Affect, Anticipation and Adaptation: Affect-Controlled Selection of Anticipatory Simulation in Artificial Adaptive Agents.

Joost Broekens,

Walter A. Kosters,

Fons J. Verbeek.

Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands.

Correspondence to:

Joost Broekens

Niels Bohrweg 1

2333CA, Leiden

The Netherlands

Phone: +31 (0)71-5275779

Fax: +31(0)71-5276985

Email: broekens@liacs.nl

Abstract

Emotion plays an important role in thinking. In this paper we study affective control of the amount of simulated anticipatory behavior in adaptive agents using a computational model. Our approach is based on model-based reinforcement learning (RL) and inspired by the *Simulation Hypothesis* (Cotterill, 2001; Hesslow, 2002). The simulation hypothesis states that thinking is internal simulation of behavior using the same sensory-motor systems as those used for overt behavior. Here, we study *the adaptiveness of an artificial agent, when action-selection bias is induced by an affect-controlled amount of simulated anticipatory behavior*. To this end, we introduce an affect-controlled *simulation-selection* mechanism that uses the predictions of the agent's RL model to select anticipatory behaviors for simulation. Based on experiments with adaptive agents in two nondeterministic partially observable gridworlds we conclude that (1) internal simulation has an adaptive benefit and (2) affective control can reduce the amount of simulation needed for this benefit. This is specifically the case if the following relation holds: positive affect decreases the amount of simulation towards simulating the best potential next action, while negative affect increases the amount of simulation towards simulating all potential next actions. In essence we use artificial affect to control *mental* exploration versus exploitations. Thus, agents "feeling positive" can think ahead in a narrow sense and free up working memory resources, while agents "feeling negative" must think ahead in a broad sense and maximize usage of working memory. Our results are consistent with several psychological findings on the relation between affect and learning, and contribute to answering the question of *when* positive versus negative affect is useful during adaptation.

Keywords: affect, action selection, *anticipatory simulation, simulation selection, working memory, simulated adaptive agents.*

1 Introduction

Emotion plays an important role in thinking. Evidence ranging from philosophy (Griffith, 1999) through cognitive psychology (Frijda, Manstead & Bem, 2000) to cognitive neuroscience (Damasio, 1994; Davidson, 2000) and behavioral neuroscience (Berridge, 2003; Rolls, 2000) shows that emotion is both constructive and destructive for a wide variety of cognitive phenomena. Normal emotional functioning appears to be necessary for normal cognition.

Emotion influences thought and behavior in many ways. Emotion in general is related to the urge to act (e.g., Frijda & Mesquita, 2000), influences how we evaluate stimuli, and what potential next actions we consider (e.g., Damasio, 1996). Specific emotions trigger specific behaviors (e.g., fight or flight). Emotion influences information processing in humans; positive affect facilitates top down, "big-picture" heuristic processing while negative affect facilitates bottom up, "stimulus analysis" oriented processing (Ashby, Isen & Turken, 1999; Forgas, 2000; Phaf & Rotteveel, 2005).

In this paper we specifically focus on the influence of affect on learning. Affect and emotion are concepts that lack a single concise definition, instead there are many (Picard *et al.*, 2004). In general, the term emotion refers to a set of in animals naturally occurring phenomena including motivation, emotional actions such as fight or flight behavior and a tendency to act. In most social animals facial expressions are also included in this set of phenomena, and so are—at least in humans—feelings and cognitive appraisal (see, e.g., Scherer, 2001). A particular emotional state is the activation of a set of instances of these phenomena, e.g., *angry* involves a tendency to fight, a typical facial expression, a typical negative feeling, etc. *Time* is another important aspect in this context. A short-term (intense, object directed) emotional state is often called an *emotion*; while a longer term (less intense, non-object-directed) emotional state is referred to as *mood*. The *direction* of the emotional state, either positive or negative, is referred to as *affect* (e.g., Russell, 2003). Affect is often differentiated into two orthogonal (independent) variables: *valence*, a.k.a. pleasure, and *arousal* (Dreisback & Goschke, 2004; Russell, 2003). Valence refers to the positive versus negative aspect of an emotional state. Arousal refers to an organism's level of activation during that state, i.e., physical readiness.

We use *affect* to denote the *positiveness* versus *negativeness* of a situation. In this study we ignore the arousal a certain situation might bring. As such, positive affect characterizes a situation as good, while negative affect characterizes that situation as bad (e.g., Russell, 2003). Further, we use affect to refer to the mid- to long-term timescale: i.e., to mood.

Several psychological studies support that enhanced learning is related to positive affect (Dreisbach & Goschke, 2004). Others show that enhanced learning is related to negative affect (Rose, Futterweit & Jankowski, 1999). Although much research is currently being carried out, it is not yet clear how affect is related to learning in detail. Therefore we have set up a computational modeling study. Here we study affective control of the amount of information processing in artificial adaptive agents; we use affect as meta-learning parameter (Doya, 2002). We do not model categories of emotions nor use emotions as information in symbolic-like reasoning.

In order to simulate affective control of information processing, we propose a measure for artificial affect that relates to an adaptive agent's relative performance on a learning task. As such, artificial affect measures how well the agent improves. Our adaptive agent learns by reinforcement; reward and punishment. Thus, in our case, “how well” is defined by the average reinforcement signal. Therefore, the agent’s performance is defined by the difference between the long-term average reinforcement signal (“what am I used to”) and the short-term average reinforcement signal (“how am I doing now”) (cf. Schweighofer & Doya, 2003). Our artificial affect thus relates to natural affect: it characterizes the situation of the agent on a scale from good to bad. Our measurement relates more to mood than emotion, as it is based on average reinforcement signals (see Section 4.3 and 7.1).

We have developed a variation to the model-based Reinforcement Learning (RL) paradigm (Sutton & Barto, 1998). This variation enables the study of information processing in light of the *simulation hypothesis* (Cotterill, 2001; Hesslow, 2002). The simulation hypothesis states that thinking is internal simulation of behavior using the same sensory-motor systems as those used for overt behavior (Hesslow, 2002). The main reason for adopting the simulation hypothesis is that it argues for evolutionary continuity between agents that consciously think and agents that do not. We believe this is a critical aspect in studying behavior, emotions, consciousness and cognition. In this paper, we refer to simulation as described by the simulation hypothesis.

Currently, an important issue is how simulation of interaction is integrated with real interaction while using the same mechanisms (see, models by, e.g., Shanahan, 2006; van Dartel & Postma, 2005; Ziemke, Jirenhed & Hesslow, 2005). Our agents are able to internally simulate anticipatory behavior using their RL model. The agent thinks ahead by selecting one or more potential next action-state pairs for internal simulation. This action state and its associated value are fed into the RL model as if these were actually observed. This introduces a bias to predicted values. Our action-selection mechanism uses these biased values to select the agent's next action. Subsequently, the values are reset to the original values before simulation. Thus, internal simulation temporarily biases the predicted values in the RL model, thereby biasing action selection.

We report on a study on *the adaptiveness of an artificial agent, when action-selection bias is induced by an affect-controlled amount of simulated anticipatory behavior*. The main contributions of this paper to the affect and learning and simulation hypothesis literature are:

1.) The introduction of an affect-controlled mechanism for the selection of internally simulated behavior instead of actual behavior; we define this mechanisms as *simulation selection*.

2.) An investigation of the influence on learning, if affect is used to control the amount of internally simulated interactions, where simulated interactions bias actual action selection. As we use internal simulation as a model for information processing, we investigate affect as a modulator for the distribution of internal versus external information processing effort (Aylett, 2006).

2 Emotion and Affect

In this section we present the rationale for the concept of emotion used, that is, positive and negative affect. We first review different views on the interplay between emotion and cognition, after which we present evidence that affect influences learning, the main phenomenon investigated computationally in this paper.

2.1 Emotion, Thought and Behavior

Emotion influences thought and behavior. At the neurological level, malfunction of certain brain areas not only destroys or diminishes the capacity to have (or express) certain emotions, but also has a similar effect on the capacity to make sound decisions (Damasio, 1994) as well as on the capacity to learn new behavior (Berridge, 2003). These findings indicate that these brain areas are linked to emotions as well as to "classical" cognitive and instrumental learning phenomena. At the level of cognition, a person's belief about something is updated according to the emotion: the current emotion is used as information about the perceived object (Clore & Gasper, 2000; Forgas, 2000), and emotion is used to make the belief resistant to change (Frijda & Mesquita, 2000). Ergo, emotions are "at the heart of what beliefs are about" (Frijda *et al*., 2000).

Emotion is related to the regulation of behavior. Emotions can be defined as states elicited by rewards and punishments (Rolls, 2000). Behavioral evidence suggests that the ability to have sensations of pleasure and pain is strongly connected to basic mechanisms of learning and decision making (Berridge, 2003; Cohen & Blum, 2002). These studies directly relate emotion to reinforcement learning. Behavioral neuroscience teaches us that positive emotions reinforce behavior while negative emotions extinguish behavior. At this level, emotion has a direct—mostly associative—effect, though other effects are reported (Dayan & Balleine, 2002).

At the level of cognition, emotion plays a role in the regulation of the amount of information processing. For instance, Scherer (2001) argues that emotion is instrumental in allocating resources to process stimuli. Furthermore, in the work of Forgas (2000) the relation between emotion and information processing strategy is made explicit: the influence of mood on thinking depends on the strategy used.

To summarize, emotion can be produced by low-level mechanisms of reward and punishment, and can influence further information processing. As affect is a useful abstraction of emotion, these aspects inspired us to study (1) how artificial affect can result from an artificial adaptive agent's reinforcement signal (Section 4.3), and (2) subsequently influence information processing in a way compatible with the psychological literature on affect and learning. In the next subsection we present some of the psychological findings related to the latter.

2.2 Learning is Influenced by Positive and Negative Affect

The influence of affect on learning is typically studied with psychological experiments. Take two groups, one control group and one experimental condition group. Induce affect (positive or negative) into the subjects belonging to the experimental condition group by showing them unanticipated pleasant images or giving them small unanticipated rewards, or violent, ugly images and punishment if negative affect is to be induced in the subject. Measure the subjects' affect. Let the two groups do a cognitive task. Finally, compare the performance results between both groups. If the experimental condition group performs better, the induction is assumed to be responsible for this effect, ergo: affect influences the execution of the cognitive task.

We focus on the influence of affect on learning. Some studies find that negative affect enhances learning. For instance, Rose, Futterweit and Jankowski (1999) found that when babies aged 7 - 9 months were measured on an attention and learning task, negative affect correlated with faster learning. Attention mediated this influence. Negative affect related to more diverse attention, i.e., the babies' attention was "exploratory", and both negative affect and diverse attention related to faster learning. Positive affect resulted in the opposite. This relation suggests that positive affect relates to exploitation and negative affect relates to exploration, a notion also supported by von Hecker and Meiser (2005) who state that attention is more evenly spread when in a negative mood.

Interestingly, other studies suggest an inverse relation. For instance, Dreisbach and Goschke (2004) found that mild increases in positive affect related to more flexible behavior but also to more distractible behavior. The authors used an attention task, in which human subjects had to switch between two different "button press" tasks. In such tasks a subject has to repeatedly choose to press one out of two different buttons, based on some criteria in a complex stimulus. After some trials, the task is switched, by changing several stimulus characteristics. The authors measured the average reaction time of the subjects' button press just before and just after the task switch. They found that increased positive but not neutral or increased negative affect relates to decreased task switch cost, as measured by the difference between pre-switch reaction time and post-switch reaction time. So, it seems that in this study positive affect facilitated a form of exploration, as it helped to remove the bias towards solving the old task when the new task had to be solved instead.

