Building Robots with Analogy-Based Anticipation*

Georgi Petkov, Tchavdar Naydenov, Maurice Grinberg, and Boicho Kokinov

Central and East European Center for Cognitive Science
New Bulgarian University
21 Montevideo Str., Sofia 1618, Bulgaria
gpetkov@cogs.nbu.bg,
tnaydenov@gmail.com,
{mgrinberg,bkokinov}@nbu.bg

Abstract. A new approach to building robots with anticipatory behavior is presented. This approach is based on analogy with a single episode from the past experience of the robot. The AMBR model of analogy-making is used as a basis, but it is extended with new agent-types and new mechanisms that allow anticipation related to analogical transfer. The role of selective attention on retrieval of memory episodes is tested in a series of simulations and demonstrates the context sensitivity of the AMBR model. The results of the simulations clearly demonstrated that endowing robots with analogy-based anticipatory behavior is promising and deserves further investigation.

1 Introduction

Our everyday behavior is based on explicit or implicit employment of predictive models. If we are looking for an object it may simply happen that we see it by chance and go to take it (reactive behavior) or we can imagine where it could possibly be and go to that place to check whether it is there (anticipatory behavior).

IT systems will be closely related to everyday environments in the near future and they will have to perform complicated tasks related to these environments, including searching for lost objects. In order for them to be successful and autonomous, to deal with novel and dynamic environments, to be pro-active and trustworthy in supporting people in their activities, such robots and devices need to have sophisticated cognitive capabilities based on *anticipation*.

Some models, related to this goal, use connectionist networks that generalize from past experience and predict the future on the basis of these generalizations. For example, ALVINN [16] not only reactively responds to the environment but also predicts what will be seen in the next step. The Anticipatory Learning Classifier Systems (ALCs) ([17], [4], [5]) form a class of models that combine reinforcement learning with online generalization and thus propose diverse anticipatory mechanisms. The DYNA-PI systems [18] use reinforcement learning mechanisms to plan on the basis of a model of the world. Planning iteratively generates "chains of predictions" starting from the current state and using the model of the environment.

* This work has been supported by the MIND RACES project funded by the 6th FP of the EC (IST Contract 511931).

C. Freksa, M. Kohlhase, and K. Schill (Eds.): KI 2006, LNAI 4314, pp. 76 – 90, 2007.

Recently these models implement this planner with connectionist networks ([1], [2]). Finally, the architecture of Rodney Brooks [3] proposes a set of layered modules that interact dynamically with each other and with the environment. Thus, "behavior-based robots" based on the concept of the "action circuit" are proposed. However, the architecture is mainly reactive and is not able to anticipate future events.

This paper proposes an alternative approach towards anticipation based on reasoning by analogy. Thus, anticipations are formed not only by capturing the regularities in the world, but by using a single past episode. Sometimes the analogy between the current situation and an episode in memory can be very superficial. For example, the new and the old episodes can share the same set of objects but they can be placed differently, an analogy of this type would be: last time the bone was behind the red block, so the robot may expect the bone to be behind the red object again. However, sometimes the analogy could be much deeper and thus generating non trivial predictions. For example, suppose that in the past episode the bone was behind an object with unique color (the only green object among many red ones). If in the target situation all the objects are having the same color, but there is an object with unique form (the only cube among many cylinders), then the robot can predict by analogy that the bone is now behind this unique object.

Many share our assumption that analogy-making is not only a specific and rare human capability but is fundamental for human cognition [7]. A number of cognitive models exist modeling various parts of the analogy-making process. SMT [6] assumes that only the relations are important and thus the attributes are completely ignored. The analogy is based on mapping identical relations from the target to the base situation. On the contrary, ACME [8], LISA [9], and AMBR [10] allow for mapping of relations with different names. LISA differs from AMBR in the sense that the former is not capable of mapping too complex structures. ACME constructs all possible correspondences and thus the model can not be scaled up. None of the models presented so far has been used for prediction and anticipation. We decided to extend the AMBR model in order to build anticipatory capabilities in robots based on the DUAL cognitive architecture ([11] [15]).

In this paper we present a first exploratory step in this direction based on a simulation of a robot in the Webots environment. We use AMBR as a reasoning core of the system and connect it to the simulated robot and physical environment.

As it became clear from the above discussion, our long-term project is to design robots that are able to demonstrate anticipatory behavior based on analogy. However, in order to explore in detail how analogy can be used in anticipation we started with a relatively simple example based on a simulated house-like environment (see **Fig. 1**a) and a simulated AIBO robot. The simplicity of the task makes it possible to investigate the role of the various mechanisms involved in analogy as assumed by the AMBR model.

