An Agent Memory Model Enabling Rational and Biased Reasoning

Annerieke Heuvelink[1, 2]
[1]*TNO Defence, Security, and Safety*
Department of Training and Instruction
P.O. Box 23, 3769 ZG Soesterberg
The Netherlands
Annerieke.Heuvelink@tno.nl

Michel C. A. Klein[2], Jan Treur[2]
[2]*Vrije Universiteit Amsterdam*
Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam
The Netherlands
Michel.Klein@cs.vu.nl, *Treur@cs.vu.nl*

Abstract

This paper presents an architecture for a memory model that facilitates versatile reasoning mechanisms over the beliefs stored in an agent's belief base. Based on an approach for belief aggregation, a model is introduced for controlling both the formation of abstract and complex beliefs and the retrieval of them based on their activation history. An implementation of the presented mechanisms illustrates how it can be used in intelligent agents that exhibit human-like (biased) reasoning as well as rational reasoning.

1. Introduction

Agents commonly store beliefs about the state of the world in what is often called a world model or belief base that grows over time. Reasoning can be done by applying inference rules to the stored beliefs. This way of storing and using beliefs seems not very similar to the human way of using memory. For example, humans often forget specific details, but still remember aggregated details or abstractions of specific facts. Therefore, human reasoning is not always sound and rational, but sometimes heuristically or biased, based on abstractions or generalizations that are (more easily) available.

A notion describing the human way of using memory is "episodic memory" [14, 15]. This refers to the memory of events, times, places, associated emotions, etcetera in relation to an experience, which is formed automatically. Episodic memory is contrasted to semantic memory, which is the memory of factual or conceptual knowledge. It is sometimes believed that episodic memories are converted into semantic memories over time. In this process, most knowledge about the specific event is generalized, and abstractions are stored in the semantic memory. Nuxoll and Laird [12] distinguish 4 phases in episodic memory, namely encoding, storage, retrieval and use of the retrieved memories.

Taking these ideas about human memory as a point of departure, it has been explored how an agent may form, store, retrieve, and use aggregated and abstracted beliefs. Which perspective and level of aggregation is needed, is application and task dependent. For that reason, the algebraic approach developed in [5] uses a general aggregation operator parameterised by a specific constraint that can be used in a recursive manner. The formalism allows for the specification of complex beliefs, for example, a "last most certain belief".

In this paper, a model to control the formation and use of aggregations is proposed. The control mechanism has a hybrid nature: abstract beliefs are automatically generated, while complex beliefs are only generated if the agent has a specific focus, induced by the task of the agent. Retrieval of the various belief types is based on their availability, which depends on their activation history, and on the task focus of the agent.

The model builds on the long-standing idea that forgetting is the key to proper functioning of memory [8]. Recently, Schooler and Hertwig [13] performed simulation experiments that demonstrate that forgetting might facilitate human inference performance. To implement forgetting in a computational model, beliefs usually get an availability value, which decays over time. However, it is human-like that although specific details cannot be remembered, abstractions of beliefs can. Such beliefs are more available than the more detailed beliefs they are abstractions of, especially in case of multiple detailed instances. This feature of memory is taken into account by the approach introduced.

The remainder of the paper is structured as follows. First, Section 2 introduces the various belief types that are distinguished within an agent's memory, and describes important concepts that are used for the modelling of the agent's memory. Section 3 demonstrates the operationality of the model by presenting a Prolog implementation of the occurring memory processes during a cognitive reasoning task. The effects of these processes on rational and biased task execution are elaborated on in Section 4 with the help of a scenario. Section 5 relates the work to other research on the modelling of (episodic) memory, summarises the research, and exposes future research plans.

2. Memory Model Concepts

In this section the predicates of the memory model are introduced that bring about the aspects mentioned in 1.