Combined, these results suggest that different affective states can help learning but perhaps at different phases during the process (Craig, Graesser, Sullins and Gholson, 2004). Our paper addresses exactly this issue. We investigate the relation between affect, the amount of internal simulation, and learning performance. We define a measure for artificial affect and use this measure to control the amount of internally simulated anticipatory behavior of an adaptive agent. Artificial affect thus controls how many thoughts the adaptive agent is allowed to have at a certain moment. Internally simulated actions influence action selection by temporally adding values to potential next actions. Internal simulation thus temporally favors certain actions while disfavoring others. Action selection on its turn influences learning performance. We test three different hypotheses about what assists learning: (1) positive affect decreases the amount of internal simulation and negative affect increases this amount, (2) negative affect decreases the amount of internal simulation and positive affect increases this amount, and (3) high intensity of affect increases the amount of simulation and low intensity decreases this amount.

3 Internal Simulation of Behavior as a Model for Thought

Our approach towards anticipatory simulation is inspired by the simulation hypothesis stating that conscious thought consists of "simulated interaction with the environment" (Hesslow, 2002). Thoughts consist of internally simulated chains of interaction with the environment and evaluation of those simulated interactions. As such, thoughts are virtual versions of real interactions. For this to be possible, a brain must be able to simulate actions, perceptions and evaluations internally . That is, the brain has to simulate potential interaction with the environment while simultaneously controlling the body such that it is able to successfully interact with the environment. Hesslow (2002) and Cotterill (2001) provide extensive evidence for the biological and psychological plausibility of such a process of internal simulation.

3.1 Thought and Internal Simulation of Interaction

Internal simulation of behavior is also a convenient model for thought, especially in the context of adaptive behavior and evolutionary continuity. First, if an agent is able to internally

simulate a certain interaction, this simulation can reactivate the value of that interaction and thereby (1) influence decision making with predictions based previous experiences and, (2) enhance learning by propagating the value of that interaction to other related interactions. Second, the simulation hypothesis is said to provide a bridge between species that consciously think and those that do not (Hesslow, 2002): no fundamentally different additional mechanisms are needed for thought, apart from those that enable off-line simulation of interaction.

Recently, strong evidence for a link between internal simulation, adaptive behavior and evolutionary continuity has been presented. Foster and Wilson (2006) showed that awake mice replay in reverse order behavioral sequences that led to a food location; a finding crucial for the above mentioned link. First, it suggests that mice are able to internally simulate interaction with the environment, showing that simulation mechanisms need not be restricted to humans. This supports the possibility of evolutionary continuity of the human thought process. Second, internally replaying a sequence of interactions can potentially increase learning in mice in the same way as *eligibility traces* can enhance learning in reinforcement learning (Foster & Wilson, 2006). An eligibility trace (see Sutton & Barto, 1996) can be seen as a sequence of recent interactions with the environment. Delayed reinforcement is distributed over all the interactions stored in the trace. This mechanism can dramatically increase learning performance of simulated adaptive agents, and therefore provides a plausible argument for an immediate benefit of internal simulation (different from benefits related to complex cognitive abilities such as planning).

3.2 Working Memory, Simulation Selection and Internal Simulation of Behavior

If a thought is an internally simulated interaction, and working memory (WM) contains the thoughts of which we are consciously aware, then WM contains a set of currently maintained internally simulated interactions—specifically the episodic buffer that is a multi-modal limited-capacity storage buffer (Baddeley, 2000). Further, for a specific thought to enter WM, it is often assumed that the thought has to be active above a certain threshold (see, e.g., Deheane, Sergent & Changeux, 2003).

In the “internal simulation thought process”, an agent in a specific situation starts to pay attention to several situational aspects. These aspects start entering the central executive of working

memory (Baddeley, 2000) and are thereby above threshold. Now, the central executive pushes a multi-modal simulation of future (or related) interactions from long-term memory to the episodic buffer, where it is maintained. As the episodic buffer has limited capacity, the interaction can reside in the buffer until being replaced by new simulated interactions. Thus, filling the buffer depends, among other things, on how critical the filter (central executive) is in passing information to the buffer. The episodic buffer is filled with those internally simulated interactions that are attended to with sufficient intensity. Therefore, the higher the selection threshold, the smaller the amount of internally simulated behaviors maintained in the episodic buffer.

Interestingly, if thought is internal simulation of behavior using the same sensory-motor mechanisms as real behavior, then the selection of those thoughts should resemble the selection of behaviors. Action selection has been defined as the problem of continuously deciding what action to select next in order to optimize survival (Tyrell, 1993). "Thought selection", to which we refer as *simulation selection*, can therefore be defined in a similar way. Simulation selection is the problem of continuously selecting behaviors for internal simulation such that action selection is assisted, not hindered. The latter is critical as, according to the simulation hypothesis, action selection and simulation selection should be tightly coupled: both use the same mechanisms. Errors in simulation selection can directly influence action selection and thereby be responsible for actions that are erroneous too. In our computational model we introduce a simulation-selection component based on precisely these principles. The selection threshold in our model is dynamically controlled by artificial affect (Section 4.2, 4.3).

4 Model

In this section we explain the computational model used to study the main question. We use adaptive agent based modeling. Our agents "live" in gridworlds. Figure 1 shows the overall architecture of our computational approach.

The affect mechanism calculates artificial affect based on how well the agent is doing compared to what it is used to. The simulation-selection mechanism selects next interactions for simulation, using a threshold controlled by artificial affect. The threshold filters which potential next

interactions are simulated and which not. Selected interactions are fed into the RL model as if they were real. This biases predicted values of states in the RL model. The action-selection mechanism selects an action based on these biased values using a greedy algorithm. The action is executed, and the agent perceives the next state. Our approach is related to *Dyna* (Sutton, 1990); see also Section 7.

First we discuss the components of the model and how it learns using RL principles. Next we explain how we have implemented the simulation hypothesis on top of our model. Subsequently we explain how we model artificial affect and how this is used to control the amount of internal simulation the agent uses to bias the predicted values employed by its action-selection mechanism. Finally, we explain how the action-selection mechanism integrates everything.

(Figure 1 about here)

4.1 Hierarchical State Reinforcement Learning (HS-RL): A Variation of Model-Based RL

Our model is a combined forward (predictor) and inverse (controller) model for learning agent behavior (Demiris & Johnson, 2003). The model learns to predict the next state given the current state and an action, enabling forward simulation of interaction. At the same time it learns to predict the values for potential next actions, enabling agent control. Basically, the agent's memory structure is a directed graph that is learned by interaction with the environment. Two types of nodes exist: (1) nodes that encode $<a, s>$ tuples, where s is an observed state and a the action leading to that state, and (2) nodes that encode $<h_l, a', s'>$ tuples, to which we refer as *interactrons*. Here, h_l is a history of observed action-state pair transitions $<a^{t-l}, s^{t-l}><a^{t-l+1}, s^{t-l+1}>\ldots<a^{t-1}, s^{t-1}>$ with l the history length not greater than a maximum length k, and $<a', s'> = <a^t, s^t>$ the action-state pair predicted by history h_l at time t. The existence of type 1 nodes depends on the states experienced by the agent. The existence of interactrons (type 2 nodes) and the connectivity between type 1 nodes and interactrons depend on observed transitions from $<a, s>$ to $<a', s'>$. Thus, the memory is initially empty and is constructed while the agent interacts with its environment; our agent learns online; we assume *certainty equivalence*. This is closer to real life than a forced separation between exploration and exploitation phases, even though the model might be highly suboptimal at the start (Kaelbling, Littman, & Moore, 1996).

The model is constructed as follows. The agent selects an action, $a \in A$, from its set of potential actions, A, using the action-selection mechanism (Section 4.4). It executes the action and perceives the result, s. A type 1 node $<a, s>$ is created *if and only if there does not exist such a node*. Consider, for example, an agent that has chosen some action $\hat{a}$ and experiences some state $\hat{s}$. Because its model does not yet contain a node that represents $< \hat{a}, \hat{s}>$ it is created (e.g., s_1 in Figure 2a). Note that we use s_i (indexed) to refer to $<a, s>$ tuples (type 1 nodes) instead of s to refer to observed states. Now the agent selects and executes a new action, resulting in a new situation $s_2 = < \hat{a}', \hat{s}'>$, giving a new node that represents s_2 (Figure 2b). To model that s_2 follows s_1 (s_1 predicts s_2), the previous situation, s_1, is now connected to the current situation, s_2, by creating an interactron that is connected to s_1 and s_2 with edges as shown in Figure 2c. This interactron I_1 thus encodes $<h_1, s_2>$ with h_1 being the history of length 1 before the transition to action-state pair s_2,, in our example $h_1 = s_1$. This process continues while exploring and the process is applied hierarchically to all active nodes. A type 1 node is active if the current situation $<a^t, s^t>$ equals the $<a, s>$ tuple encoded by that node. An interactron $<h_l, a', s'>$ is active if and only if h_l equals the most recent observed history $<a^{t-l}, s^{t-l}><a^{t-l+1}, s^{t-l+1}>\ldots<a^{t-1}, s^{t-1}>$ and the prediction $<a', s'>$ equals $<a^t, s^t>$. For example, node I_1 and s_2 in Figure 2c are active. An additional example is presented in Figure 2d and 2e. If situation s_2 is followed by a new situation s_3, the resulting memory structure is shown in Figure 2d, with active nodes s_3, I_2 and I_3. If, on the other hand s_2 is followed by s_1, the resulting structure is shown in Figure 2e, with active nodes s_1, I_2 and I_3. Note that the maximum length of a history encoded by a node is bounded by k, therefore the maximum number of active interactrons is k (for computational reasons $k = 10$, Broekens & DeGroot, 2004; see also below).

(Figure 2 about here)

Every interactron $<h_l, a', s'>$, has three properties r, v, and υ, with r the reward and v the value (a.k.a. Q-value) of the tuple $<h_l, a', s'>$, and finally υ is a statistic for the transition probability between h_l and $<a', s'>$. Note that from here on we use the term *reward* and *reinforcement* to refer to any reinforcement: positive, negative or zero. If at a later time the sequence of situations $h_l s_i$ is *again* observed by the agent, then the statistic υ of the interactron encoding the tuple $<h_l, s_i>$ is

incremented— υ is a counter that is initially zero and represents the *usage* of an interactron. Thus, υ can be used to calculate the transition probability $p(s_i \mid h_l)$ using the following more generic formula:

$$p(x \mid y) = \upsilon_x \Big/ \sum_{i=1}^{|X_y|} \upsilon_{x_i} , \qquad (1)$$

where y is an interactron encoding $<h_{l-1}, a, s>$ with $h_l = h_{l-1}s_y$ and $s_y = <a, s>$, and $x \in X_y$. Here $X_y = \{x_1, \ldots, x_n\}$ is the set of interactron nodes that encode $<h, a', s'>$ tuples and are predicted by y, x is the interactron $<h_l, s_i>$ of which we want to know the transition probability $p(s_i \mid h_l)$, and υ_x and υ_{x_i} are the counters belonging to x and x_i respectively. This function calculates the conditional probability of observing an action-state pair $<a, s>$ (interactron x) after having observed a history of action-state pairs h_l (interactron y). For clarity: y refers to an active interactron that represents the current state of affairs (and, as mentioned earlier, maximally k of such y's can be active at one moment in time each representing the current state with a different history length), while x refers to a particular predicted next state at t+1, assuming y, and x_i refers to all other predicted next states assuming that same y.

We define a global threshold, θ, representing the minimal "survival probability" for an interactron. If $p(x \mid y) < \theta$, the corresponding interactron x is forgotten and removed from memory, including all of its predictions. In this manner the stability of an agent's long-term memory is modeled, and it corresponds to Bickhard's (2000) notion of interaction (de)stability based on consistent confirmation of predicted interactions (Broekens & DeGroot, 2004). In our experiments we use θ to vary the speed with which the agent forgets knowledge.

