Affect and Learning: a computational analysis

Broekens, D.J.

Affect and Learning

A Computational Analysis

Joost Broekens

Affect and Learning

A Computational Analysis

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. P.F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 18 december 2007
klokke 16:15 uur

door

Douwe Joost Broekens

geboren te Beverwijk, Nederland in 1976

Promotiecommissie

Prof. dr. Joost N. Kok	promotor
Dr. ir. Fons J. Verbeek	copromotor
Dr. Walter A. Kosters	copromotor
Dr. Joanna J. Bryson	referent
Prof. dr. Bernhard Hommel	
Prof. dr. Thomas H. W. Bäck	
Prof. dr. Sjoerd M. Verduyn Lunel	

ISBN: 978-90-8891-0241

Contents

Chapter

1

Introduction

The research described in this computer science PhD thesis is positioned somewhere between computer science and psychology. It is about the influence of affect on learning. Affect is related to emotion; *affect* is about the *positiveness* and *negativeness* of a situation, thought, object, etc. We will define affect more precisely in Section 1.4, but for now this definition suffices. Affect can influence learning and behavior in many ways. For example, parents use affective communication to influence the behavior of their children (praise versus disapproval). Affect can also influence how individuals process information (e.g., positive affect favors creativity, negative affect favors critical thinking). The research described in this thesis uses computational modeling to study affective influence on learning. The goal has been twofold: first, understand more about potential mechanisms underlying relations between affect and learning as found in the psychological literature, and second, study if the concept of affect can be used in computer learning, most notably to control the learning process. Both aspects are considered of equal importance in this work. The topic is quite interdisciplinary and the individual chapters present the results of focused studies. However, in an attempt to clarify to a broader public what the research questions are and why these are of interest, the introduction is intentionally kept broad and is written so that it is understandable to readers with general knowledge of computer science and an interest in psychology. Readers that want to skip the introduction can read Section 1.5 for an overview of the thesis.

1.1 Informal Introduction to the Topic

This thesis is about affect and learning, a topic everyone is intuitively familiar with. We all know the effects of anger and sadness (two different negative affective states) or happiness and excitement (two different positive affective states) on our own functioning and decisions. Sometimes, we regret these decisions, while others worked out quite fine—better than expected—afterwards. In everyday life, we just accept that we have emotions and that our emotions influence our behavior. It is common sense knowledge that it is sometimes the head, sometimes the heart that decides our future and we rarely ask ourselves when and how affect exactly influences our decisions. Interestingly, it is quite difficult to reflect upon a decision, let's say the last decision you made, and discriminate between the "affect" part versus the "rational thought" part that influenced that decision. Instead, affect and "rational thought" seem to be intertwined in many cases, a notion put forward by Antonio Damasio in his seminal book *Descartes' Error* (Damasio, 1994). It is by now generally accepted that "rational thought" does not exist, at least not in the sense we thought (hoped?) it did. Nothing is decided purely based on a logical evaluation of pros and cons of which the pros and cons are again (recursively) a result of a logical evaluation of *their* pros and cons of which the pros and cons … etc. This kind of recursive and analytic thought process is very rare, chess-like game play being perhaps a partial exception, and it is by no means necessary for normal functioning in society; other animals don't need it either and are quite adaptive to their environment. What seems to be more the case is exemplified by the following "should I stay or should I go" scenario (also a nice song by *The Clash* showing human indecisiveness):