The house-like environment consists of several rooms and doors between some of them. In the rooms there are various objects like cubes, balls, cylinders, etc. The goal of the AIBO robot is to find a bone (or bones) hidden behind an object (Figure 1a). In a more complicated task there could be many robots: some of the robots should find and collect some 'treasures', whereas other robots play the role of 'guards' that try to keep the treasures and hide them dynamically or block the way of the treasure-hunters. (**Fig. 1**b).

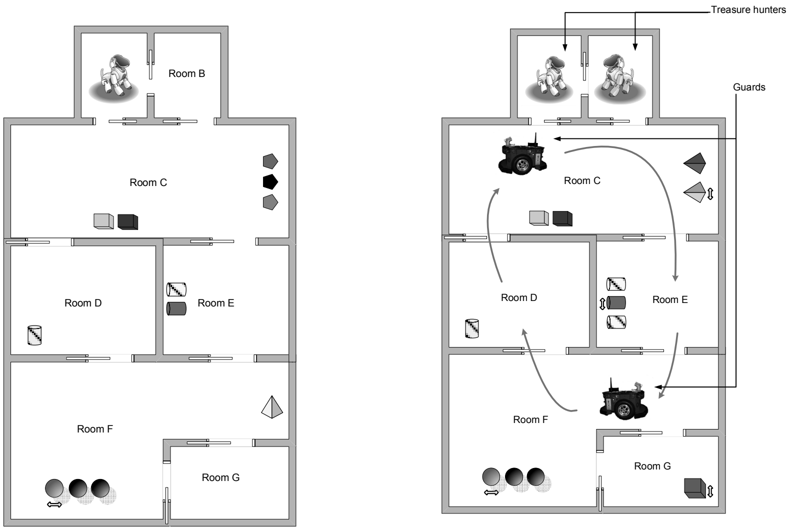

Fig. 1. The micro-domain of a house-like environment. a) The goal of the robot is to find the bone. b) There are 'treasure-hunters' and 'guards' with different goals and strategies.

The strategy of exhaustive search would be very inefficient in real time and sometimes simply impossible especially when the environment is dynamic and changes over time as in the case of Figure 1b. Moreover, we believe that in structured environments like the one shown in Figure 1, anticipation based on analogy will be the most efficient approach, because it will take full advantage of the structure of the environment memorized in previous episodes. A simulated robot and its simulated environment – a room in which there are three cubes and possibly a bone hidden behind one of them – is shown in Figure 2.

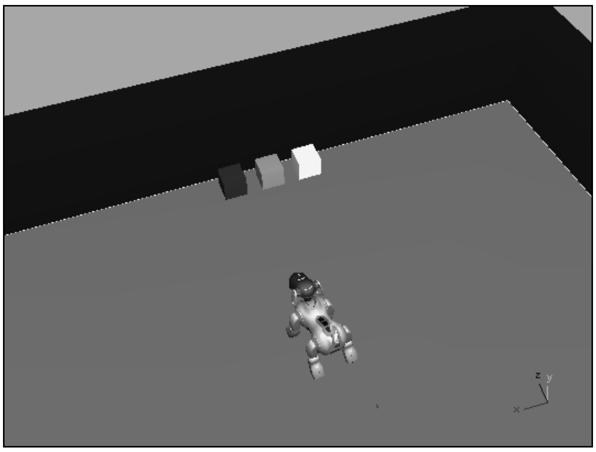

Fig. 2. Robot-environment interaction (real or simulated)

2 The DUAL Architecture

2.1 Basic Properties

DUAL is a cognitive architecture, launched by Kokinov in 1994 ([11], [15]). It consists of memory structures and processing mechanisms, organized around the following principles:

- *Hybridity* – DUAL combines the symbolic with connectionist approaches, by integrating them at the micro-level.
- *Emergent computations* – the global behavior of the architecture emerges from local interactions among a huge number of small entities, called DUAL-agents. There is no central executor that monitors the whole system.
- *Dynamics and context–sensitivity* – The behavior of DUAL changes continuously in response to the influence of the dynamic changes in context. There is no clear-cut boundary between the task and its context. Instead, the context is assumed to be the state of the system in any certain moment, i.e. the pattern of activation over the set of DUAL-agents. This pattern is assumed to reflect the relevance of the various pieces of knowledge in the current context. Some DUAL-agents might be relevant because the corresponding elements are currently perceived from the surrounding environment, others – because they reflect the current goals of the system, and finally, some agents might be relevant because they were recently used and thus they have some residual activation.

2.2 DUAL–Agents

The basic structural and functional element in DUAL is the DUAL–agent. It is hybrid in two ways – it has both connectionists and symbolic aspects, and it serves both as a representational and a functional unit.

Connectionist's Aspect. From the connectionist's perspective, each agent is a node in a localist neural network. It continuously receives activation, updates its current activation level, and spreads it through associative links to other agents. An important feature of DUAL is that it distinguishes the semantic meaning of the agents from their relevance, considering them as independent. The *activation level* is a numeric value that codes the *relevance* of each agent. The pattern of activation does not represent any concept or scheme, but just the current context.