2.1. Basic Belief Type

The need for a memory model as proposed in this paper was identified in the context of military training simulation [3, 4]. An example agent task is the formation of a tactical picture of the environment, which entails the classification and identification of surface radar contacts. For modelling its behaviour it is needed to explicitly represent the time at which a belief was actually believed by the agent, to enable (biased) reasoning over (possible inconsistent) beliefs over time. For example, when at time t it is believed that the position of a contact is [x1, y1], while at time t + n it is believed to be position [x2, y2], the average speed of the contact can be inferred. This is useful because the speed may contain information concerning its identity, e.g., large ships that are neutral usually do not sail faster than 20 knots.

In order to logically represent time and other aspects, such as the uncertainty and source of information, every belief receives a number of labels, and is represented by *belief(P, O, V, T, S, C)*. This denotes the belief that property P holds for object O with value V at time T, based on source S and with certainty C. An example belief denoting that it is believed at time 8 with high certainty that the identity of contact1 is friendly because of radio conversation is: *belief (identity, contact1, friendly, 8, radio, 0.9)*. The value, time, and certainty label of basic beliefs about a specific property and object are often used to reason about trends in those beliefs, which can lead to new beliefs. For example, a new belief can be formed about a contact being a merchant, and therefore neutral, due to beliefs about it sailing in a straight line. The certainty of the former is determined by the period, as by the certainty, with which the latter is believed. For the deduction of other beliefs it is often important to deduce the last, or most (un)certain belief about something. For example, the highest certainty with which it was once believed that a contact fired is relevant for deducing whether it might be hostile.

2.2. Types and Control of Aggregations

The need was identified for the memory model to support in addition to the conscious reasoning processes on beliefs, more unconscious processes on beliefs, referred to as aggregations. The approach introduced in [5] allows for the formation of arbitrary aggregations over beliefs, which result in aggregated beliefs. In this approach an aggregated belief is defined as an aggregation that takes the form of a constraint (e.g., *highest*) that must hold for a certain variable (e.g., *T*) of a partially specified belief (e.g., *belief(identity, contact1, friendly, T, S, C)* in which the P, O, and V are specified while the T, S, and C are left variable). Examples of aggregated beliefs that an agent can form are mentioned in section 2.1, e.g., a belief about the highest certainty with which something was believed.

Furthermore, two types of aggregations are identified. The first process type aggregates over beliefs with many detailed arguments to form an abstract belief that 'forgets' certain details. Such a process is referred to as a *basic aggregation*. This process simply abstracts from a belief occurrence by leaving one or more of its arguments free. The basic aggregation process generates an *abstract_belief (AbstractionType, ForArguments, WithRanges)* as output and requires a *belief(P, O, V, T, S, C)*, or another *abstract_belief* as input.

For example, the basic belief *belief(identity, contact2, hostile, 4, vision, 0.9)* can be abstracted into:

abstract_belief(certainty_temporal_abstraction,
for(identity, contact2, hostile, free, vision, free),
with_ranges(nr, nr, nr, any, nr, any))

by executing the following aggregation:

agg(certainty_temporal_aggregation(C, T),
abstract_free_free(_, any, _, any, P, O, V, S),
is_available_at_current_time(belief(P, O, V, T, S, C)))

These abstract beliefs are formed in an automatic way, as a feature of memory. However, which abstractions happen and the amount of detail that is abstracted from, might be susceptible to certain internal aspects, e.g., stress. Therefore, it is required that these basic aggregations can be controlled at each time point. For this the predicate *in_memory_focus_at(B, T)* is used. When a certain belief is in memory focus at a certain time, the aggregation at which it is defined will execute; this determines which aggregations happen automatically (bottom-up). It is assumed that only abstract beliefs can be in memory focus, since their formation is unconscious and a general characteristic of the functioning of memory. Note that only a limited number of basic aggregations exist, due to the limited number of arguments that can be abstracted from.