To learn based on reinforcement, every interactron has a value v, with:

$$v = r + \gamma \cdot v_{next} \text{ , } \textit{with v bounded by } \min(r, v_{next}) \le v \le \max(r, v_{next}) \qquad (2)$$

where r is the learned reward for a certain interactron, γ the discount rate (γ = 1.0, see comments below) and v_{next} is a back-propagated value from next predicted future states. As multiple nodes can be active at the same time, these nodes learn simultaneously. Several steps are involved. First, all k active interactrons are reinforced by a signal from the environment, r_t, at time t. For every such interactron y, $r(y)$ is adapted according to the formula:

$$r(y)^{t+1} = r(y)^t + \alpha(r_t - r(y)^t) , \qquad (3a)$$

where α is the agent's learning rate. Second, for every interactron y, $v_{next}(y)$ is calculated as follows:

$$v_{next}(y)^{t+1} = \sum_{i=1}^{|X_y|} v(x_i \mid y)^t \times p(x_i \mid y)^t \ , \qquad (3b)$$

where $v(x_i \mid y)^t$ is defined as the value of interactron x_i, with x_i predicted by y. This indirect part of an interactron's value is thus the weighted average of the values belonging to the interactrons X_y that represent the situations that y predicts, where the weighting is according to the probabilities $p(x_i \mid y)^t$ at time t over all i. Note that only *active* nodes y are updated, i.e., we use lazy propagation whereby only the interactrons predicted for t+1 (the x's) are used to update the active interactrons at time t (the y's).

In an agent control setting, the model can be summarized as follows. At every step, all active interactrons predict potential next situations, at most k of these interactrons can be active, and the 1st to k^{th} interactron predicts potential next action-state pairs <a', s'> using a history of length 1 to k respectively (e.g., I_3 is a k=2 interactron with history s_1s_2). As such, this memory learns 1st…k^{th} order Markov Decision Processes (MDPs) in parallel. This property enables it to cope with partially observable worlds in which the partial observability can be resolved using at most a history of length k. At most k MDPs are active at the same time, each predicting values for action-state pairs based on a different history length. Action selection integrates not over the predictions of one MDP but over the predictions of at most k MDPs (see Section 4.4). Note that our model underuses the Markov property, as it keeps track of, and constructs nodes for, all history up to k steps back *all the time*, *not only when a certain history is actually needed to solve the partial observability of the world*. For an interesting approach that relates to ours and that proposes some solutions for better using the Markov property see McCallum's (1995) *utile suffix memory*.

An important difference between our approach and many other model-based RL approaches is that our MDPs have a maximal length of k steps and nodes only propagate values to their own history. On the one hand this is a benefit in that reward/value propagation is never cyclic. Values are propagated back through multiple, partly overlapping k-finite MDPs. This makes our model particularly robust in cyclic learning tasks (even for cycles smaller than k steps): our world model forces values to propagate from a well defined end with a long history to a well defined beginning

with no history, the values are *not* recursive. As a result, in our model the discount factor can be equal to 1.0. On the other hand this characteristic also poses a problem, as values further than k steps away cannot be propagated back, resulting in the need for regular reward intervals. This could be resolved (at the expense of cyclic-task robustness) by allowing values to propagate *not only* to nodes encoding for a shorter history at the previous timestep but *also* to nodes encoding for a history of equal length at the previous timestep, effectively making values recursively defined. That is, a node $s_1h_{l-1}s_t$ encoding for a situation s_t with a history s_1h_{l-1} of length l not only propagates its value to a node $s_1h_{l-2}s_{t-1}$ with $h_{l-1}=h_{l-2}s_{t-1}$, but also to a node $s_0s_1h_{l-2}s_{t-1}$.

To summarize; with every step of the agent, our model updates (1) the world model, (2) its statistics and rewards, and (3) the values. A maximum of k nodes is updated at every step. Every node encodes the current action-state, an action-state history equal to the most recent action-state history, a reward, a value and a usage statistic.

4.2 Internal Simulation of Behavior: a Temporary Bias to Predicted Action-State Values

We now explain how internal simulation of action-state pairs (a.k.a. interactions/situations) temporarily biases the predicted value of next actions, and thereby influences action selection. Instead of action selection, the following steps are involved:

1.) S*imulation selection*: at time t select a subset of to-be-simulated interactions (action-state pairs) from the set of interactions predicted by all k active interactrons.

2.) *Simulate*: use a selected interaction from that subset as if it was a real interaction. The agent's memory advances to time t+1. As this is a simulation step, we lack the reinforcement signal r_t that accompanies real interactions. Instead, r_t is simulated using the value, v, of the simulated interaction. We simulate a predicted interaction and its associated value as if they were both real.

3.) *Reset state*: to be able to select an appropriate action in step 4, reset the memory's state (the active nodes) to the previous timestep, i.e., time t. The net effect of step 2 and 3 is that, due to the value propagation mechanism, a temporary bias—based on future predictions at t+1—is introduced to the value of predicted next interactions. Step 2 and 3 are repeated for every to-be-simulated interaction. These biased values are reset in step 5 (after action selection in step 4). If we would keep this bias

after action selection, it would break our model (in RL the reward r must be used to make the value v converge, using v_{t+1} instead introduces a problem of cumulative prediction errors).

4.) *Action selection*: select the next action using the mechanism explained in Section 4.4. Thus, the propagated values of the simulated predicted interactions directly bias action selection. Our anticipation mechanism is best understood as *state anticipation* (Butz, Sigaud & Gerard, 2003).

5.) *Reset values*: reset the reinforcement related variables v, r and v_{next} of the interactions that were changed at step 2 (simulation) to the values of v, r and v_{next} of these interactions before step 2.

In the studies reported in this article, simulation is bounded to a depth of 1, i.e., anticipation is just one step ahead. However, our simulation mechanism can easily support the simulation of multiple time steps ahead by processing step 1 to 3 backwards from $t+d$ to $t+1$ in all possible branches of potential next interactions, with d the simulation depth. Now, action selection at time t is biased by accumulated simulated values of interactions up to d steps ahead. A potential problem is the build-up of small prediction errors. This invalidates the values of next actions, severely compromising action selection. To enable multi-step simulation, accumulation of prediction errors during multi-step simulation should be investigated (e.g., Hoffmann & Möller, 2004).

Step 1 is the s*imulation-selection* mechanism and selects predicted interactions to be simulated. This is a critical component in our simulation mechanisms as it defines the amount of internally simulated information per time step. In our experiments we use four static simulation-selection mechanisms and several dynamic ones (also referred to as *simulation strategies*):

1.) Static simulation selection: sort anticipated interactions according to their predicted value. Select a number of the best anticipated states for simulation. The selected interactions are sent to the model for simulation (step 2).

2.) Dynamic simulation selection: again, anticipated interactions are sorted according to their predicted value. In contrast to static selection, here affect is used to control the amount of predicted interactions that are selected from the sorted list. We explain this in Section 4.3.

In essence, simulation selection is controlled by a selection threshold, t_s, of a t_s-Winner-Take-All (WTA) simulation selection. This threshold, t_s, is used by the simulation-selection mechanism to filter the set of predicted interactions that are simulated, i.e., to select potential next behaviors for

processing in working memory. Our simulation-selection mechanism uses t_s in the following way: t_s defines the percentage of winning *predicted best next interactions* that should be internally simulated (so in a sense it is an inverse threshold). If $t_s < 0$ no simulation is done: the threshold is overly selective, i.e., in WTA terms there is too much "inhibition" to have any winners at all. If $t_s \approx 0$ only the interaction with the highest predicted value is simulated: the threshold is very selective, i.e., in WTA terms there is a lot of "inhibition" and therefore only one winner. If $t_s \approx 1.0$ all interactions are simulated: the threshold is non-selective, i.e., in WTA terms there is no "inhibition" and therefore all predicted next interactions are winners and can be used for internal simulation. The final result of simulation can be summarized as follows: anticipatory simulation introduces a bias to the values of the set of predicted next possible action-state pairs, thereby influencing the result of action selection. In the next section we explain how artificial affect is used to dynamically set the threshold t_s, instead of statically (Broekens, 2005).

4.3 Affective Modulation of WM Content: Affect Controls the Amount of Internal Simulation

Here we introduce our measure for artificial affect, and show how this measure for artificial affect can be used to control the amount of internal simulation of behavior.

4.3.1 Artificial Affect: How Well am I Doing, Compared to What I am Used to?

To model the influence of affect on learning, we first need to model affect in a psychologically plausible way. Our agent learns based on Reinforcement Learning, so at every step it receives some reinforcement r_t. Here we explain how our agent's artificial affect is linked to this reinforcement signal r_t.

Two issues regarding affect induction are particularly important. First, in studies that measure the influence of affect on cognition, affect relates more to long-term mood than to short-term emotion. Affect is usually induced before or during the experiment aiming at a continued, moderate effect instead of short-lived intense emotion-like effect (Dreisbach & Goschke, 2004; Forgas, 2000; Rose *et al.*, 1998). Second, the method of affect induction (explained earlier) is compatible with the method

used for the administration of reward in reinforcement learning. Affect is usually induced by giving subjects small *unanticipated* rewards (Ashby *et al*., 1999; Custers & Aarts, 2005). The fact that these rewards are unanticipated is important, as the reinforcement signal in RL only exists if there is a difference between predicted and received reward. Predicted rewards thus have the same effect as no reward. It seems that learning and affect follow the same rule: *if it's predicted it isn't important*.

We compute artificial affect by:

$$e_p = (r_{star} - (r_{ltar} - f\sigma_{ltar}))/2f\sigma_{ltar} \qquad (4)$$

Here, e_p is the measure for affect. If e_p=0, we assume this means negative affect, if e_p=1 we assume this means positive affect. The short-term running average reinforcement signal, r_{star}, with *star* defining the window size in steps, is the quicker changing average based on the agent's reward, *r*, as unit of measurement at every step. The long-term average reinforcement signal, r_{ltar}, with *ltar* again defining the window size in steps, is the slower changing average taking r_{star} as unit of measurement every step. The standard deviation of r_{star} over that same long-term period *ltar* is denoted by σ_{ltar}, and *f* is a multiplication factor defining the sensibility of the measure.

Obviously, artificial affect behaves differently for different values of *f*, *ltar* and *star*. In general, for r_{ltar} to be a good estimate of what the agent is "used to", *ltar* must be considerably larger than *star*. In the studies presented here we have varied *ltar*, *star* and *f* across a wide range of values.

Our measure for artificial affect reflects the two issues mentioned above. First, r_{star} uses reinforcement signal averages, reflecting the continued effect of affect induction related to mood not emotion. Second, our measure compares the first average r_{star} with the second longer-term average r_{ltar}. As the first, short-term average, reacts quicker to changes in the reward signal than the second, long-term average, a comparison between the two yields a measure for how well the agent is doing compared to what it is used to (cf. Schweighofer & Doya, 2003). If the environment and the agent's behavior in that environment do not change, e_p converges to a neutral value of 0.5. This reflects the fact that anticipated rewards do not influence affect.

4.3.2 Affective Control over the Amount of Internal Simulation: Three Hypotheses

It has now become straightforward to model affective control of the amount of internal simulation. Control can be modeled in several ways. By equating the simulation-selection threshold, t_s, to $1- e_p$, it varies between 0 and 1 depending on affect being positive or negative respectively. Following Rose *et al.* (1999), this reflects the hypothesis that positive affect decreases the amount of internal simulation favoring narrow, exploitative thoughts (i.e., only action-states with a high value are internally simulated), while negative affect increases the amount of simulation favoring broad thoughts, including explorative ones (i.e., action-states with low values are also simulated). In our model this means that content agents (i.e., performing better than expected) simulate positive thoughts, while a discontent agent simulates many thoughts including negative ones. So:

$$t_s = 1 - e_p \qquad (5)$$

Second, we hypothesize the inverse relation, that is, negative affect decreases the amount of simulation while positive affect increases the amount of action-state pairs that can enter working memory for simulation:

$$t_s = e_p \qquad (6)$$

Now, positive affect *increases* the thought-action repertoire (Ashby *et al.*, 1999). This relates to results found by Goschke and Dreisbach (2004).