Symbolic Aspect. From the symbolic point of view, DUAL–agents are organized in a semantic network. Each agent 'stands' for something – an object, property, relation, etc. It has its own *micro-frame*. The micro-frames have *slots*, which in turn may have *facets*. There are two kinds of slots – general ones (*G-slots*), and frame-specific ones (*S-slots*).

G-slots have labels, the meaning of which is invariant across the agents, for example *:type*, *:subc*, etc. S-slots also have labels, but their labels are arbitrary, i.e., *:slot1* in one agent may mean something very different from *:slot1* in another agent. S-slots (and only S-slots) have facets, i.e., slots within slots. The same set of labels applies to both G-slots and facets.

Symbolic Processing. DUAL – agents interact with each other. These interactions are relatively simple – they always involve two agents – one of them *sends* some information, and the other one *reads* it.

Each DUAL – agent has a *symbolic processor*. It can receive or construct symbolic structures, transform them, store them in its own *local memory*, and send them to neighboring agents. A typical symbolical transaction involves receiving a symbolic structure, comparing it with the other symbolic structures in the local memory, storing it, transforming it via specific to the agent's type routines, and sending its modification. Each one of these steps is discrete. DUAL – agents manipulate symbols sequentially, one after another, with a frequency that reflects the relevance of the respective agents.

Relationships between Connectionist and Symbolic Processing. All aspects of the agents are merged in a single whole and each one influences the others.

Only a small number of agents whose activation exceeds a certain threshold form the *Working Memory* (WM). The agents that are outside of the WM cannot perform any symbolic operations – they are assumed invisible. For the agents that are involved in the WM, the activation level determines the *speed* of the symbolic processing. Each elementary symbolic operation (namely *read*, *send*, *modify*, etc.) has a 'price' that is paid with activation. Whenever an agent wants to perform such an operation, it begins to accumulate activation in order to pay the required price. Only after it is ready, it can perform the operation. Therefore, the most active agents work rapidly, the less active ones – slowly, and the inactive ones do not work at all. In this manner, the relevance influences the symbolic part of the architecture.

There is also an opposite dependence. The symbolic operations cause new agents to be born, and new connections to be established. These operations change the overall pattern of activation and thus, the symbolic operations influence the pattern of relevance too.

Types of agents. Each DUAL-based model consists of nothing but of DUAL-agents with various types and with various properties. For the purpose of the AMBR-based robots with anticipatory capabilities are used the following types of agents:

Concept–Agents (for short *concepts*) represent classes of homogeneous entities and are organized in a semantic network. They are interconnected via vertical links, for pointing respectively to their super-classes and some of their sub-classes. They are interconnected also horizontally, pointing to some associations and prototypical relations. Note that concepts can stand for objects as well as for relations and abstract terms.

Instance-Agents (*instances*) represent individual entities. Each instance has a G-slot that points to its respective category concept. There are also links in the opposite direction that connect the concepts to some of their instances. Some instances are permanent – they are part of LTM and represent concrete memory traces. Other instances are temporary – they are constructed on the spot because of certain inferences.

Hypothesis-Agents represent possible correspondences between entities. They are temporary agents; they do not participate in LTM; and if they lose their relevance, they disappear. Hypotheses are organized in a constraint satisfaction network. Each hypothesis has its 'life cycle' – it can transform itself from *embryo* to *mature hypothesis*, and then to *winner*. These sub-types reflect the degree to which the hypotheses are novel and attractive.

Anticipation-Agents represent possible entities and relations, predicted by the system. They are also temporary agents. The predictions can be confirmed or rejected by the perceptual systems. In the first case the anticipation agents would mutate into permanent ones, whereas in the second case they would disappear.

Cause-agents represent a special kind of relations. They are equipped with a special procedure that allows them to make decisions whether particular movements to be performed. For example, if the system anticipates that the bone is behind the blue cylinder, then the knowledge transferred from the past situation that movement to the blue cylinder would cause finding the bone would be represented by a cause relation. In turn, the concrete action (in the example – movement to the blue cylinder) would be actually performed.

Action-Agents code procedural knowledge about particular movements. Note that if an action-agent becomes relevant this does not imply that the particular movement would be necessarily performed. The decision should be taken by a cause-agent.

2.3 The Coalitions of Agents

DUAL – agents are very simple and some of the more important properties of the architecture could be observed if looked at from a distant perspective.

DUAL – agents form *coalitions*, i.e., sets of agents, together with the pattern of interactions among them. Coalitions represent more complex entities like propositions or situations. However, the coalitions are not part of the strict computational description of the architecture. Instead, they enhance the conceptual understanding only.