The second process type that the memory model supports is *complex aggregation*, and aggregates over retrievable beliefs to form a *complex belief.* A complex belief aggregates information from the level of the beliefs that were the input of the complex aggregation that formed it. These can be basic or abstract beliefs, as other complex beliefs. A complex aggregation reasons over arguments of beliefs taking a specific constraint into account. An example of a complex belief is:

complex_belief(last, for(identity, contact2, hostile, free, free, free), with_ranges(nr, nr, nr, any, any, any), has_values(given, given, given, 4, vision, 0.9))

It is formed by the following aggregation:

agg(temporal_source_certainty_aggregation(T, S, C),
highest_free_free_free(_, any, _, any,_ ,any, P, O, V),
is_available_at_current_time(belief(P, O, V, T, S, C)))

Many complex aggregations exist, due to the wide variety of possible constraints. The constraint that is taken into account will be situation and task dependent. Therefore, whereas the process itself is automatic, the start of a formation of a complex belief is controlled.

For this the predicate *in_task_focus_at(B, T)* is used. Which beliefs are in task focus at a certain time should be determined top-down, based on the current goals of the agent. Complex beliefs that are in task focus are either retrieved from memory, or actively formed by the aggregation as which they are defined. In either case, their occurrence is always tailored to the current (part of the) task that is attended to. Basic beliefs in task focus are always attempted to be retrieved.

2.3. Activation History and Availability

All beliefs are at the time of their formation present in the part of the agent's memory referred to as its working memory (WM). In addition, it is possible that they are present in WM at a later time, after they have been retrieved from long term memory (LTM). When transferring beliefs from an agent's WM into its long LTM, it is important to indicate when they were formed or retrieved in WM. For this the two-place predicate *active_at(B, T)* is used. For every *belief(P, O, V, T, S, C)* that is found in the agent's belief base it holds that *active_at(belief(P, O, V, T, S, C), T),* with *T* the time it was formed, thus active in WM. This explicit reference to the time at which a belief is active in WM, is required to determine for later time points whether it is possible or easy to retrieve it from LTM. This depends on the availability of a belief, denoted by *has_availability_at(B, A, T).*

In the literature many theories can be found on the factors that influence the chance that a belief is retrieved from LTM, and even a few computational models. It was decided to base the beliefs availability determination on one these models, namely the one embedded in the cognitive theory and architecture ACT-R [1, 2]. In ACT-R declarative knowledge is presented by chunks, whose activation values determine their availability. The full equation that determines a chunk's activation takes into account several aspects, but for this paper only the base-level activation is taken into account. The base level activation B_i reflects the recency and frequency of use of the chunk, and is calculated by:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) + \beta_i$$

n**:** The number of presentations for chunk *i*
t_j: The time since the j_{th} presentation.
d: The decay parameter. Standard this one is set at 0.5 [1].
β_i: a constant offset

In the memory model proposed here, chunks are replaced by beliefs. A belief B's presence in WM, i.e. each *active_at(B, T)*, is interpreted as a 'presentation' of a chunk. The constant offset β_i varies between the belief types. For basic and complex beliefs it holds that β_I represents the belief's initial impression value, denoted by *has_impression_value(B, I).* For abstract beliefs, β_I represents its deduced impression value; *has_deduced_ impression_value(B, I).*

The latter is the summation of the impression value of the belief that it was abstracted of, with an additional amount that depends on its abstraction type. Currently, each free argument of an abstract belief adds 0.2 to the deduced impression value of that abstract belief, so for an abstract belief of type *temporal_source_certainty_ abstraction*, this amounts to 0.6. This ensures that in principle beliefs at a certain abstraction level, i.e. beliefs of which some details are forgotten, are more available than beliefs at a lower level of abstraction.

The impression values of beliefs can be influenced by various factors, e.g., by the emotional response it triggered, or by its importance for the agent's current task. For the current paper it was determined that the impression value of new basic beliefs is determined by it importance for the current task, expressed by *has_ importance_for_task_at(B, I, T).* The complex beliefs that are top-down determined to be required for the task, and are therefore in task focus, inherit the same importance value as the basic belief from which their determination stems. The model has the ability to determine such an initial value based on the context of the task; currently all new basic beliefs receive the same impression value of 2.