A third hypothesis is that the intensity of affect controls the amount of simulation, instead of the positiveness and negativeness of affect. Here, intense is either negative affect (e_p=0) or positive affect (e_p=1) while not intense is neutral (e_p=0.5). If affect is intense, simulate a lot (reflecting the fact that significant changes occurred that might need extra processing (Scherer, 2001)). If affect is not intense, do not simulate a lot. Note that intensely positive or negative does not necessarily mean arousing, arousal is considered out of scope for this article. The simulation threshold is:

$$t_s = 2 \times abs(0.5 - e_p) \qquad (7)$$

And, as a control condition, the inverse relation is:

$$t_s = 1 - 2 \times abs(0.5 - e_p) \qquad (8)$$

Systematic studies on the influence of affect-modulated internal simulation on the adaptiveness of artificial agents are presented in Section 6.

4.4 Integrating Everything: Greedy Action Selection over Biased Value Predictions

In our approach, action selection integrates over the action-state values as predicted by all *k* active nodes, each node representing a possible "current state". This is an important difference with standard model-based RL as such models typically use the values for next actions as predicted by one "current state" (e.g., Kaelbling, Littman & Moore, 1996). As a result, our action-selection mechanism is slightly different. It is inspired by parallel inhibition and excitation of actions in the agent's set of actions, *A*. The inhibition/excitation originates from the *k* active interactrons and is calculated as follows:

$$l(a)^t = \sum_{i=1}^{k} \sum_{j=1}^{|X_{y_i}|} v(x^i{}_j \mid y_i)^t \times p(x^i{}_j \mid y_i)^t \quad , \qquad (9)$$

where $l(a)^t$ is defined as the level of activation of an action $a \in A$ at time *t*, and y_i an active interactron at time *t*. Further, x^i_j must predict action *a*. Therefore, $x^i_j = \langle h, a, s \rangle$ with $h = h(y_i) s_{y_i}$ and $\langle h(y_i), s_{y_i} \rangle = y_i$ and $s_{y_i} = \langle a^t, s^t \rangle$ (note that we still use s_i (indexed) to refer to <*a*, *s*> tuples, and *s* to refer to observed states). This clause enforces that any of the action-state pairs that are predicted by any of the *k* active interactrons should inhibit (negative value) or excite (positive value) the corresponding action, *but not other actions*. Finally the action *a* to be executed is such that:

$$l(a)^t = max\,(l(a_1)^t, \ldots, l(a_{|A|})^t) \qquad (10)$$

If there are only bad actions (i.e., $l(a)^t < 0$) a weighted stochastic selection based on $l(a_1)^t, \ldots, l(a_{|A|})^t$ is made instead; the action with the highest activation has proportionally the highest chance of being chosen resulting in a probabilistic WTA action selection. As such, action selection uses a super-threshold greedy selection with sub-threshold linear weighted stochastic selection.

Depending on when the action-selection mechanism is invoked it either uses unbiased (before simulation) values to select the next action, or biased (after simulation) values to select actions. This allows us to address the main question of our study: what happens if action-selection bias is induced

by an amount of simulated anticipatory behavior, and if this amount is dynamically controlled by artificial affect?

To summarize, the number of thoughts that occupy working memory is often interpreted as an indicator of the intensity of information processing. As (1) a thought equals an internally simulated behavior in our model, and (2) the number of thoughts that occupy working memory equals the amount of internally simulated behavior, it is now clear that we indeed study affective control over information processing.

5 Method

To investigate the influence of affect-controlled anticipatory simulation of future action-state pairs, we have set up a gridworld environment consisting of walls, roadblocks, cues, food and empty spaces. We use two nondeterministic, partially observable gridworlds. Common to our two gridworlds is that the agent *can* walk on walls, but is discouraged to do so, which is why we call our "wall" "lava" (reinforcement r=−1.0). The agent moves around by selecting an action a from the set of possible actions A={*up*, *down*, *left*, *right*}, and observing its immediate surroundings (not its position) using a four-neighbor-plus-center metric just after executing the action. This is an <a, s> tuple as used in the model (Section 4).

The first gridworld is taken from (Broekens & Verbeek, 2005), and aims to test how well agents using different simulation strategies can cope with a sudden change in both reward and world structure (Figure 3). In this world, the agent (black square) learns to cope with two alternating goal and start locations (f=food, reinforcement r=1.0). Alternation is random and after every trial. A trial ends when the agent has found the goal: the agent is put back at a randomly chosen start location after having reached the randomly chosen goal location. The total number of trials to learn a task is 500. We define such sequence of 500 trials as a *run*. Additionally, at trial 250, the world is changed: two negatively reinforced roadblocks (b=block, r=−0.5) are placed in front of the goal locations, and the food reward is increased to 1.75 to compensate for the roadblocks. Consequently, both the world and the reward structure of that world change. The agent is unaware of this change, and, as our model

learns lazily, no value updates or world-model changes are made. The agent has to learn these new characteristics of the world. We call this gridworld the *switch-to-invest* gridworld, as it is constructed to measure how an agent copes with a change in the environment that introduces an investment to be made before an otherwise easily obtainable goal.

(Figure 3 and 4 about here)

The second world is based on a typical psychological method in which subjects have to learn to cope with a cue-meaning inversion (see, e.g., Goschke & Dreisbach, 2004). This type of method is used to investigate the effect of an experimental variable, e.g., affect (Goschke & Dreisbach, 2004) on working memory flexibility by measuring reaction time just after the cue-meaning inversion. It is also used to measure adaptation speed to the new cue-meaning relation after having learned the old relation. In the case of our simulated gridworld, a cue is coupled to a specific food location, while the absence of that cue is coupled to a different food location. At trial 250 the locations are inversed. This means that whereas before trial 250 the cue indicated to the agent that food is at location 1, after trial 250 the cue ('c' in Figure 4) indicates that food is at location 2. We call this world the *cue-inversion* world. In contrast to the switch-invest task, the agent is also reset to its (fixed) starting position when it arrives at the non-goal location. The non-goal location has a negative reinforcement of r=–0.5.

To test our three hypotheses, we vary the simulation-selection mechanism of our agents. In total, we define four static simulation-selection mechanisms:

1.) No simulation; simulation is off (called *nosim* in the experiments).

2.) Simulation of the best predicted action-state pair; t_s=0, (*simbest*).

3.) Simulation of the best half of predicted action-state pairs, i.e., t_s=0.5, (*simbest50*).

4.) Simulation of all predicted action-state pairs, i.e., t_s=1, (*simall*).

We also define four dynamic simulation mechanisms, introduced in Section 4.3.2. These are:

1.) Positive affect = little simulation (select best predicted action-state pairs), and vice versa, (*dyn*).

2.) Negative affect = little simulation, and vice versa, (*dyn inv*).

3.) High intensity of affect = little simulation, and vice versa, (*dyn intensity*).

4.) Low intensity of affect = little simulation, and vice versa, (*dyn intensity inv*).

In the switch-to-invest experiments we have used all four static simulation strategies and only the first two dynamic ones. In the cue-inversion experiments we have used all eight simulation strategies. We varied the three parameters that define the behavior of our measure of affect, i.e., we varied *f* (sensitivity of affect), *ltar* (the window size of the long-term averaged reward that defines "how well is usual"), and *star* (the window size of the short-term average reward that defines "how am I doing").

In our switch-to-invest gridworld experiments we varied these according to Table 1, resulting in 30 different affect-parameter settings. In our cue-inversion gridworld experiments we varied these only according to the *f*=1 column in Table 1, resulting in 10 different affect-parameter settings.

In our switch-to-invest experiments we varied the learning rate, α = [0.8, 0.9, 1.0], and the rate at which the model forgets information about the world as defined by the "survival potential threshold" of nodes, θ = [0, 0.01, 0.02, 0.03]. In the cue-inversion experiments α and θ are not varied but fixed at 1 and 0 respectively.

(Table 1 about here)

6 Experimental Results

We first describe the results obtained with the switch-to-invest gridworld, after which we describe the results obtained with the cue-inversion gridworld. Data was analyzed as follows. To investigate the effect of learning rate, α, rate of forgetting, θ, and simulation strategy we compare between results of different <α , θ, *simulation strategy*> configurations. Static simulation strategies have been executed 200 times per <α , θ, *simulation strategy* > configuration, e.g., the simulate-best strategy has been executed 200 times for every <α , θ> combination. These 200 runs are the basis for further analysis. Dynamic simulation strategies have been executed 15 times per <α , θ, *f, ltar, star, simulation strategy*> configuration. For every <α , θ, *simulation strategy*> configuration, the resulting runs for all of its <*f, ltar, star*> settings is aggregated. For example in the switch-to-invest experiments, for α = 1, θ = 0, and strategy=*dyn* we aggregated all 15 x 30 (*nr of runs* times *nr of affect-parameter settings*, respectively) runs into 450 runs. These runs are the basis for further analysis. In the cue-inversion experiments the same aggregation protocol was used, but, as mentioned above, here we use only one

<α , θ> configuration and we vary only *star* and *ltar* (not *f*). Further, we used 50 runs per <α , θ> configuration resulting in 50 x 10 runs =500 runs being aggregated for only one setting (α = 1 and θ = 0). We aggregated the data as our goal is to investigate the effect of affective control of simulation selection *in general*, not to find specific values that "work" for the agent. We did not seek to optimize any parameter but to investigate different relations between affect and simulation selection. Between simulation strategies we compare the following:

(1) A measure for the behavioral *effort* involved in completing a run (i.e., learning the complete task) for each specific simulation strategy. Effort is calculated by *first averaging trial length in steps over all trials for each run, resulting in an effort for that run.* This is our unit of measurement for statistical analysis (e.g., if there are 450 runs for one strategy, we have 450 measures of effort to use in our statistical analysis for that strategy). To *display* the average effort for a certain simulation strategy, we *average over the measure of effort for all runs for that strategy*. For example in a static selection mechanism (α = 1 and θ = 0), the displayed effort equals the mean number of steps needed for one trial over all 500 trials in all 200 runs resulting in, e.g., the number 20. For a dynamic simulation mechanism the average is constructed in the same way using aggregated runs for every <α , θ> configuration instead. The Wilcoxon ranked-sum test (non-parametric, we cannot assume normality) is used to compare effort between simulation strategies. Comparison is based on sets of effort measures (switch-to-invest: n=450; cue-inversion: n=500). For static strategies 450 samples (switch-to-invest) or 500 samples (cue-inversion) are pooled from the 200 runs that are available.

(2) A measure for the total *simulation effort* involved in completing a run, i.e., the same as above but using a trial length counted in terms of internally simulated action-state pairs. This represents "mental effort" during a task, and as such is linked to energy consumption used to maintain and focus on information in working memory. Again, the Wilcoxon test is used to compare between simulation strategies.

To give an informal idea of the learning behavior of the agent, several learning curves of agents are plotted. Learning curves are plots of the *average number of steps taken per trial* and smoothed using a sliding mean (window size=10) to improve readability.

6.1 Results of Experiment 1: Switch-to-invest

Results in this specific gridworld show that simulation in general has a stable positive effect on learning. This trend is shown by the learning curves in Figure 5, and more formally in Figure 6 showing that *nosim* uses more effort to complete a run than any other simulation strategy ($p<0.001$). The larger the amount of internally simulated anticipations, the better the learning result (*simall* costs less effort than *simbest*, $p<0.05$ for all settings except $\alpha=1$ & $\theta=\{0, 0.01\}$, Figure 6). When affect is used to control this amount, performance is better than the static simulation mechanism that simulates the best strategy (a significant difference between *dynsim* and *simbest*, $p<0.05$ for all settings except $\alpha=1$ & $\theta=\{0, 0.01\}$, Figure 6). Interestingly, the size of the effect interacts with the learning rate and forgetting rate. As θ increases, the benefit of simulation also increases, and as α decreases the benefit of simulation increases (Figure 6). In terms of size, we did not find important differences between (1) the dynamic strategy that relates negative affect to more simulation and (2) the dynamic strategy that relates positive affect to more simulation. Even though the strategies are each other's inverse, the difference in effort was at most about 5% (Figure 7, left). However, for all $<\alpha, \theta>$ settings, the average amount of simulation effort was considerably less for *dyn* than for *dyn inv* ($p<0.001$). Further, both strategies simulated considerably less than *simall* ($p<0.001$), while *dyn* used less simulation effort than *simbest50* ($p<0.001$) (Figure 7, right, shown only for $\alpha=0.8$). Finally, results for $\alpha=0.9$ are not shown, as these appeared to be an interpolation between the results for $\alpha=0.8$ and $\alpha=1.0$.