I'm at work, writing the introduction of my thesis. Some chapters still have to be written, so quite some writing still has to be done. However, today actually is a local holiday called "Leids Ontzet" feasting the liberation of the city of Leiden (The Netherlands) ending the Spanish occupation of that city in the year 1574. The faculty is closed but I went in with my key to do some work. So, the decision is: should I stay the whole day and write as much as possible, or should I go home and do something else taking advantage of the fact that today is a local holiday. Now here's my "rational choice": I went to work this morning, because I am not originally from Leiden, so I do not really care about Leids Ontzet. My partner also went to work, because she works in The Hague (not in Leiden: thus no local holiday). I do care about playing video games in my spare time, and therefore I like having a day off. However, I do have a lot of work to do on my thesis, and I want my thesis to be finished in time. (Why? Because my supervisor wants me to? Because it is good for my future? Because it just feels like the right thing to do?). So, here am I, having to decide on two things: go home and play games (which I like), versus stay and write my thesis (which I like). It is a holiday, but my partner has to work. So, taking a day off *now* enables me to play games, but I won't be able to work on my thesis, and it takes away my option to take another day off when my partner does have a day off. What do I do? I work in the morning on my thesis, write a fair part of the introduction, and take the afternoon off and play games. I get to do two things I like, and keep the option of taking half a day off to do nice stuff with my girlfriend later, which I also like. So isn't this a win-win-win situation? It probably is, but the decision itself is not rational, it is emotional and social and there is no deep logical evaluation behind the value of the alternatives. The only thing that might be called rational is the process by which I generate the alternatives. However, the decision is made based on a "what feels best" criterion, and I just "weight" the values of the alternatives using social and emotional associations. One could even argue that I did not decide anything at all: none of the alternatives is excluded; instead I have chosen a mixture of things that feels good to me. Many decisions resemble this scenario, and I think we can agree that our life's course is a long sequence of such decisions, none of them being exclusively rational, none exclusively affective.

The question seems to be how and when affect influences decision making, thought, learning, and the many other cognitive phenomena known in cognitive psychology. For example, psychologists like Joseph Forgas, Alice Isen and Gerald Clore have studied the influence of emotion and affect on human decision making for quite some time (for references see Chapter 2). Although much debate is going on, as discussed for example in Chapter 3, decades of research indeed converged into a general consensus that affect *does* influence cognition in important ways. These ways include affect manipulating how we approach problems—e.g., do we look at the details of a problem, or approach it from the top—, affect influencing what we think about objects and people, and affect influencing creativity and open-mindedness.

Although much is known on the influence of affect on cognition, the mechanisms by which affect influences cognition are largely unknown. This is partly because it is very difficult to experimentally manipulate and subsequently measure affect, let alone affective influence on, for example, decision making and learning. This is exactly where the computer enters (fortunately, as this is a Computer Science PhD thesis, and some might at this point be wondering where the computer went). Computers enable scientists to develop computational models (programs) that can actually produce "new things", based on the assumptions of the theoretical model (e.g., a psychological theory describing the influence of affect on learning) underneath the computer model. These "new things" are, in a very real sense, predictions of the psychological theory: they result from the computational model that is a highly detailed version, an implementation, of the psychological theory. As such, computational models help psychological theory development. As computer models need to "run", they need to execute a sequence of commands and manipulate the results of these commands; computer models are particularly good at investigating mechanism, because they exist by the virtue of mechanism. Mechanism happens to be the thing that is notoriously difficult to investigate based on observation of behavior (whether that is body movement, data from brain scanners, facial expressions, or biochemical markers). We can thus conclude that computational modeling is a useful method to study potential mechanisms proposed by psychological and neurobiological theories, including theories about the influence of affect on learning.

This thesis presents research on the influence of affect on learning by means of computational modeling. As such, both affect and learning need to be computationally modeled. A successful model for task-learning is *reinforcement learning* (RL). It has been applied to many computer learning problems, such as computers that learn to play games, steer cars, and control robots (see Sutton &

