on the Ark?" Despite the fact that people know that it was Noah not Moses who built the Ark and took animals on it, people frequently answer "two" to the question like they would if Noah were the subject of the question. This phenomenon of answering a question which, in principle, has no answer due to an exchanged agent, object or verb, has been termed the *Moses illusion* (see Park & Reder, 2004, for an overview).

Table 1: Empirical and LTM$^C$ model results with respect to the first experiment of Erickson and Mattson (1981). Model results for each question are based on 100 model runs.

| **Question** | **Study Results** | **Model Results** |
|---|---|---|
| Noah / Moses | 81% | 81% |
| Jonah / Joshua | 40% | 46% |
| Bell / Edison | 44% | 48% |
| Ahab / Nemo | 44% | 43% |

Several explanations for this phenomenon have been proposed. Although the debate which explanation might be correct is not completely settled yet, as Park and Reder (2004) argue, the empirical results accumulated over the years best support the assumption that the Moses illusion stems from partial matching. According to this explanation all of the information given in the question, including "Moses", is used to request facts from LTM. Since there is no information on any Moses having taken animals on an Ark, this retrieval request does not perfectly match any content of LTM. Yet, Moses is assumed to be sufficiently similar to Noah to frequently retrieve the required information, namely "two", from LTM (e.g., Moses as Noah is part of a biblical tale and both are described as being old). It is argued that the question with Moses partially matches the facts which are stored with Noah and his Ark to retrieve this information. To model such a partial matching effect in ACT-R, the similarity between Moses and Noah would have to be coded explicitly. On the contrary LTM$^C$ does not need any hand-coded similarity values to account for such a partial matching effect.

To corroborate this claim, we modeled the first experiment reported by Erickson and Mattson (1981). In this experiment, 28 participants had to answer the four following questions: "How many animals of each kind did Moses take on the Ark?", "In the biblical story, what was Joshua swallowed by?", "What is the nationality of Thomas Edison, inventor of the telephone?", and "In the novel 'Moby Dick', what colour was the whale that Captain Nemo was after?". In a subsequent test, Erickson and Mattson (1981) determined which participants had the knowledge necessary to answer the questions in their correct formulation, and for those determined the percentage with which the illusion occurred (see Table 1).

In modeling these experimental results all information necessary to answer the four questions was inserted into LTM$^C$. For example, nodes and links were created to encode that Noah sailed the Ark, Moses and Noah are both old and part of a biblical tale, the Ark contained pairs of animals, etc. Note however, that the task-relevant information was not the only knowledge in LTM$^C$ during the simulation. Rather, all instance nodes were linked to category nodes and a considerable number of additional nodes (about 700) were part of LTM. Thus, the memory effects exhibited have been obtained in the scope of a considerable amount of additional and potentially interfering information.

As can be seen in Table 1, the modeling results mirror those which have been observed in the original experiment: the correlation between model and empirical results is $0.99$. To obtain these results, the only parameters which were fit were the overall amount of activation (set to 50) inserted into LTM on a retrieval request and the fraction $f_{act}$ of activation distributed from a node to its neighbors (set to 0.7). In particular, no similarity values between certain entities had to be set to obtain those partial matching results. For example, the percentage difference between the Noah / Moses and the Jonah / Joshua questions arises directly from the knowledge representation itself. Because Moses has more links to Noah (e.g., both are old and part of a biblical tale) than Jonah has to Joshua (in fact, nothing much is assumed to be known about Joshua) activating Moses will spread more activation to the relevant knowledge than activating Joshua will regarding the second question.

This model not only shows that LTM$^C$ can account more parsimoniously for partial matching effects than ACT-R's LTM but also, in contrast to usual ACT-R models, obtained its results on the basis of more realistic knowledge base, i.e., one containing more than just the task-relevant knowledge.

## Related Work

LTM$^C$ bears some resemblance to semantic networks as proposed by Collins and Loftus (1975). Yet, there are several notable differences: first, their approach is more qualitative than quantitative. That is, activation spreading processes are only roughly sketched giving no account of how exactly activation is distributed from node to node. Second, there is no mechanism for changing the availability of knowledge due to practice (i.e., no learning). Third, relations are represented as edges between the nodes, not as nodes. As our fully implemented LTM$^C$ addresses all of these issues it is also a real improvement on the LTM Collins and Loftus (1975) proposed.

## Conclusions

In this article, a critical analysis of the realization of LTM in current cognitive architectures has been presented. This analysis revealed several problems associated with each of the existing approaches. Based on these issues, the new LTM component LTM$^C$ has been conceptualized and implemented for cognitive architectures improving on the existing ones. More precisely, LTM$^C$ accounts for (a) basic human memory phenomena such as context and time dependence of knowledge accessibility, (b) various additional memory phenomena like the fan effect, (c) the structural flexibility of human LTM, (d) partial matching phenomena more straightforwardly and parsimoniously than ACT-R, and (e) the fact that a cognitively plausible architecture should allow for a task-unspecific knowledge base.

From a technical perspective LTM$^C$ is a software module with a clearly defined interface allowing LTM$^C$ to be integrated into existing cognitive architectures. Currently, we are working on integrating LTM$^C$ as an additional module into ACT-R which will make this improved LTM available to a larger community of cognitive modelers.

## 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, Shane Lile, and Johannes Quistorp for their help in implementing the system.

## References

Altmann, E. M. (2000). Memory in chains: A dual-code associative model of positional uncertainty. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the third international conference on cognitive modeling.* Veenendaal, The Netherlands: Universal Press.

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.

Anderson, J. R., & Lebiere, C. (2003). The newell test for a theory of cognition. *Behavioral and Brain Sciences*, *26*, 587 - 639.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: new results and new theories. *Journal of Experimental Psychology: General*, *128*, 186 - 197.

Collins, A. M., & Loftus, E. F. (1975). A spreading activation theory of semantic processing. *Psychological Review*, *82*(6), 407 - 428.

Ebbinghaus, H. (1885). *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Leipzig: Duncker & Humblot.

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.

Kandel, E. R., & Schwartz, J. H. (1985). *Principles of neural science* (2 ed.). Amsterdam, The Netherlands: Elsevier.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, *12*, 391 - 438.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University 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., Bertel, S., Barkowsky, T., Freksa, C., & Seifert, I. (2005). Casimir: Comprehensive computational modeling of mental spatial knowledge processing. In *Poster Proceedings of the Conference on Spatial Information Theory (COSIT). 14. - 18. Sept. 2005, Ellicottville, New York.*

Wixted, J. T., & Ebbesen, E. B. (1991). On the form of forgetting. *Psychological Science*, *2*, 409 - 415.