(Figure 5, 6 and 7 about here)

6.2 Discussion of the Switch-invest Task Results

The fact that more simulation results in better performance is not surprising. Internal simulation as an anticipatory heuristic can use more knowledge if it selects more potential next interactions. Thereby, it influences final action selection in a more balanced way. Interestingly, there is an interaction effect produced by learning rate, rate of forgetting and simulation. Regarding the learning rate this effect is easily explained. As internal simulation enables the agent to "look ahead" one step, predicted values can be temporarily propagated back. Even though the model does not learn based on simulation (i.e., nodes, their value, reward and statistic are not permanently updated due to simulation), simulation has

an immediate benefit for action selection, as more information is temporarily available. If the learning rate is high (α=1.0), this effect is minimized: at every step the agent takes, the lazy update rule propagates future values back in full, so simulation cannot add a lot of future value information. However, if the learning rate is small(er) (e.g., α=0.8), the future value is not propagated in full. Now, internal simulation can temporarily propagate values that were not yet propagated in full, and the action-selection mechanism can benefit from the extra information provided by simulation. This phenomenon causes a performance increase due to simulation in lower learning rate settings.

It is not yet clear from our experiments what causes the interaction between rate of forgetting and simulation, although it is clear that it can not be simulation per se, as simulation does not change the model's statistics. A possible explanation is that simulation in general forces the agent to use known interaction patterns more often than new or less-tried patterns. As such, simulation actually reduces the probability of forgetting useful interactions. This could help solving the maze with a forgetful long-term memory. This requires further investigation in future research.

The fact that the two dynamic simulation strategies tested (a) do not differ in terms of learning performance, (b) perform at about the same level as the static simulation strategy that simulates all potential next interactions, and (c) use a considerably reduced amount of simulation compared to this static *simall* strategy, indicates two things: (1) dynamic adaptation is beneficial as it reduces simulation needs (an interesting result), and (2), it does *not* matter if positive affect implies more simulation or less, as the two dynamic simulation strategies result in less simulation *and* better learning performance. If the latter is indeed the case, this implies one of the two following possibilities: (I) affect has nothing to with the result. Instead, the average amount of simulation is responsible for the increase in learning performance. This possibility is supported by our results, as the *dyn inverse* strategy uses more simulation than *dyn* (Figure 7, right) and seems to perform slightly better than the latter (Figure 7, left). On the other hand, it could also imply that (II), affect *does* have to do with the result, but both relations—i.e., positive affect = more simulation, and positive affect = less simulation—are wrong. This is possible if the relation instead is: higher intensity affect=more simulation. We study this in the second experiment, and use the intensity-of-affect-based simulation strategies. In this experiment we use the second gridworld, i.e., the cue-inversion world (Section 5).

6.3 Results of Experiment 2: Cue-inversion

Results in this gridworld show the following. The *simbest* static simulation strategy does not have a large positive effect (even though the effect is significant $p<0.01$), contrary to the results in experiment one where the effect was more pronounced. However, *simall*, *simbest50* as well as all dynamic simulation strategies do have an important positive effect ($p<0.001$); effort is reduced with 0.6 to 1 step per trial. Thus, a moderate positive influence of simulation on learning performance exists. Note that the smaller effects of simulation in general, as compared to the previous experiment, are due to the fact that in this experiment $\alpha=1$ and $\theta=0$. As such, smaller effects are expected and confirm our explanation in the discussion of the previous experiment.

Again, dynamic strategies are quite close to the *simall* strategy in terms of their learning performance (Figure 8, left), the only significant difference in effort is between *simall* and *dyn intensity* ($p<0.01$). However dynamic strategies use considerably less simulation effort to get to this increased level of performance (Figure 8, right, all strategies use less simulation than *dynall,* $p<0.001$). An important difference in effort exists between the two intensity-based dynamic simulation strategies. The *dyn intensity inverse* strategy (i.e., if affect is neutral, 0.5, simulate a lot, while if affect is extreme, 0 or 1, simulate little) has a better performance than *dyn intensity* ($p<0.001$, Figure 8, left), but also uses a lot more simulation ($p<0.001$).

Last, we plot the average behavior (over 50 runs) of our measure for artificial affect as it is influenced by *ltar* and *star*. A large long-term window to calculate the agent's measure of comparison based on reward (i.e., "what I am used to") results in less noisy affect (Figure 10). A small short-term average (i.e., "how am I doing") results in a faster affective reaction to the cue inversion (inset).

(Figure 8, 9, and 10 about here)

6.3.1 Discussion of the Cue-inversion Results

The fourth dynamic control strategy based on the inverse intensity of affect (*dyn intensity inv*) results in a better performance than the third, intensity based, control strategy. Again, this inversed version (i.e., neutral affect results in a lot of simulation and extreme affect in a little) uses more simulation on average. Thus, this result does not rule out the possibility that the average amount of simulation is

responsible for the learning performance increase as opposed to affective control. We need to correct for the average amount of simulation. To do so, we defined the *gain* ratio, a measure that calculates how much effort reduction a strategy gives relative to no simulation, weighted by the amount of simulation effort. As such,

$$gain_i = (effort_{non} - effort_i) / (sim\_effort_i / effort_i), \qquad (11)$$

where $effort_i$ equals the effort for a certain simulation strategy i, $effort_{non}$ equals the effort of the *nosim* strategy and sim_effort_i equals the simulation effort for a certain strategy i. Such a gain factor is a plausible measure to evaluate and compare simulation strategies: one is interested in the efficiency of simulation, not just the absolute result. As simulation—i.e., information maintenance in working memory—costs resources, the question is which strategy uses these resources best. When we compared the gains for the different simulation strategies, a different picture emerged (Figure 9). Simulating all is not very efficient compared to dynamic strategies. Interestingly, our original coupling of affect and amount of simulation seems most promising (Broekens and Verbeek, 2005). This is the only strategy of which the gain confidence interval does not overlap with either *simall* or *simbest50*. This means that, although the relation "positive affect equals less simulation and negative affect equals more simulation" is not the best one in terms of effort reduction, it is the optimal one in terms of *relative gain when considering the amount of simulation needed for that effect*.

7 General Discussion

First we ground our approach more firmly, and relate our work to the work of others. Finally we present some directions for future research.

7.1 Model Grounding

Our findings are compatible with psychological findings that show that both positive and negative affect influence learning in a beneficial way (Dreisbach & Goschke, 2004; Rose *et al.*, 1999). We found that learning benefits the most when positive affect relates to less simulation and negative to more simulation. As such, our findings indicate that positive affect is associated with less diverse thoughts when a task has successfully been learned, while negative affect is associated with diverse

thoughts when a task is confusing or changing. Our findings support the studies by Rose *et al.* (1999) who find that broad attention is associated with faster learning and negative affect, when a new task has to be learned. Our findings are also consistent with the relation that has been found between subclinical depression and defocused attention (von Hecker & Meiser, 2005). In agreement with these authors, we would like to stress that our results do not necessarily argue for a "positive affect equals reduction of capacity" view. More selective maintenance of information is not the same as a reduction of capacity. Selectivity of maintenance in WM that depends on affect can be an adaptive strategy to cope with the changing world around us, without enforcing any capacity constraints.

By defining artificial affect purely in terms of reward one could argue that we interpret affect in a too narrow sense. We do not agree. Our meaning of artificial affect is still the same as the meaning of affect: it defines the goodness/badness of a situation for the agent. Further, it is quite compatible with certain theories of emotion (e.g., Rolls, 2000) that emphasize that emotion is fundamentally grounded in (the deprivation/expectancy of) reward. Finally, as rewards define what behavior an artificial RL agent should pursue and avoid, reinforcement *is* the definition of good and bad for such agents. We believe our measure for artificial affect as well as how we use it are firmly grounded as we have: (1) linked the time scale and the elicitation of artificial affect to the time scale and elicitation of natural affect, (2) tested three psychologically plausible hypotheses of affective control over internal simulation, and (3) based our cue-inversion task on psychological measurement methods that measure the influence of affect on cognition.

In our approach, internal simulation influences action selection in a way that is compatible with the somatic marker hypothesis (SMH) (Damasio, 1994). In short, the SMH states that somatic (i.e., of the body) signals are coupled with representations of situations and thereby function as a value signal that enables the organism to filter potential behaviors. As a result, some of these potential behaviors are selected for conscious contemplation in working memory while others are not. Our threshold determines how discriminating our simulation-selection mechanism is, thereby selectively allowing some anticipated behaviors to enter working memory and influence future behavior. Of course we do not argue that we have an embodied approach; our agent is quite disembodied. However, our action-state value v can be interpreted as a simulated marker, as it accumulates future

values of potential situations. As such, it is an abstraction of the somatic signal that, in an embodied modeling approach and in nature, is grounded in the body. We argue that our mechanism of simulated interaction selection, and thus selection of WM content, is compatible with the mechanism by which somatic markers are used to prune large amounts of thoughts. Both mechanisms prioritize different anticipated behaviors based on a comparison of their markers. Only potential behaviors (thoughts) that have highly positive markers—or *strong* markers, if the *intensity* of artificial affect is used as selection threshold (cf. Section 4.3.2)—are able to influence future behavior by temporarily transferring a portion of their own marker value to the marker value of considered actions (see also Damasio, 1994). In our model, transfer of marker values is a natural consequence of simulating a particular future interaction (Section 4.2).

Concerning the relation between our model and the simulation hypothesis, several similarities are particularly important. Hesslow (2002) states that fundamentally new mechanisms should not be needed for internal simulation of behavior. The only mechanism we introduce is an interaction feedback loop to the RL model. We do not introduce a conscious reasoning process or a central intelligence that enables planning. Compared to such measures, our addition is just a minor change to the overall agent architecture, and comparable with the addition of a feedback connection in neural network models that investigate internal simulation (van Dartel & Postma, 2004; van Dartel, Postma & van den Herik, 2005). Further, our mechanism for simulation selection is very similar to that of action selection: the RL model is used in the same way in both the simulation (cognitive) and non-simulating (reactive) setting; simulation selection uses the action-selection component; and the representations used for simulation are the same as those used for action.

Hesslow (2002) also states that internal simulation of behavior uses the same sensory-motor mechanisms as actual behavior, and as such uses similar sensory-motor encoding. Our interactions encode features of the world coupled with actions, and our model uses these same interactions for simulation. More importantly, in our model, simulation influences action indirectly: an influence that is caused *only* by making use of the same mechanisms needed for action. This is very compatible with the simulation hypothesis stating that simulation and action are tightly coupled. Our mechanism for influencing action selection is therefore a useful addition to the simulation hypothesis by postulating a

potential mechanism by which internal simulation could influence action: i.e., simulation temporarily biases next actions *because* the simulation mechanism and action mechanism overlap and therefore simulation activates potential next actions to some extent, resulting in the "markers" of the simulated consequences to be temporarily attached to these next actions.