2.4. Retrieval and Aggregation Costs

In ACT-R a chunk's activation value does not only determine which chunk will be retrieved, but also how long it will take. In the proposed memory model instead of process duration, a more general factor of reasoning or retrieval actions cost has been incorporated. Intuitively, the costs to retrieve a belief should depend on its availability, therefore it was decided that its retrieval costs are inversely proportional to its availability. In addition, a threshold is required that denotes the minimal availability a belief should have in order to be retrievable. This threshold is arbitrary; for mathematical reasons it was set at 1. The costs of a complex aggregation are determined to be equal to the costs of the retrieval of all relevant beliefs, i.e., beliefs that it takes into account for its formation. So the costs of forming *complex_belief(last, for(identity, contact2, hostile, T, S, C), with_ranges(nr, nr, nr, any, any, any), has_values(given, given, given, X, Y, Z))*, is equal to the summation of the retrieval costs of all retrievable beliefs *belief(identity, contact2, hostile, T, S, C)*.

3. Implementation

The developed model was implemented in Prolog [16]: http://www.few.vu.nl/~heuvel/IAT-MemoryModel.pl. In this section, the implementation choices are explained using fragments of the program.

A run of the agent starts by requesting the start of a specific scenario. This sets the current time at 0, fills the agent's LTM with certain beliefs relevant for that scenario, and starts the agent's execution cycle by calling the *scenario_loop* clause:

```
scenario_loop(N):-
    current_time(T1), retract(current_time(T1)),
    T2 is T1 + 1, assert(current_time(T2)),
    determine_availabilities_beliefs,
    determine_availabilities_abstract_beliefs,
    sense_and_form_beliefs(N),
    set_task_focus(N),
    retrieve_beliefs,
    deduce_complex_beliefs,
    reason_and_form_beliefs(N),
    set_memory_focus(N),
    deduce_abstract_beliefs,
    scenario_end(N).
```

The scenario loop starts by propagating the current time with 1 step. Next, the availabilities of basic and complex beliefs are determined following the ACT-R formula. The determination of the availability of abstract beliefs varies slightly from that of basic beliefs, and is therefore called separately. Basic beliefs always have a maximum of one *impression_value* attached to them, which is determined at their formation time. However, abstract beliefs might have multiple *deduced_impression_value*'s attached to them, because various basic beliefs may be abstracted into the same abstract belief. Therefore the constant offset $\boldsymbol{\beta}_I$ of an abstract belief is determined to be the average of all its *deduced_impression_value*'s.

Next, a predicate is called with a specific reference to the scenario that is executed; *sense_and_form_beliefs(N)*. Its clauses constitute the agent's working memory input stemming from the agent's interaction with the outside world. They define for each time step of a specific scenario which beliefs enter memory, together with their costs (fixed at 1) and impression value (fixed at 2).

After this another scenario-specific predicate is called; *set_task_focus(N)*. Its clauses define for each time step of a scenario which beliefs are required from the task's perspective, and with which importance, see Section 2.2. The *in_task_focus_at* predicates embed belief templates of which an instantiation has to become active in WM. In case of a basic belief template this is attempted by a retrieval action. In case of a complex belief template it is first attempted to retrieve it, and when this fails, it is actively formed by executing the aggregation as which it is defined.

Because complex beliefs always have a temporary nature, their retrieval, instead of formation, allows for biases. Therefore, it is desired that this retrieval can be influenced, e.g., by stress. For that reason the retrieval process of complex beliefs takes a specific, tunable, retrieval threshold into account. In the current scenarios this threshold is fixed at 100, so complex beliefs can never be retrieved and are always newly formed.