Three important properties of the architecture become visible only at the level of coalitions. The coalitions are *decentralized*, *emergent*, and *dynamic*. None of these properties is presented in the individual agents. The coalitions vary in the intensity of the interactions among their members in comparison to the intensity of interactions with the outside agents. 'Tight' and 'loose' coalitions could be distinguished in respect to this ratio and there is a whole continuum between these two extremes.

Coalitions do not have clear-cut boundaries. Instead, the same agent can participate in two or more coalitions and to a different extent. In the course of time, the coalitions can become 'tighter' or 'looser', can break up, and new coalitions can emerge.

3 The AMBR Model

3.1 Main Ideas

AMBR is a model for analogy making based on DUAL. It treats analogy making as an emergent result of the common work of several overlapping sub-processes – perception, retrieval, mapping, transfer, evaluation, and learning. However, AMBR is a long-term project and unfortunately, at the current stage only the processes of retrieval, mapping, and transfer are modeled and integrated.

AMBR is capable of capturing some similarities between local structures of agents and mapping them, creating *hypotheses for correspondences*. It is a pressure for these mappings to grow, involving other agents. In this way, a Constraint Satisfaction Network is formed. Just as in the process of crystallization, the system strives to a stable equilibrium, changing quantitatively itself. Because the structure–based mappings emerge locally and grow, often some inconsistent hypotheses meet and compete with each other and sometimes blending between episodes occurs.

3.2 Mechanisms Used in AMBR

Spreading Activation. The sources of activation are two special nodes – INPUT and GOAL. Their activation level is always equal to the maximal value. The node INPUT represents the perception of the system, whereas the node GOAL – the current tasks of the system. More sophisticated perceptual and goal-analyzing modules are under construction now. They should replace the INPUT and GOAL nodes. AMBR's work begins when some agents that represent the environment, are attached to INPUT, and some other agents, which represent the task – to GOAL. Various context or priming objects can be attached or removed from INPUT at any moment in time. The activation then spreads through the Long-Term Memory (LTM) network.

The spreading activation mechanism defines the working memory as the set of all agents, which activation level exceeds a certain threshold; determines the speed of the symbolic processes performed by each individual agent; and underlies the relaxation of the constraint satisfaction network.

Marker Passing. Generally speaking, the marker-passing mechanism serves to find out whether a path between two agents is present or not. All symbolic interactions between agents, i.e. exchanging messages, are local and involve only two neighbors. The marker passing mechanism is not an exception, but it allows information about agents to be carried through longer distance as an emergent result.

AMBR marks the instance-agents, which enter in the WM; they in turn mark their respective concepts; then the markers spread to their neighbors that are up in the class hierarchy, and so on.

The main purpose of the mechanism is to justify some semantic similarities between two agents. Whenever two markers cross somewhere, AMBR creates a hypothesis about a correspondence between the two marker origins. The justification for this hypothesis is the fact that these two origins have a common super-class, i.e., they are similar in something. Note, that AMBR makes such inferences only if the whole paths of the markers involve only relevant agents.

Structural Correspondences. Like marker passing, this mechanism creates hypotheses between agents. The difference is that the former is sensitive to semantic justifications, whereas the latter – to propositional ones. There are different kinds of structural correspondences – if two relations are mapped, their arguments should also be mapped; if two instances are mapped, their respective concepts and situations should be mapped.

Because several mechanisms create hypotheses independently, it is possible for some of them to be duplicated, or some of them to be contradictory. A special procedure, attached to each agent monitories the hypotheses in which the respective agent is involved and establishes inhibitory or excitatory links between them.

The Constraint Satisfaction Network. The Constraint Satisfaction Network (CSN) consists of hypotheses for correspondences and is interconnected with the main one. Each hypothesis receives activation from its arguments and from its justifications. It is also inhibited from its competitors (responding to the pressure for one-to-one mapping). Thus, CSN simultaneously reflects the semantic, pragmatic, and structural pressures of the analogy–making task. Due to the CSN, the global behavior of the system emerges from the local interactions between agents. However, it is important to note that no time is spent waiting for the CSN to settle in order to read out the 'solution' from the activation pattern. This allows cognition to be viewed as a continuous process, without breaks between the given tasks.

Rating and Promotion. Because at some moment the system should finish its work, each agent rates its competing hypotheses at regular time intervals. If one of them holds for a long time as a leader, it is promoted to a winner.

Some of the instance-agents (the elements of the target situation) are authorized to use the rating mechanism. The purpose of this mechanism is to monitor all hypotheses that involve the agent and to send *promotion incentives* to those that emerge as stable and unambiguous leaders.

Each authorized instance keeps a data structure, called *rating table*. The *individual ratings* for each registered mature hypothesis are stored in this table. Individual ratings are just a numbers that characterize the relative success of the respective hypothesis – how long, how recently, and how strongly has it been a leader, according to its rivals. The instance-agents periodically (in a fixed time interval and whenever a hypothesis registration request come) adjust the individual ratings. The rating of the current leader increases, whereas those of the other hypotheses decrease. The amount of the change is proportional to the difference between the activation levels of the leader and its closest competitors.