7.2 Related work

To show that simulation in our model can indeed be seen as an instantiation of simulation as meant by the simulation hypothesis we compare it with the models by van Dartel & Postma (2005), van Dartel *et al*. (2005) and Ziemke, Jirenhed and Hesslow (2005). These models use a genetic algorithm to train a neural network to produce predictions of future states one time step ahead. These predictions are used to bias perception of the current state (van Dartel and coworkers), or explicitly used as input to the neural network controller to enable "'blindfolded' corridor following behavior" based on these simulated next states (Ziemke and coworkers). Although our action-state encoding and learning mechanism are different, our overall architectural approach is similar, especially to the work of van Dartel and coworkers. Simulation in the latter work is modeled as follows. A copy of the output layer (encoding actions) of the neural network is projected to the input layer. This output copy consequently influences perception, and as such influences action selection. The feedback from this copy to the input represents a simulated next state as predicted by the model (Dartel & Postma, 2005). These authors explicitly suggest that in their model internal simulation "serves the function of building up sufficient activation in the neurocontroller to produce a certain move". This is equivalent to what happens when in our model future interactions are simulated, as these simulated interactions bias the "markers" of current potential actions and as such can help certain actions to be executed. The work of Ziemke *et al.* (2005) is a bit different. They train an "input prediction layer" to predict the next observed state based on the current one. This prediction is used as input to an already trained sensory-motor network responsible for collision-free corridor-following behavior. The predicted state is used as real input to the sensory-motor network such that the agent as a whole walks through the corridor based on mental simulations of interaction with the corridor, i.e., it is walking "blindfolded". The characteristic difference between this model and our model is that Ziemke *et al*. use the predicted next

state as input for action selection, while in our model the simulated input is used as a bias, as in the model by van Dartel. However, from an architectural point of view, the three models are all instantiations of the simulation hypothesis: the models internally simulate predicted interaction with the environment in order to influence actual interaction, while using the same encoding and the same mechanisms for both real and simulated interaction.

Simulation in our approach is to some extent similar to planning in *Dyna* (Sutton, 1990). However, several important differences exist. First, our model learns multiple MDPs in parallel and uses all of these MDPs in action selection. Second, anticipatory simulation in our model (cf. planning in *Dyna*) is always a one-step forward simulation from the current state, not a simulation of a random state. This reflects our choice of basing the model on anticipatory simulation of behavior, and not on planning or dynamic programming in general. As such, the potential of simulation in our model is more limited. Third, our model can only simulate actions it has tried already, effectively restricting the exploration potential of simulation: our agent cannot really *explore* mentally, it can only consider the many known future options, in contrast to *Dyna* in which untried actions can be simulated. However, in order to do so, *Dyna* requires a non-empty world to start learning (Sutton, 1990). We have chosen to start learning with a completely empty model. Therefore we could not simulate untried actions, at least not without making major changes to the representations of action-state pairs and transitions between them. Finally, simulation in our model has a temporary effect on values of next states, while in *Dyna*, planning can change these values. Notwithstanding these differences, our method of internal anticipatory simulation of states replicates some of the results obtained with *Dyna* (Sutton, 1990), of which the most relevant in the context of the presented results is that simulation (and more simulation rather than less) has a positive effect on learning speed.

Our results show that internal anticipatory simulation is beneficial to artificial adaptive agents. Simulation introduces a temporary bias to the values used in action selection. This approach is similar to the one proposed by Gandanho (2003). In their RL based adaptive system, however, stochastic action selection is biased by a fixed value produced by a rule-based cognitive system. In contrast, in our system this value is dependent on the predicted states and the cognitive process is not separated from the adaptive system. We did not separate these systems as the simulation hypothesis is

underlying our approach. As internal simulation of behavior is based on existing sensory-motor mechanisms, it made sense to investigate the benefit of anticipatory simulation using as many functions as possible already provided by our RL model.

Our work relates to emotion-, and motivation-based agent control. It explicitly defines a role for emotion in biasing behavior selection (Avila-Garcia & Cañamero, 2004; Cos-Aguilera *et al*., 2005; Velasquez, 1998). The main difference is that in these studies emotion directly influences action selection (or motivation(al states)), while we have studied the indirect effect of emotion-controlled information processing influencing action selection.

In a recent variation of this type of research (Blanchard and Cañamero, 2006) artificial novelty and affect are coupled to exploration behavior of a robot that has to autonomously explore different possible distances to a box. Familiarity (inverse novelty) modulated by positive affect is coupled to exploration. However, their concept of exploration (in contrast to ours) is limited to the single behavioral choice of whether or not the robot should approach the box. This strongly narrows down the meaning of exploration, as acknowledged by the authors. Our approach thus contributes to this research by systematically investigating how affect can be used to modulate (mental) exploration in a broader sense.

Strongly related to our approach to affect-modulated exploration is the research by McMahon *et al.* (2006). The authors show how the discrete choice between exploration and exploitation trials can be controlled by a probability value that is derived from measures inspired by affect. Several interesting differences between their approach and ours should be noted. First, our artificial affect dynamically modulates the amount of mental exploration that influences action selection, while their probability is used for a discrete choice between whether a trial is an exploration or an exploitation trial. Second, their reward-related measure of affect is based on a scaled value for the current reward, where scaling is based on the min and max rewards obtained in the environment. This means that this measure is unable to model "boredom" (McMahon *et al.*, 2006). Our measure of affect—also related to (the history of) rewards—addresses this issue and, as such, is a useful extension to the work of McMahon and colleagues. When our agent has acted in the same environment for a long time, the long and short-term averages will converge to the same value and as such artificial affect will be

lower, even though the agent might receive huge rewards. In our first hypothesis, low artificial affect results in higher (mental) exploration. This is "boredom" in exactly the same nature as proposed in (McMahon *et al.*, 2006). Third, we have extended the analysis of the psychological plausibility of reward-related measures for artificial affect, which is an issue of future work in (McMahon *et al.*, 2006).

Two fairly different approaches towards studying the relation between affect and adaptive behavior are the work by Lahnstein (2005) and the work by Salichs and Malfaz (2006). Lahnstein shows how the short-term emotive episode can result from anticipation of reward in the first phase of approaching a reinforced object, while in the second phase the emotive episode is taken over by an evaluation of the actual reward received from that object. This research is important to understand the process of emotion elicitation in adaptive agents in the spirit of, e.g., Rolls (2000). The main difference between Lahnstein's approach and ours is that we use affect in the "mood" (long term) sense as influence on the broadness of mental exploration, while Lahnstein focuses on the process of elicitation of the short-term emotive episode produced by mental anticipation and reward evaluation. It would be interesting to integrate Lahnstein's result with ours, such that our measure of long-term affect is based upon averages over the positive/negative aspect of Lahntein's short-term emotion.

Salichs and Malfaz (2006) show how affect can be embedded into the value function Q of a standard reinforcement learning method. They enhance Q-learning such that the reward is based on the happiness/sadness of the agent, where happiness and sadness are derived from the agent's well-being. Well-being is a function over the extent to which the agent's drives are met. This means that their agent is intrinsically motivated by affect, and strives to "maximize happiness". Further, their agents use fear to dynamically modulate the amount of risk taking. Their approach differs from ours, but both approaches could be integrated such that well-being based on drives provides the reward signal and thus our measure for artificial affect is based upon well-being averages.

7.3 Future work

It is worth investigating how other simulation-selection and action-selection mechanisms (e.g., non-greedy) behave in relation to affective control. In recent experiments we have investigated

how a Boltzmann-based action-selection mechanism (Broekens, Kosters & Verbeek, 2007) can be controlled by affect. These studies indicate that affect can successfully be used to control the exploration rate (the Boltzmann *Beta* in the action-selection process) of an adaptive agent during the learning process.

The maximum total amount of simulation could be fixed, while affect controls *when* to simulate. Now, experiments can be conducted to completely control for the generic effect of the positive influence of more simulation on learning. Arousal could control simulation by, e.g., controlling the depth of anticipation (or the forgetting rate of the memory so that arousal influences the adaptation speed of the memory).

Even though affective control of internal simulation of behavior seems promising for adaptive behavior and is compatible with psychological findings, our learning model is specific. This means that our claims are hard to generalize. A good way to further investigate the mechanisms of affective control introduced in this paper is to use different learning architectures, such as *Soar*, or ACT-R. Using the ACT-R architecture, Belavkin (2004) shows that affect and arousal can be used to control the search through the solution space, which resulted in better performance. The “Salt” model by Botelho and Coelho (1998) relates to this approach in the sense that the agent's effort to search for a solution in its memory depends also on the agent's mood valence.

As *Soar* has recently been extended with RL mechanisms, called *Soar*-RL (Nason & Laird, 2004), it is becoming a good candidate for adaptive behavior research. First, *Soar* is a well understood architecture. Second, *Soar* allows many forms of planning, enabling a better comparison between affective control of planning versus forward internal simulation. We are currently investigating affect-based control techniques in *Soar*-RL (Hogewoning, Broekens, Eggermont & Bovenkamp, 2007).

Affective control should be investigated in other (more realistic, more complex, larger) types of tasks and learning environments, as different environments have their own set of difficulties and particularities for action selection and learning, and imply different functions and benefits for emotion (Cañamero, 2000).

On the biological level, there is considerable evidence of the link between positive affect, adaptive behavior and dopamine (Ashby *et al.*, 1999), as well as dopamine, RL, and adaptive behavior

(Dayan & Balleine, 2002; Montague, Hyman & Cohen, 2004; Schultz, Dayan & Montague, 1997). Relating our model to this literature is a direction for future work.

8 Conclusion

Using a computational model based on reinforcement learning, we have investigated affective control of anticipatory thoughts, where thoughts are defined as internal simulation of potential next behavior (Cotterill, 2001; Hesslow, 2002). We have introduced a simulation-selection mechanism that is controlled by affect and selects anticipatory behaviors for simulation from the predictions of the RL model used by the agent. The selected anticipatory behaviors are used to bias the predicted values of next action-state pairs. Action selection is over these biased pairs, thereby influenced by the simulated anticipations. Based on experiments with adaptive agents that learn two nondeterministic partially observable gridworlds we conclude that (1) internal simulation has an adaptive benefit and (2) affect can be used to control the amount of simulation. The results show that affective control reduces the amount of simulation needed to get a performance increase due to simulation.

The positive effect of internal simulation has been shown to exist for two nondeterministic partially-observable worlds, and already has been shown to exist in other worlds (Broekens, 2005). However, selecting all possible next action-state pairs for simulation provides quite some computational overhead, or, in more biological terms, consumes a considerable amount of energy to maintain stable representations in working memory (WM) that can be used to construct anticipatory associations. In this study we have shown that affect can regulate the amount of anticipatory simulation in such a way that learning is still improved considerably. Although it is difficult to generalize from computational experiments that contain many variables, in terms of WM-affect relation our results indicate that affective control of the amount of anticipatory thoughts in WM enables an adaptive agent to make more efficient use of WM.

The most beneficial relation between affect and internal simulation is observed when positive affect decreases the amount of simulation towards simulating the best potential next action, while negative affect increases the amount of simulation towards simulating all potential next actions. Ergo, agents “feeling positive” can think ahead in a narrow sense and free up working memory resources,

while agents “feeling negative” must think ahead in a broad sense and maximize usage of working memory. Our results are consistent with several psychological findings on the relation between affect and learning, and contribute to answering the question of *when* positive versus negative affect is useful during adaptation. Furthermore, our results show that simulation selection is a useful extension to action selection, specifically in the context of the *simulation hypothesis* (Hesslow, 2002).

Acknowledgements

We would like to thank Joanna Bryson and all other reviewers for their excellent criticism. Their input has been of considerable benefit to this paper.

References

Ashby, F. G., Isen, A. M., & Turken, U. (1999). A neuropsychological theory of positive affect and its influence on cognition. *Psychological Review, 106*(3), 529-550.

Aylett, R. (2006). Emotion as an integrative process between non-symbolic and symbolic systems in intelligent agents. *Proc. of the AISB'06 Symposium on Architecture of Brain and Mind* (pp. 43-47). AISB Press.

Avila-Garcia, O., & Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. *From Animals to Animats 8: Proc. 8th Intl. Conf. on Simulation of Adaptive Behavior* (pp. 243-252). Cambridge, MA: MIT Press.