Barto, 1998). The RL paradigm is quite analogous to instrumental conditioning. Instrumental conditioning is a paradigm by which animals (including humans) can learn new behaviors, by trying new actions (exploration) and receiving rewards and punishments (reinforcement) for these actions. Rewards and punishments can transfer to the actions the animal chose to do just before the action resulting in the reinforcement, and to actions before *that* action, and before *that* action, etc. As a result, the animal learns to execute a sequence of actions in order to get to a reward or avoid a punishment; the animal is said to exploit its knowledge after a period of exploration of its environment. Reinforcement Learning is a detailed computational model that describes how reinforcement can propagate back to earlier actions (this process of propagation is also known as *credit assignment*), as well as how the *values* of actions need to be adapted to reflect the received reinforcement (Section 1.3). Recently, neuroscientists have found evidence that parts of the human brain (and brains of other animals) seem to be involved in exactly this process of reward processing. The basal ganglia (an important dopamine system in the brain responsible for the initiation of action) are involved in the selection of actions, and neurons in the basal ganglia seem to encode the reinforcement signal, i.e., the change that needs to be made to the value of an action. Neurons in the prefrontal cortex (responsible for planning and executive, reflective processing) seem to encode the value (i.e., the effective credit a certain action is responsible for) of actions in a certain context. In studying learning, Reinforcement Learning seems to be a good candidate model; a point of view that is detailed in Section 1.3.

In Chapter 2 we introduce a measure for artificial affect that relates to a simulated animal's relative performance on a learning task (let's say, a simulated mouse in a maze searching for cheese). As such, artificial affect measures how well the simulated animal improves. Our animal learns by reward and punishment, thus, in our case, how "well" can be defined as the average reinforcement signal. Therefore the animal's performance can be defined as 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). Artificial affect is a measure for how good or bad the situation of the agent is.

In this thesis we explore, among other things, how affect can be used to influence learning by controlling when to *explore* versus *exploit*. As mentioned earlier, animals need to sometimes explore their environment, sometimes exploit the knowledge they have of that environment. Simulated animals also need to do so. To learn where the cheese is, learn different routes to the cheese, learn alternative cheese locations, adapt to new cheese locations, etc., a simulated mouse sometimes needs to explore (to find new stuff) and sometimes needs to exploit (to eat cheese). Controlling exploration versus exploitation is an important problem in the robot learning domain. By using artificial affect to control exploration, and by coupling artificial affect to affect in the psychological literature, an important step is made towards autonomous control of learning behavior in a way compatible with nature. We show that in some cases it is indeed beneficial [1] to the learning simulated animal to control exploration and exploitation by means of artificial affect.

A second aspect explored in this thesis is how affect can be used to control learning more directly, much like a parent that approves or disapproves of a child's behavior. We study, using a simulated robot, the effect of a human observer parenting a robot "child". The robot has to learn a certain task, and the human observer can approve or disapprove the robot's actions by expressing emotional expressions to a camera. The expressions are analyzed in terms of positive and negative affect and fed to the learning robot. This reinforcement signal is used to train the robot, in addition to the normal reinforcement signals given to the robot by the environment it behaves in. We show that learning can improve[2] if such social-based feedback is added to the learning mechanism.

1.2 Computational Models, Psychology and Artificial Intelligence.

Before entering the specifics of the research described in this thesis, a short introduction into the relation between computational models, psychology and artificial intelligence is useful. Computers can be used to model many different phenomena and systems. For example, weather forecasts in fact result from computational (mathematical) models that simulate interaction patterns between the different elements that constitute "the weather", such as air pressure, wind speeds, land elevation, etc. So in essence, a weather forecast is a prediction of the "theory of the weather" by means of a computational model of that theory. In the same spirit, computational models exist that are inspired by, based on, or explicitly implementing psychological theories. Depending on the level of fidelity to the theory, the model can be used to gain insights into, and potentially predict consequences of the psychological theory.

On the other hand, natural theories (such as psychological, economical and biological ones), once implemented, can be very useful in the computer science domain itself. Consider, for example, the Traveling Salesman Problem (TSP), a

[1] Beneficial in terms of (1) effort involved (steps) in finding solutions, and (2) more rewarding solutions.