When the individual rating of some hypothesis exceeds a certain threshold, the respective instance-agent sends to it a symbolic structure, called *promotion incentive*. In addition, it eliminates all looser hypotheses. When a hypothesis agent receives such message, it transforms itself to winner hypothesis.

Note, that there is not any central executor that monitors the CSN and decides whether the network is relaxed enough. Instead, some hypotheses become winners locally and in asynchrony. This allows blending between episodes to occur, or unique solutions to be found (of course, sometimes useless solutions are also proposed by the system).

Instantiation. There are two mechanisms for instantiation - skolemization and transfer. The former augments the descriptions of the retrieved episodes based on semantic and structural information. The latter adds new information to the target situation. These mechanisms ensure tolerance towards the lack of information and make the reformulation of the task possible. However, in the current paper we describe a novel instantiation mechanism for creating anticipation-agents.

Imagine the following situation: Let the green cube in the target situation be mapped onto the red cylinder from an old situation in memory. Let the system recalls that the bone was behind the red cylinder. This is a reason to anticipate that now the bone should be behind the green cube. A new instance of the relation 'behind' should be created and should be connected to its arguments – ‘bone’ and ‘green cube’. This will be an example of anticipation-agent.

The exact algorithm for creating anticipation agents is the following: each agent that wins in the mapping competition, checks whether it is an argument of a relation. If yes, the agent informs the relation it is part of. Thus, each relation monitors whether all of its arguments are mapped with winners. If this is the case, the relation decides whether to instantiate a self-copy procedure. If it lacks any promising hypotheses, it starts the process of self-copy. If there are promising hypotheses then the mapping is correct and these hypotheses may be promoted at a later moment, thus there is no need to create a new instance.

The just born agent starts its life cycle as anticipation agent. Some of the anticipation agents serve to direct the attention. For example, the relation ‘same color’, connecting two red cylinders from a base situation can instantiate itself. Thus, a new anticipation-agent would connect the respective correspondences of the two cylinders. It is now the responsibility of the perceptual system to check whether the anticipation is correct or not.

An interesting result emerges, however, when the perceptual system cannot check directly the new prediction, for example, when the relation ‘behind’ is transferred. In such a case, multiple new anticipations emerge. In particular, the causal relation from the base that represents the knowledge “The bone was behind the red cylinder and that caused AIBO to move to it and it successfully found the bone” would instantiate itself using the same mechanism. Analogously, an action-agent “move to the green cube” would also be created.

Finally, the new casual relation would recognize that this certain movement would cause reaching the goal and would activate the corresponding procedural knowledge for actual movement.

3.3 Anticipation by Analogy

We have designed new mechanisms for creation of anticipations. AIBO transfers some relations and objects from the past episodes to the current situation. This transferred knowledge is assumed to be an expectation. Again, following the main principles of the DUAL architectures, all instantiation operations are performed locally and only looking at the system “from above”, the whole pattern of new anticipation-agents could be viewed as a certain anticipated state of the world.

Note, however, that all mechanisms in AMBR work in parallel and thus, the instantiation mechanisms influence other cognitive processes. In particular, we assume that the anticipation-agents play an important role for controlling the attention and in turn the processes of retrieval and mapping.

4 Simulation Results

Suppose the AIBO robot faces several objects in a room. It must predict where (behind which object) the bone is hidden and then to go to the chosen object. Here, we present the simulated version of the real world scenario.

4.1 Mapping Between Close Situations

In the first simulation the mechanisms for transfer and skolemization were switched off in order to check the mapping process. The AIBO robot had four past episodes encoded in its memory (**Fig. 3**).

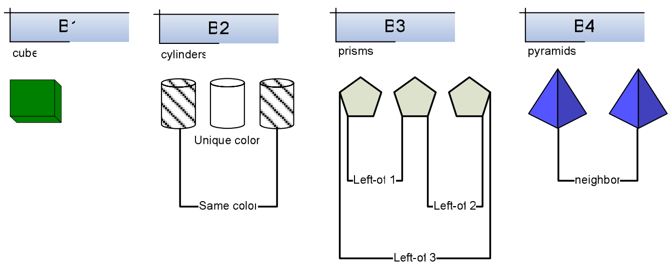

Fig. 3. Old episodes in the memory of the robot, used in the first simulation

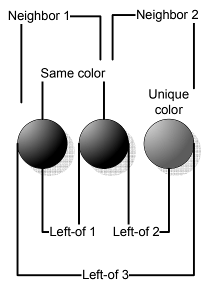

Fig. 4. The target situation, used in the first simulation

The specific number, color and shape of the objects varied across the episodes. The relations between the objects also varied. In all six runs of the simulation one and the same target situation (see **Fig. 4**) was given to the model but different aspects of it were attached to the INPUT node, thus simulating different attention biases. The representation of the target included DUAL-agents representing the objects (OBJECT1, OBJECT2, OBJECT3), as well as some of their properties (BLACK1, BLACK2, GREEN1, BALL1, BALL2, BALL3). Other DUAL-agents represented object properties (e.g. COLOR-OF1, SHAPE-OF1, etc.). Finally, there were relations that involved one or more objects, e.g. LEFT_OF, SAME-COLOR1, NEIGHBOR1, etc. (see **Fig. 4**).