Baddeley, A. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences, 4*(11), 417-423.

Belavkin, R. V. (2004). On relation between emotion and entropy. *Proc. of the AISB'04 Symposium on Emotion, Cognition and Affective Computing* (pp. 1-8). AISB Press.

Berridge, K. C. (2003). Pleasures of the brain. *Brain and Cognition, 52*, 106-128.

Bickhard, M. H. (1998). Levels of representationality. *Journal of Experimental and Theoretical Artificial Intelligence, 10,* 179-215.

Blanchard, A. J., & Cañamero, L. (2006). Modulation of exploratory behavior for adaptation to the context. *Proc. of the AISB'06 Symposium on Biologically Inspired Robotics (Biro-net)* (pp 131-137). AISB Press.

Botelho, L. M., & Coelho, H. (1998). Information processing, motivation and decision making. *Proc. 4th International Workshop on Artificial Intelligence in Economics and Management*.

Broekens, J. (2005). Internal simulation of behavior has an adaptive advantage. *Proc. of the CogSci'05 Conference* (pp. 342-347). Mahwah, NJ: Lawrence Erlbaum Associates.

Broekens, J., & DeGroot, D. (2004). Emergent representations and reasoning in adaptive agents. *Proceedings of the Third International Conference on Machine Learning and Applications* (pp. 207-214). IEEE press.

Broekens, J., Kosters, W. A., & Verbeek, F. J. (2007). On affect and self-adaptation: Potential benefits of valence-controlled action-selection. In: J. Mira and J.R. Álvarez (Eds.), *IWINAC 2007, Part I, LNCS 4527* (pp. 357-366). Springer-Verlag.

Broekens, J., & Verbeek, F. J. (2005). Simulation, emotion and information processing: Computational investigations of the regulative role of pleasure in adaptive behavior. *Proc. of the Workshop on Modeling Natural Action Selection* (pp. 166-173). AISB Press.

Butz, M. V., Sigaud, O., & Gerard, P. (2003). Internal models and anticipations in adaptive learning systems. In: *LNAI 2684: Anticipatory Behavior in Adaptive Learning Systems* (pp. 86-109). Springer-Verlag.

Cañamero, D. (2000). Designing emotions for activity selection. *Dept. of Computer Science Technical Report DAIMI PB 545*. University of Aarhus, Denmark.

Clore, G. L. & Gasper, K. (2000). Feeling is believing: Some affective influences on belief. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge Univ. Press, Cambridge, UK.

Cohen J.D., & Blum K. I. (2002) Reward and decision. *Neuron,* 36, 193-198.

Cos-Aguilera, I., Cañamero, L., Hayes, G. M., & Gillies, A. (2005). Ecological integration of affordances and drives for behaviour selection. *Proc. of the Workshop on Modeling Natural Action Selection* (pp. 225-228).

Cotterill, R. M. J. (2001). Cooperation of the basal ganglia, cerebellum, sensory cerebrum and hippocampus: Possible implications for cognition, consciousness, intelligence and creativity. *Progress in Neurobiology, 64,* 1-33.

Craig, S. D., Graesser, A. C., Sullins, J., & Gholson, B. (2004). Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media, 29*(3), 241-250.

Custers, R., & Aarts, H. (2005). Positive affect as implicit motivator: On the nonconscious operation of behavioral goals. *Journal of Personality and Social Psychology, 89*(2), 129-142.

Damasio, A. R. (1994). *Descartes' Error*: *Emotion, Reason, and the Human Brain*. New York, NY: Penguin Putnam.

Davidson, R. J. (2000). Cognitive neuroscience needs affective neuroscience (and vice versa). *Brain and Cognition, 42*, 89-92.

Dayan, P., & Balleine, B. W. (2000) Reward, motivation, and reinforcement learning. *Neuron 36*(2), 285-298.

Deheane, S., Sergent, C., & Changeux, J-P. (2003). A neuronal network model linking subjective reports and objective physiological data during conscious perception. *Proceedings of the Natural Academy of Sciences, 100*(14), 8520-8525.

Demiris, Y., & Johnsons, M. (2003). Distributed, predictive perception of actions: A biologically inspired robotics architecture for imitation and learning. *Connection Science, 15*(4), 231-243.

Doya, K. (2002). Metalearning and neuromodulation. *Neural Networks, 15*(4), 495-506.

Dreisbach, G., & Goschke, K. (2004). How positive affect modulates cognitive control: Reduced perseveration at the cost of increased distractibility. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*(2), 343-353.

Forgas, J. P. (2000). Feeling is believing? The role of processing strategies in mediating affective influences in beliefs. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature, 440*, 680-683.

Frijda, N. H., & Mesquita, B. (2000). Beliefs through Emotions. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

Frijda, N. H., Manstead, A. S. R. & Bem. S. (2000). The influence of emotions on beliefs. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

Gandanho, S. C. (2003). Learning behavior-selection by emotions and cognition in a multi-goal robot task. *Journal of Machine Learning Research, 4*, 385-412.

Griffith, P. E. (1999). Modularity & the psychoevolutionary theory of emotion. *Mind and Cognition: An Anthology*. Blackwell.

Hecker, von, U., Meiser, T., (2005). Defocused attention in depressed mood: Evidence from source monitoring. *Emotion, 5*(4), 456-463.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences, 6*(6), 242-247.

Hoffmann, H., & Möller, R. (2004). Action selection and mental transformation based on a chain of forward models. *Proc. of the 8th International Conference on the Simulation of Adaptive Behavior* (pp. 213-222). Cambridge, MA: MIT Press.

Hogewoning, E., Broekens, J., Eggermont, J., & Bovenkamp, E.G.P. (2007). Strategies for affect-controlled action-selection in Soar-RL. In: J. Mira and J.R. Álvarez (Eds.), *IWINAC 2007, Part II, LNCS 4528* (pp. 501-510). Springer-Verlag.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*, 237-285.

Lahnstein, M. (2005). The emotive episode is a composition of anticipatory and reactive evaluations. *Proc. of the AISB'05 Symposium on Agents that Want and Like* (pp. 62-69). AISB Press.

McCallum, A. (1995). Instance-based utile distinctions for reinforcement learning with hidden state. *Proc. of the Twelfth International Conference on Machine Learning* (pp. 387-395).

McMahon, A., Scott, D., Baxter, P. & Browne, W. (2006). An autonomous explore/exploit strategy. *Proc. of the AISB'06 Symposium on Nature Inspired Systems* (pp. 192-201). AISB Press.

Montague, P. R. Hyman S. E & Cohen J. D. (2004). Computational roles for dopamine in behavioural control. *Nature, 431*, 760-767.

Nason, S. & Laird, J. E. (2005). Soar-RL, integrating reinforcement learning with soar. *Cognitive Systems Research, 6*(1), 51-59.

Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C. Cavallo, D., Machover, T., Resnick, M., Roy, D. & Strohecker, C. (2004). Affective learning — A manifesto. *BT Technology Journal, 22*(4), 253-269.

Phaf, R. H., & Rotteveel, M. (2005). Affective modulation of recognition bias. *Emotion, 5*(3), 309-318.

Rolls, E. T. (2000). Précis of The brain and emotion. *Behavioral and Brain Sciences, 23*, 177-191.

Rose, S. A., Futterweit, L. R., & Jankowski, J. J. (1999). The relation of affect to attention and learning in infancy. *Child Development, 70*(3), 549-559.

Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological Review, 110*(1), 145-72.

Salichs, M.A., Malfaz, M. (2006). Using emotions on autonomous agents. The role of happiness, sadness and fear. *Proc. of the AISB'06 Symposium on Integrative Approaches to Machine Consciousness* (pp. 157-164). AISB Press.

Scherer, K. R. (2001). Appraisal considered as a process of multilevel sequential checking. *Appraisal processes in emotion: Theory, Methods, Research*. Oxford Univ. Press, New York, NY.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science, 275*(5306), 1593-1599.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proc. of the Seventh International Conference on Machine Learning* (pp. 216-224). Morgan Kaufmann.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press.

Schweighofer, N., & Doya, K. (2003). Meta-learning in reinforcement learning. *Neural Networks, 16*, 5-9.

Tyrell, T. (1993). *Computational Mechanisms for Action Selection*. PhD Thesis, University of Edinburgh.

van Dartel, M.F., & Postma, E.O. (2005) Symbol manipulation by internal simulation of perception and behaviour. *Proc. of the 5th International workshop on Epigenetic Robotics. Nara, Japan. Lund University Cognitive Studies, 123,* 121-124.

van Dartel, M., Postma, E. & van den Herik, J. (2004) Categorisation through internal simulation of perception and behaviour. *Proc. of the 16th Belgium-Netherlands Conference on Artificial Intelligence* (pp. 187-194).

Velasquez, J. D. (1998). A computational framework for emotion-based control. *In: SAB'98 Workshop on Grounding Emotions in Adaptive Systems.*

Ziemke, T., Jirenhed, D., & Hesslow, G. (2002). Internal simulation of perception: A minimal neuro-robotic model. *Neurocomputing, 68*, 85-104.

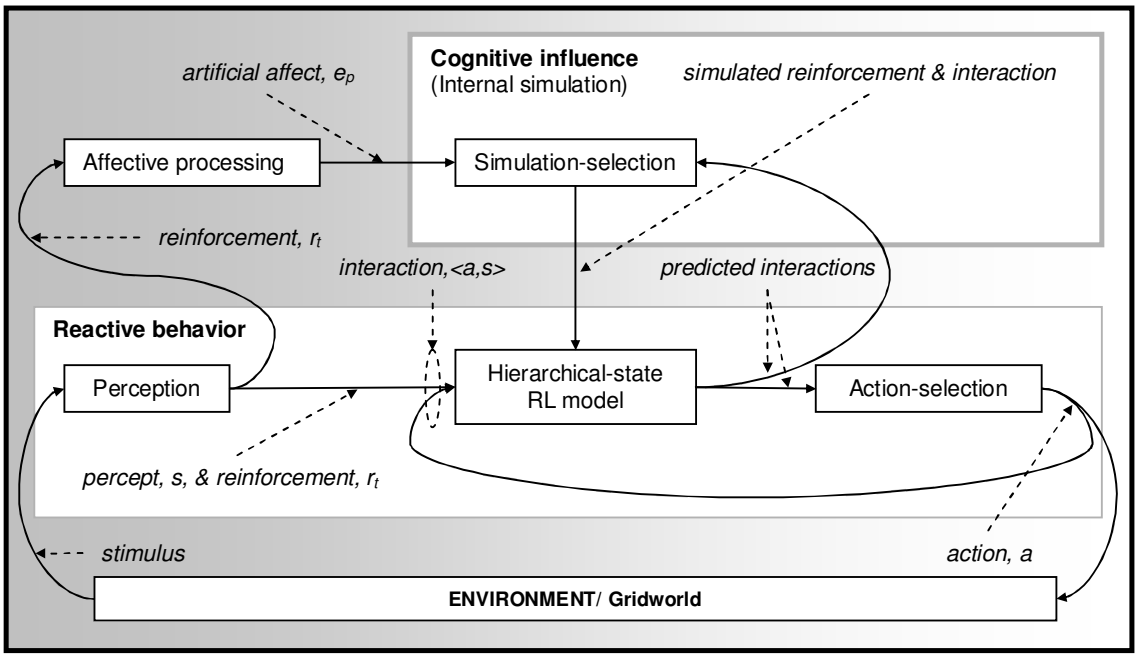

Figure 1. Overview of the different components in our model. Components are detailed below.