The *retrieve_beliefs* clauses try to retrieve instances from the agent's LTM for all the beliefs that are currently in task focus. For each of them it executes:

```
determine_retrieval_of_belief(BT):-
    current_time(N),retrieval_action(BT, B, RB),
    has_availability_at(B, A, N),
    assert(active_at(B, N)), assert(active_at(RB, N)),
    K is 1 / A, assert(costs_to_retrieve_at(B, K, N)),
    assert(costs_to_ perform_retrieval_at(RB, K, N)).
```

The *retrieval_action(BT, B, RB)* determines which belief *B* from LTM is retrieved based on the belief template *BT* that is in focus. Retrieved belief *RB* denotes the instantiation of *BT* with the values of belief *B*. The retrieval action retrieves that belief that is coherent to the requested belief template and has the highest availability value of all the beliefs that are. Next, both *B* and *RB* are asserted to the agent's memory. The first because it is again active in WM, the second as a short cut for the reasoning process that will operate on it later. Then, the availability of belief *B* is queried and used to determine its retrieval costs, and thus of the formation of retrieved belief *RB*, which is the inverse of its availability.

An important feature or the memory model is the way in which the *retrieval_action* operates when a basic belief is the object of retrieval, e.g., *BT* is *belief(new_detected, contact2, _, free, _, _)*. This template can be filled by a basic belief, but also by an abstract belief, as long as the arguments that are designated as *free* in the template are among those where the latter abstracts from. This feature implements that, given that abstract beliefs have multiple instances and are as such more available that the specific beliefs they were abstracted from, abstract beliefs might still be retrievable while the detailed beliefs may not.

The execution cycle of the agent continues with the *deduce_complex_beliefs* predicate. Its clauses will execute the aggregations at which the complex beliefs are defined that are in task focus, but could not be retrieved. For the details concerning the execution of these aggregations, see [5]. Note that when a certain aggregation requires a complex belief as input, it is first attempted to retrieve it. When that fails, the required complex belief is formed by execution the aggregation at which it is defined. The result, so the complex belief that was formed as an intermediate step to the required complex belief which is in task focus, is also memorized. The costs of the formation of a complex belief by an aggregation are equal to the sum for all the basic beliefs that were retrieved for it, as explained in Section 2.4.

After all the beliefs that are required for reasoning have been retrieved or formed by aggregation in WM, the *reason_and_form_beliefs(N)* predicate is called. Its clauses specify for each time step of a scenario the specific reasoning rules that execute. These rules operate on the beliefs in WM and enter one or more new beliefs into memory, together with the costs of their generation. The latter is the summation of the costs to retrieve and/or aggregate the beliefs that functioned as the rule's input.

At the end of the scenario loop the memory's intrinsic feature of performing basic aggregations to form abstract beliefs is executed. The basic aggregations that happen are defined by the predicate *in_memory_focus_at* and these are set by calling the scenario-specific predicate *set_memory_focus(N).* In the current model it is assumed that at each time step all possible basic aggregations of the *T*, *S* and *C* arguments of beliefs are in memory focus. This entails that from a single basic belief 7 abstracted beliefs of different types are formed. For an abstract belief the number of additional abstract beliefs that can be abstracted from it depends on its abstraction level.

When *deduce_abstract_beliefs* is called, the basic aggregations that are in memory focus execute for each of the basic and abstract beliefs that are in WM. The process of abstracting basic beliefs into abstract beliefs that, when formed on the basis of multiple basic beliefs, are more and therefore longer available, resembles the transfer of episodic to semantic memory.

The last predicate of the scenario loop, *scenario_end(N),* ensures that as long as the *current_time* is not equal to the defined end time of scenario N, *scenario_loop(N)* keeps being called.

4. Results

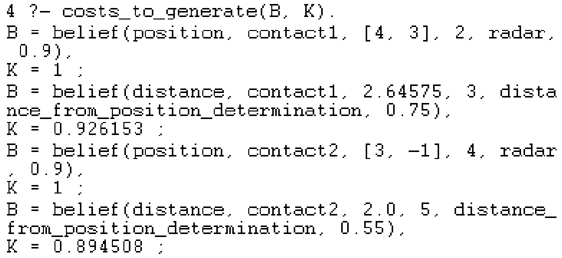
This section presents a scenario from the domain of [3, 4] to demonstrate the results of the various processes within the memory model. The results here are obtained from the Prolog program. The table below shows for each time point 1) the **beliefs** that are newly formed based on input from the environment, and 2) the *beliefs* that the agent wants to form by reasoning.