The task of the robot, in this simulation, was to establish a mapping between the target and certain base situation. The target situation (more precisely, some aspects of the target situation) was used for recalling old episodes from memory. These old episodes were gradually and partially retrieved; gradual mappings between various target and base elements emerged; the mapping in turn was influencing the retrieval process. Thus, implicitly the memorized bases competed with each other for the best mapping with the target. It is important to note, however, that all these mappings emerged from local interactions only, without any central mechanism that would calculate the best match among all episodes available in the long term memory.

The results of the simulation are systematized in Table 1.

Table 1. The results from the first situation

Run	Relations at the INPUT	Winner base situation
1	The three balls	B2
2	The three colors	B4
3	The three left-of relations	B3
4	The two neighbor relations and the three colors	B4
5	Only the concept 'green'	B2
6	The relations 'same-color' and 'unique-color'	B2

As shown in Table 1, by varying only the attention bias, the model retrieved different past situations for further analogy making. Note, that this result was achieved by AMBR, testing it in extreme conditions: all old episodes are very similar. They all involve simple geometrical objects and relations that are relatively close semanticaly. Typically, analogy-making models (SME, ACME, LISA, as well as AMBR in previous simulations) are used and tested with dissimilar episodes in memory.

4.2 Single Run of the Model

After all additional mechanisms for instantiation and moving were added, four new bases were designed, including a bone hidden behind a specific object in each of them (see **Fig. 5**). The target consisted only of three red cubes and a specific task to find the bone was also represented and attached to the GOAL list. Again certain properties of the cubes, namely their colors, were highlighted by attaching them to the INPUT node. The simulated AIBO robot was supposed to use the target to retrieve an old episode and to establish mappings between the elements of the two situations. In addition, the position of the bone in the base situation had to be transferred in order to predict the place of the hidden bone in the target situation. Finally, a motor command had to be sent in order to execute the movement. Again, all these operations emerged only from the local interactions of a large number of microagents. Each DUAL-agent just made its specific job with a speed, proportional to its relevance to the task, and the observed behavior was a result of all these small micro-mechanisms simultaneously at play.

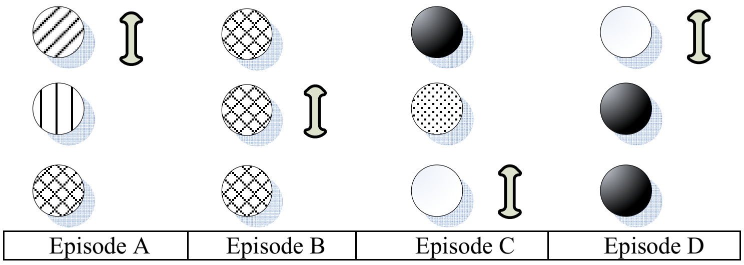

Fig. 5. Old episodes in the memory of the robot used in the second and third simulations

In Figure 6, part of the log from the run of the model is presented:

```
T=52.10, #<PR SAME-COLOR-SIT-003> received in SAME-COLOR-SIT-
003<-->SAME-COLOR-THREEGREEN.
T=56.20, #<PR SIT-SIT-003> received in SIT-SIT-003<-->SIT-
THREEGREEN.
T=56.90, #<PR FIND-SIT-003> received in FIND-SIT-003<-->FIND-
THREEGREEN.
T=61.00, #<PR AIBO-I-SIT-003> received in AIBO-I-SIT-003<--
>AIBO-I-THREEGREEN.
T=69.20, #<PR INITST-SIT-003> received in INITST-SIT-003<--
>INITST-THREEGREEN.
T=69.30, #<PR BONE-SIT-003> received in BONE-SIT-003<-->BONE-
THREEGREEN.
T=119.00, #<PR IN-FRONT-OF-SIT-003> received in IN-FRONT-OF-
SIT-003<-->IN-FRONT-OF-THREEGREEN.
T=290.20, #<PR MIDLE-CUBE-SIT-003> received in MIDLE-CUBE-
SIT-003<-->MIDLE-CYLINDER-THREEGREEN.
Time: 294.600: Just created agent: ANTICIP-MOVE-THREEGREEN-
Time: 295.400: Just created agent: ANTICIP-BEHIND-THREEGREEN-
Time: 303.400: Just created agent: ANTICIP-CAUSE-THREEGREEN-
THE ACTION ANTICIP-MOVE-THREEGREEN- IS EXECUTING!!
T=314.60, #<PR RED-2-SIT-003> received in RED-2-SIT-003<--
>GREEN-2-THREEGREEN.
```