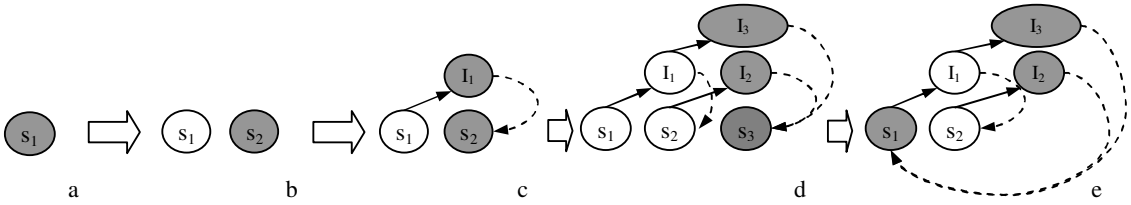

Figure 2a-e. Examples of the agent's memory structure

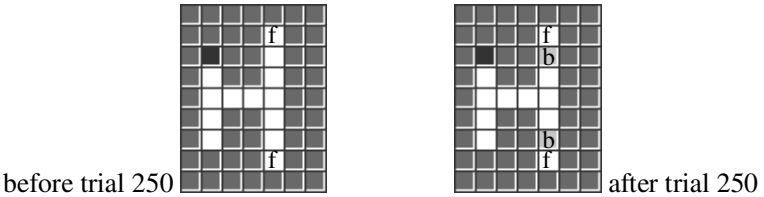

Figure 3. Switch-to-invest task. Potential start locations are alternated between the top-left and bottom-left arms, food locations (f) are alternated between the top-right and bottom-right arms, and roadblocks (b) are placed before the food after the task-switch.

Figure 4. Cue-inversion world. The first two and second two pictures show the possible worlds before and after the cue inversion at trial 250 respectively. f=food, c= cue, black square is the agent.

f:	*1*		*1.5*		*2*	
star:	*50*	*100*	*50*	*100*	*50*	*100*
ltar:	200	400	200	400	200	400
	250	500	250	500	250	500
	375	750	375	750	375	750
	500	1000	500	1000	500	1000
	750	1500	750	1500	750	1500

Table 1: Possible ltar, star, and f configurations.

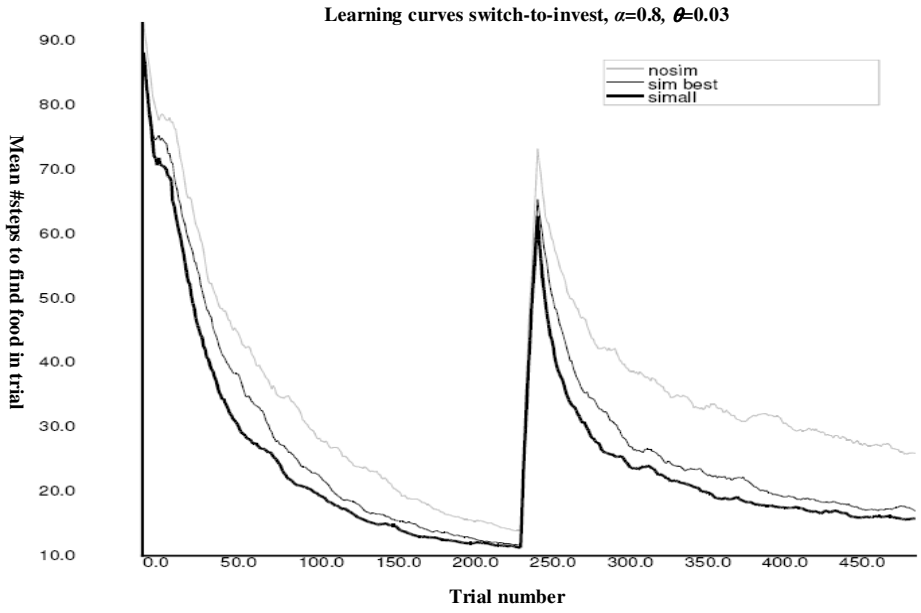

Figure 5. Smoothed learning curves (also see footnote 2) of non-, best-, and all-simulating agents in the switch-to-invest world for α=0.8, θ=0.03. Curves of other strategies are approximately in between best and all. Note that we do not use error bars in Figure 5. To validate our claims, we statistically compare between simulation strategies the effort involved in completing a run. This is appropriate; a small overall benefit can be considered important, regardless of the standard deviation over trails.

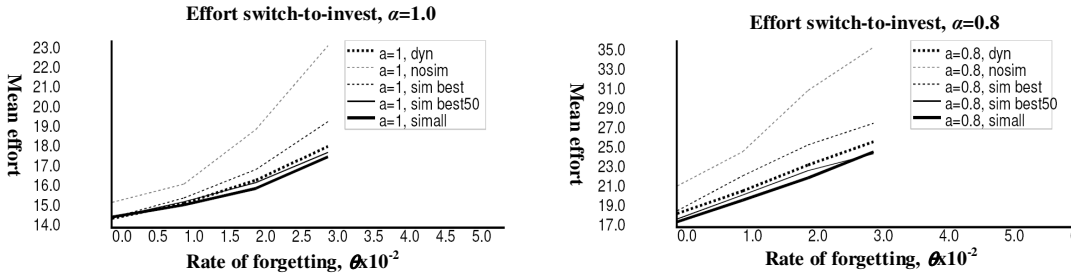

Figure 6. Effort for different simulation strategies in the switch-to-invest task.

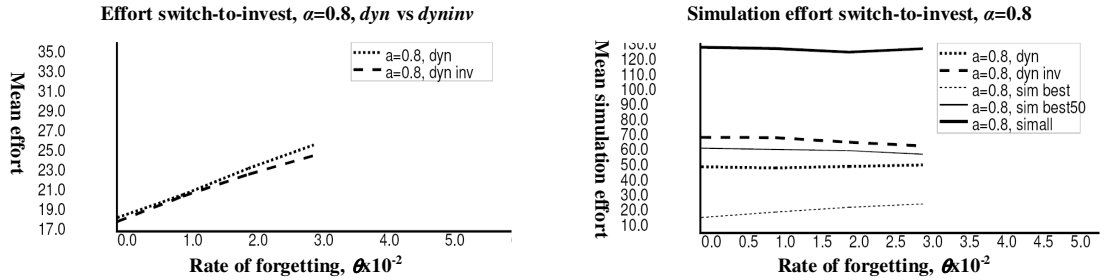

Figure 7. Left: small difference in effort between dynamic and inverse-dyn simulation strategies.

Right: difference in simulation effort between simulation strategies.

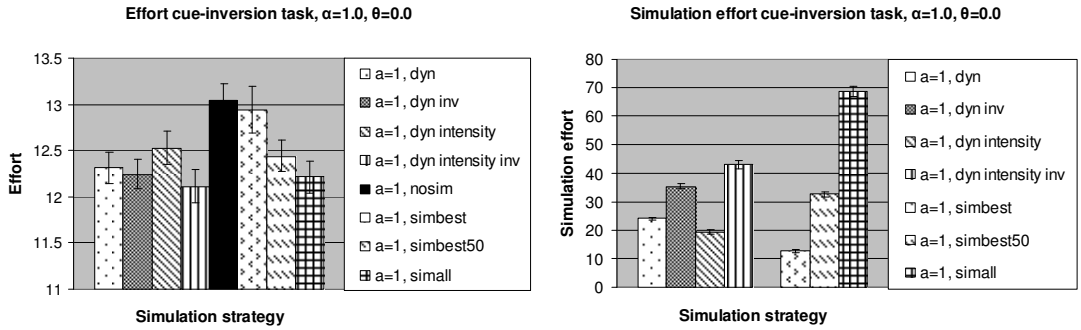

Figure 8. Left: difference (effort) between dynamic and static simulation strategies. Right: difference (simulation effort) between static and dynamic strategies. Error bars show 95% confidence interval.

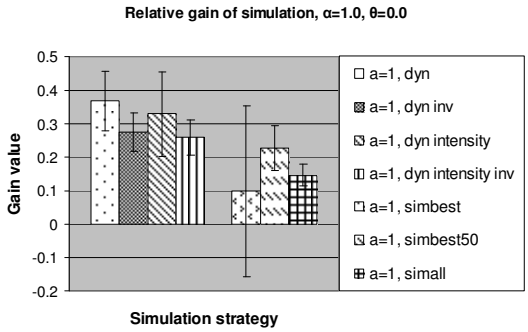

Figure 9. Gain of simulation strategies (details in text).Error bars show 95% confidence interval.

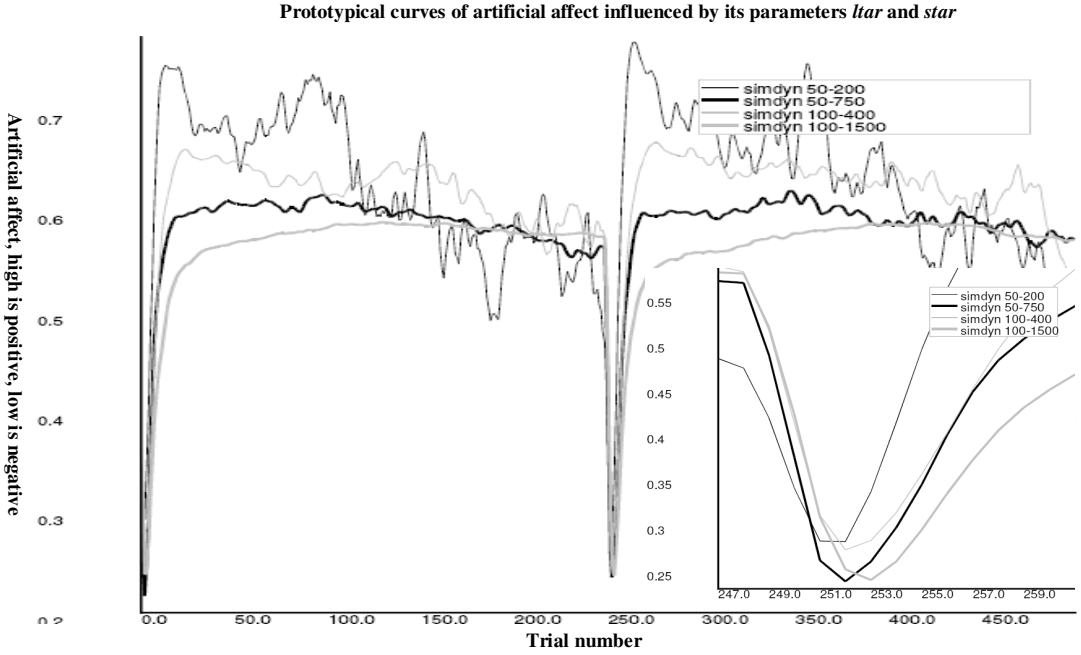

Figure 10. Depicted are affect curves for different settings (not smoothed). Inset is a detail of artificial affect at the cue inversion. Note that star=50 has the "dip" earlier than star=100.

Joost Broekens was born in 1976. He received the M.Sc. degree in computer science at Delft University in the Netherlands (2001), has worked for two years as software engineer, and obtained his Ph.D. degree in computer science at Leiden University on computational modelling of affect and learning (2007). His research interests include affective computing, (formal) cognitive modelling, emergent properties of information processing systems, reinforcement learning and human-in-the-loop robot learning. Currently, he is employed as artificial intelligence researcher at the Telematica research laboratory in the Netherlands.

Walter A. Kosters was born in 1957. He received the M.Sc. and Ph.D. degrees, both in mathematics, from Leiden University in 1981 and 1985, respectively. He is an assistant professor in computer science at Leiden University. His research interests include artificial intelligence, data mining and natural computing.

Fons J. Verbeek is born in Amersfoort, the Netherlands. He has a strong research interest in imaging, image analysis and heterogeneous data analysis in bio-medical related research areas; that is both of images and other bio-molecular data resources. He did his PhD at the Delft University of Technology (The Netherlands) in Applied Physics in the pattern recognition group on 3D image analysis. Currently, he is (co-)heading the Imagery and Media group at the Leiden Institute of Advanced Computer Science. Recently, he has developed an interest in Human Computer Interaction which now influences research conducted in his research section, i.e., Imaging & BioInformatics.