time	**Sensed** and *Required Beliefs*
1	**belief(position, self, [0,0], 1, gps, 1.0)**
2	**belief(position, contact1, [4, 3], 2, radar, 0.9)**
3	*belief(distance, contact1, _, _, _, _)*
4	**belief(position, contact2, [3, -1], 4, radar, 0.9)**
5	*belief(distance, contact2, _, _, _, _)*
6	*belief(within_weapon_range, contact2, _, _, _, _)*
7	**belief(position, contact1, [4, 2], 7, radar, 0.9)**
8	**belief(position, self, [0,1], 8, gps, 1.0)** *belief(within_weapon_range, contact1, _, _, _, _)*
9	*belief(distance, contact1, _, _, _, _)*
10	*belief(behavior, contact1, _, _, _, _)*
11	**belief(position, contact2, [2, -2], 11, radar, 0.9)**
12	*belief(distance, contact2, _, _, _, _)* **belief(behavior, contact1, receding, 12, officer, 0.6)**
13	*belief(behavior, contact2, _, _, _, _)*
14	*belief(threat, contact1, _, _, _, _)*
15	*belief(threat, contact2, _, _, _, _)*

How the *beliefs* are formed by reasoning depends on the reasoning rules that execute. This paper does not focus on the way in which it is determined which rules may execute, or whether rational or biased rules are selected. For a model that does cover the latter aspects, and can be used in conjunction to the memory model, see [6]. For the current research it is assumed that the selection of reasoning rules is simply done, and that these can either be rational or biased reasoning rules.

To demonstrate that the memory model can support rational and biased reasoning, two scenario versions are developed, named *rational* and *biased*. In the rational version the rational reasoning rules always require complex beliefs are input, which thus become in task focus. Given that the retrieval threshold of complex beliefs is set at a 100, these beliefs are always newly formed. This ensures that always the latest, most correct information is used for reasoning and therefore, this scenario performs rational reasoning.

The total costs of the rational scenario mount to 22.0445, which sums the costs to form the basic beliefs by sensing, standard set at 1 so 7 in total, as well as the basic beliefs from reasoning. The cost of the latter depends on the costs to retrieve the input for the rule that deduces it.



```
4 ?- costs_to_generate(B, K).
B = belief(position, contact1, [4, 3], 2, radar,
 0.9),
K = 1 ;
B = belief(distance, contact1, 2.64575, 3, dista
nce_from_position_determination, 0.75),
K = 0.926153 ;
B = belief(position, contact2, [3, -1], 4, radar
, 0.9),
K = 1 ;
B = belief(distance, contact2, 2.0, 5, distance_
from_position_determination, 0.55),
K = 0.894508 ;
```

The screenshot above shows the costs to form for times 2-5 beliefs concerning contact1 en contact2's **position** and *distance*. Notice that the formation of the *distance* belief is cheaper for contact2 than for contact1. This is the case because of the formation of one of the required beliefs to determine this; the most certain belief about the ships own position taken into account the time passed since the formation of the basic beliefs about this, for which all these basic beliefs are retrieved. Therefore, when later the same belief is required for determining the distance to contact2, its formation is cheaper. This is because the basic beliefs it is based on now have a higher availability value, and are thus easier to retrieve.

The rational reasoning that happens also ensures that when at time 14 the threat of contact1 has to be determined, this is based on the information that that contact is approaching, even though at time 12 an officer says the contact is receding. This is so, because the agent has deduced at time 10 that it is approaching, based on its beliefs about the contact's distances. Because these beliefs are based on beliefs about its positions, which are retrieved from radar and therefore have a high certainty, the belief about the contact approaching also receives a high certainty. The belief stemming from the officer on the other hand, received a lower certainty. Therefore, when at time 14 a complex aggregation reasons about which value of the belief about the behavior of contact it is most certain, it is correctly deduced that the contact is approaching and consequently that it is threatening.