Fig. 6. Part of the script of a single run of the system

The first several lines are reports for establishing winner hypotheses between certain target elements and elements from the four bases. The first winner connected two ‘same-color’ relations – one from the target and one from episode B, named ‘ThreeGreen’ (see **Fig. 5**). The second hypothesis was established between the two situation-agents and thus resulted in additional massive retrieval of all of the elements of situation ‘ThreeGreen’. Soon after, many relations and objects found their corresponding elements (FIND, AIBO, INITIAL-STATE, BONE, IN-FRONT-OF, MIDLE-CUBE). Note that the bone in the past episode was hidden behind the middle cylinder, which has just found its corresponding element. The system did not wait a full mapping to be established. Instead, the instantiation mechanisms immediately begun their work and soon some anticipation agents were created. Thus, gradually, in parallel with the processes of retrieval and mapping, the anticipated state emerged in

AIBO's memory system. Namely, it anticipated that the bone was hidden behind the middle cube (because the current situation was analogical to the retrieved episode of three balls with same color and a bone behind the middle one). Moreover, AIBO inferred that if it would move to the middle cube, it would achieve its goal. As a consequence, the respective action was executed. Note, that the execution of the action did not stop the process. The establishment of additional mappings could continue even after achieving the goal.

4.3 Statistical Results from Many Runs of the System

At the end, a set of eight target situation was designed, varying the shape and the color of the objects involved (Figure 7).

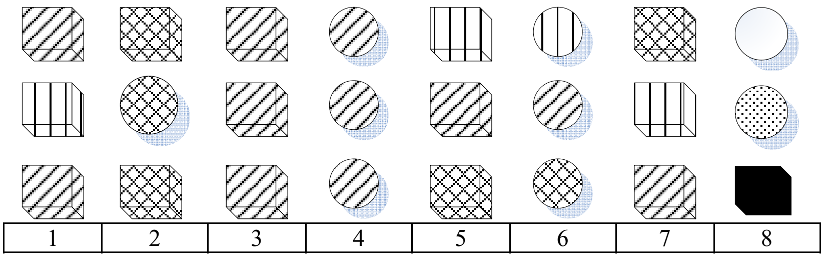

Fig. 7. The eighth target situations used in the third simulation

In order to achieve statistical results, a special tool for creating variants of the knowledge base was used. More precisely, 50 combinations of top-down links from the concepts to the particular instances were randomly created, thus simulating 50 different variants of the core memory. Each of the eight targets was run against each of the knowledge-base variants in two different conditions – focussing the attention of the system on the colors or on the shapes of the objects, respectively. Thus, two distributions of the preferred past episode were achieved (see **Fig. 8**).

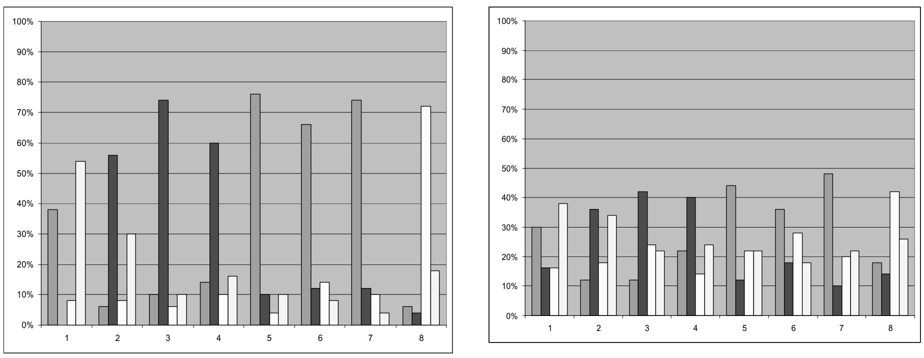

Fig. 8. Comparing the statistical data when the attention was focused on color (left panel) or shape (right panel)

It is shown that when the attention is focused on the shapes only, the base was chosen almost randomly. This is because in all base situations the shapes were equal and thus, the shape was actually irrelevant to the task. AMBR successfully responded to this irrelevance, assuming all bases as almost equally good candidates for analogy.

However, this was not the case, if the color was in the focus of attention.

Clear tendencies to prefer certain bases depending on the target can be observed. At the same time the model was flexible enough, allowing multiple solutions of the problem to be found depending on context (top-down links). These results reconfirm the context-sensitive behavior of AMBR observed previously in other domains [12].

Consider for example the retrieved bases for the second target problem (two green cubes and a green ball in between). The preferred base was B (three green balls) mainly because of the identical colors of all three objects and the fact that attention is concentrated on color. However, quite often an interesting analogy between the second target and the episode D (one white ball and two black balls) was produced. In this case the system detected that the relation `same-form` in the target could be mapped onto the relation `same-color` in the base. The system mapped these two relations and this in turn helped for the retrieval of episode D.

5 Conclusion

A new approach towards building anticipatory agents has been presented, namely building predictions using analogy with a single past episode.

The AMBR model was extended with new agent-types and new mechanisms for building anticipatory-agents based on the existing mechanisms of skolemisation and transfer. These new mechanism were successfully tested in simulation experiments. Additionally, the role of the attention bias for the choice of a base for analogy was explored. Thus, new pressures, which a future fully fledged perceptual system should account for, were defined.

These simulations showed that the newly proposed mechanism can be seriously considered as an important part of any advanced cognitive system with anticipatory capabilities.

The presented work is just a small step in a long-term project. Designing perceptual capabilities would allow for automatic encoding of new situations and thus, the behavior of real robots instead of simulated ones could be tested. Bottom-up and top-down mechanisms for locating the attention should be carefully modeled and their influence on the robot's behavior further investigated.

References

1. Baldassarre G. (2002). Planning with Neural Networks and Reinforcement Learning. PhD Thesis. Colchester - UK: Computer Science Department, University of Essex.
2. Baldassarre G. (2003). Forward and Bidirectional Planning Based on Reinforcement Learning and Neural Networks in a Simulated Robot. In Butz M., Sigaud O., Gérard P. (Eds.), Adaptive Behaviour in Anticipatory Learning Systems, Springer Verlag, Berlin, Heidelberg, pp. 179-200.

3. Brooks, R.A., (1991). How to build complete creatures rather than isolated cognitive simulators, in K. VanLehn (ed.), Architectures for Intelligence, pp. 225-239, Lawrence Erlbaum Assosiates, Hillsdale, NJ.
4. Butz, M. V., Goldberg, D. E., & Stolzmann, W. (2002). The anticipatory classifier system and genetic generalization. Natural Computing, 1, pp. 427-467.
5. Butz, M.V., & Goldberg, D.E. (2003). Generalized state values in an anticipatory learning classifier system. In Butz M., Sigaud O., Gérard P. (Eds.), Anticipatory behavior in adaptive learning systems, Springer Verlag, Berlin, Heidelberg, Germany, pp. 282-301.
6. Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7, 155-170.
7. Hofstadter, D. R. (1995). Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought, NY: Basic Books.
8. Holyoak K. & Thagard P. (1989). Analogical mapping by constraint satisfaction. CognitiveScience, 13, 295-355.
9. Hummel, J. & Holyoak, K. (1997). Distributed representation of structure: A theory of analogical access and mapping. Psychological Review, 104, 427-466.
10. Kokinov, B. (1994a). A hybrid model of reasoning by analogy. In K. Holyoak & J. Barnden (Eds.), Advances in connectionist and neural computation theory: Vol. 2. Analogical connections (pp. 247-318). Norwood, NJ: Ablex
11. Kokinov, B. (1994b). The DUAL cognitive architecture: A hybrid multi-agent approach. Proceedings of the Eleventh European Conference of Artificial Intelligence (ECAI-94). London: John Wiley & Sons, Ltd.
12. Kokinov, B., Grinberg, M. (2001). Simulating Context Effects in Problem Solving with AMBR. In: Akman, V., Thomason, R., Bouquet, P. (eds.) Modeling and Using Context. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol. 1775, Springer Verlag.
13. Kokinov, B. Petrov, A. (2000). Dynamic Extension of Episode Representation in Analogy-Making in AMBR. In: Proceedings of the 22nd Annual Conference of the Cognitive Science Society. Erlbaum, Hillsdale, NJ.
14. Kokinov, B., Petrov, A. (2001). Integration of Memory and Reasoning in Analogy-Making: The AMBR Model. In: Gentner, D., Holyoak, K., Kokinov, B. (eds.). The Analogical Mind: Perspectives from Cognitive Science, Cambridge, MA: MIT Press
15. Petrov, A. & Kokinov, B. (1999). Processing symbols at variable speed in DUAL: Connectionist activation as power supply. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99). San Francisco, CA: Morgan Kaufman, p. 846-851.
16. Pomerleau, D. (1989). "ALVINN: An Autonomous Land Vehicle. In a Neural Network," Advances in Neural Information Processing Systems 1, Morgan Kaufmann
17. Stolzmann, W. (1998). Anticipatory classifier systems. Genetic Programming 1998: Proceedings of the Third Annual Conference, 658-664.
18. Sutton, R.S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Proceeding of the Seventh International Conference on Machine Learning, pp. 216-224. San Mateo, Ca.: Morgan Kaufmann.