The reasoning rules in the *biased* scenario version do often not require complex beliefs are input, but simple basic beliefs, whose template thus become in task focus. As explained in sections 2.2 and 3, basic beliefs in task focus are attempted to be retrieved from memory. In this version no conscious complex beliefs are formed to reason about a contact's distance. Instead of aggregating the last radar belief about a contact's position, it simply attempts to retrieve any belief about its position. Often this will be the latest, since this is likely to be the moist available. However, this does not need to be, e.g., in the case an earlier belief has been retrieved more often.

Notice that biased rules do sometimes need a complex belief as input, e.g. for the rules that determine a contact's behavior. For that it uses the last, as well as second last belief about its distance. Such beliefs could never be retrieved by a basic belief template, since that only enables the retrieval of the belief with this highest availability and not also the second highest. This problem was exactly the reason to implement the beliefs are they are; time-stamped [3]. However, the aggregation that executes to determine the last and second-last belief in the biased scenario takes abstract beliefs as input, compared to basic beliefs in the rational scenario.

The total costs of the biased scenario mount to 20.1549, which sums the costs to form all the basic beliefs. The costs to form the beliefs at time 3 and 5 about contact1's and contact2's distances are 0.818635 and 0.757405 respectively. This is a bit cheaper than by rational reasoning. However, these costs will diverge much further when multiple basic beliefs exist concerning their, as well as the own position, which are required as input for deducing the distance. Multiple beliefs on the one hand decrease the costs of the retrieval of an abstract belief about them, which due to the multiple presentations has a higher availability. On the other hand they increase the costs of retrieval of the latest radar contact about it, since now more beliefs are retrievable and thus considered, which add to the cost.

As result of the biased reasoning that happens in the second version of the scenario, it is at time 14 simply retrieved what the behavior of contact1 is. Since the belief stemming from an uncertain source is the most available one due to its recentness, it is in this scenario falsely deduced that contact1 is not threatening.

5. Discussion

The memory model described in this paper is specifically designed to allow for both rational and biased reasoning, and can be used to form any type of aggregation. In these aspects it differs from most related work on memory models. Below, a comparison is provided of the presented model with memory storage in general agent architectures and with specific episodic memory models.

Memory storage is an element in all existing (cognitive) agent architectures. In a recent review study on computer-based human behaviour representations [10] it was generalized that "all the (human behaviour) models can represent either short term memory (STM) or long term memory (LTM)." However, the ways in which these memories function differ greatly. For example, ACT-R's STM is formed by a retrieval buffer that can hold one chunk, which it retrieves using an activation function from its declarative memory module (LTM) [2], while Soar's STM is formed by its working memory that is not limited in the number of elements it can hold [9]. Related to the differences in memories, differences exist in the representation of the declarative information entities stored in such modules. These representations range from nodes in a network with an activation value to first-order propositions. The functioning of the various memories are in general fixed and tuned to bring about the behaviour for which the architecture was developed.

Although those existing architectures are not specifically build to handle the abstractions and aggregations as described in this paper, it might be possible to map the specific belief construct to the memory construct of an architecture, provided that the form of the latter has a certain degree of freedom [3, 11]. For further mapping of the proposed memory model, the control mechanism for the formation of aggregations needs to be incorporated in the architecture, as well as the retrieval mechanism of beliefs from LTM into WM.

In addition to the general memory models, there exist specific "episodic memory" models. In [7] a simulation model of episodic memory is developed, which is used for the learning of concepts. Similar to the approach presented in this paper, it is assumed that abstracted knowledge is derived from a pool of episodic traces. The model exhibits basic findings from the schema-abstraction literature, such as differential forgetting of prototypes and old instances. But it does not allow for arbitrary aggregations, controlled by the agent's task.

As already mentioned in the introduction, Nuxoll and Laird [12] also provide a model of episodic memory, which has been implemented in Soar [9]. The model is based on the framework that they present, in which the most important design choices for each of the phases of episodic learning are described. Their framework allows for many possible events that could trigger automatic formation of episodes, and also for decay and removal of elements of the episode. However, in their model the decay has not been implemented yet. Also, there is no abstraction or aggregation in their implementation: all details of an episode are always retrieved.

There are a number of considerations about the implementation of the model presented in this paper that are worth discussing. First, currently all the beliefs that are used to form an aggregation are retrieved into WM. This is done because it is desired that their availability is increased, i.e., they become more easily retrievable afterwards. Although the latter seems a natural choice, it might be more valid to implement a semi-retrieval, which increases their availability without actual retrieving them.

Secondly, the costs of forming a new basic belief by reasoning are currently equal to the costs of the retrieval and aggregation of its inputs. This means that the costs of the reasoning process itself are ignored. It is an implementation decision to take those costs also into account. Related to this, the costs of basic aggregations are also zero in the current implementation, and all basic aggregations happen automatically. Another choice would be to assign different costs to different aggregations. This would allow controlling the formation of basic aggregations by certain factors, e.g., stress: if the stress factor is high, only cheap aggregations are formed. This would implement another type of bias.

In the current stage of the work, the control of the formation and retrieval is parameterized, but the values for the parameters are not yet automatically set. An aim of future work is to extend the current model with a mechanism to automatically derive the values of the control parameters from task and situation aspects.

In addition, stress might be included to allow for agents that exhibit even more human-like reasoning. There are several ways in which a stress parameter might influence the process: first, it can influence which complex beliefs are formed; second, it can influence the reasoning rules that are executed and thus the beliefs that are set in task focus; finally, it could even determine which basic aggregations happen.

References

[1] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An Integrated Theory of the Mind. *Psychological Review,* 111(4):1036–1060, 2004.

[2] J. R. Anderson, and C. Lebiere. *The Atomic Components of Thought.* Lawrence Erlbaum Associates, 1998.

[3] F. Both and A. Heuvelink. From a Formal Cognitive Task Model to an Implemented ACT-R Model. In *Proceedings of the 8th International Conference on Cognitive Modeling*, pages 199–204. Psychology Press, 2007.

[4] A. Heuvelink, and F. Both. A Cognitive Tactical Picture Compilation Agent. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 175–181. IEEE Computer Society Press, 2007.

[5] A. Heuvelink, M. C. A. Klein, and J. Treur. A Formal Approach to Aggregated Belief Formation. In *Proceedings of the 12th International Workshop on Cooperative Information Agents,* to appear.

[6] A. Heuvelink, and J. Treur. Controlling Biases in Demanding Tasks. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society,* to appear.

[7] D. L. Hintzman. 'Schema Abstraction' in a Multiple-Trace Memory Model. *Psychological Review*, 93(4):411–428, 1986.

[8] W. James. *The Principles of Psychology (Vol. 1).* New York: Holt, 1890.

[9] J. E. Laird, A. Newell, and P. S. Rosenbloom. SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[10] J. E. Morrison. A Review of Computer-Based Human Behavior Representations and Their Relation to Military Simulations. Technical Report, P-3845, Institute for Defense Analyses, 2003.

[11] T. J. Muller, A. Heuvelink, and F. Both. Comparison of Implementations of a Cognitive Model in Soar and ACT-R. In *Proceedings of the 6th International Workshop on From Agent Theory to Agent Implementation* (AT2AI-6), 2008.

[12] A. Nuxoll, and J. Laird. A Cognitive Model of Episodic Memory Integrated With a General Cognitive Architecture. In *Proceedings of the 5th International Conference on Cognitive Modeling*, pages 220–225, 2004.

[13] L. J. Schooler, and R. Hertwig. How Forgetting Aids Heuristic Inference. *Psychological Review*, 112(3):610–628, 2005.

[14] E. Tulving, E. Episodic Memory: From mind to brain. *Annual Review of Psychology*, 53:1–25, 2002.

[15] E. Tulving. *Elements of Episodic Memory*. Oxford: Clarendon Press, 1983.

[16] J. Wielemaker. An Overview of the {SWI-Prolog} Programming Environment. In *Proceedings of the 13th International Workshop on Logic Programming Environments*, pages 1–16, 2003.