[2] Improvement in terms of quicker learning of the solution to the task at hand.

typical computational problem defined by finding the shortest route (or at least a route shorter than an arbitrary given length *K*) that visits all locations from a set of locations exactly once (e.g., a traveling salesman that wants to travel from city to city in the most efficient way). TSP is an *NP*-complete problem. In short, this means that to check *if* a given route is a solution to a certain instance of the TSP problem (meaning that the route addresses all locations and is shorter than length *K*), a polynomial number of calculations is needed[3]. Checking a solution is easy in terms of time needed for checking. However, *finding* the shortest route (or deciding if a route shorter than *K* exists) generally takes an exponential amount of calculations, so finding the best route is difficult. This is due to the fact that the number of possible routes that exist between a set of locations grows exponentially with the number of locations. The number of possible routes becomes extremely large even for a small number of locations. An exact solution (i.e., the best route) to this problem is often unnecessary for a real salesman, and for large sets of locations practically impossible. Biologists have studied the behavior of ants intensively and found that ants have an interesting way to find shortest routes to food by leaving scent trails that grow stronger every time an ant uses the same route and finds food at the end. By doing so, ant colonies as a whole have evolved a practical, approximate solution (a.k.a. *heuristic)* to the problem of finding shortest paths. Currently, much research is being done on ant-colony-based heuristics to find practical solutions to, e.g., the Traveling Salesman Problem (Dorigo & Stützle, 2004). This example shows that natural theories can inspire the search for solutions to problems in computer science.

Computational models can thus be used to simulate real-world phenomena, and theories about the real-world can inspire the search for solutions to computer science problems, a notion underlying natural computing in general (Rozenberg & Spaink, 2002). Let's specifically look at the role of computational models in psychology, as well as the role of psychology in computer science.

[3] Polynomial in this context means that the number of calculations needed is expressible in terms of a power over the size of the problem. So, given *n* locations, checking if a route addresses all locations could take, e.g., n^2 calculations, denoted as $O(n^2)$, the complexity *order* is called quadratic. Note that for TSP, there are representations of the problem for which the order for checking a solution is actually $O(n)$: compare if the route contains all *n* locations; sum over all route's segments to obtain the route's length *L* and compare if $L < K$. Note also that the size of a TSP instance is not measured in terms of the number of locations, but in terms of the number of possible location transitions (the potential to move from one city to another); it is not relevant to the complexity of the problem how many locations there are, but in how many ways one can address them all. A polynomial number of calculations is assumed to be *tractable* ("easy" to solve), while an exponential number of calculations (expressible as an exponent, not as a power) is *intractable* ("hard" to solve).

Psychological theories often establish relations (correlations, effects, causality) between different aspects of the human mind and observable behavior. Such relations are often found using sophisticated psychological tests that measure the relation between different *constructs*. A construct is a measurable theoretical abstraction for a certain characteristic, e.g., the construct "intelligence" measured with an IQ test representing the level of non-specific skills a person has. Relations between constructs can be shown in different ways. Most commonly used are the experimental approach aimed at:

- causality; measure construct *A*, do something to construct *B*, than measure construct *A* again to find out if *B* influenced *A* in some way,
- correlation; measure both *A* and *B* at the same time and try to find a correlation between both, and
- longitudinal effects; measure *A* at intervals for a period of many years, manipulate *B*, and try to find trends in *A* over time.

Of course, these approaches exist with or without control groups, with or without blind and double blind setups, and so on.

Aimed at understanding the human mind, psychologists want to study not only relations between constructs but also want to understand the mechanisms responsible for these relations; a notoriously difficult goal, as experimenters cannot look in detail in a persons head. Clever experiment designs have by now been developed that aim at looking into the mind. An impressive example of this can be found in the cognitive psychology domain, e.g., in the domain of working memory and attention. To investigate a relatively simple question such as "can a person attend to, and process two different stimuli at the same time", extremely complex experiment designs have been developed to answer it; not because this is fun, but because the answer must be interpretable in terms of an underlying mechanism. In concrete terms this means that, if the answer is, for example, "yes, persons can do that", the following questions immediately pop up. How many tasks can we simultaneously execute? What task-load is permissible? What if one of the tasks is a heavy one and the other is not, and would performance on the latter be compromised? What if one of the tasks is personally relevant? What if one of the tasks was a task the person is trained on, and to what extent can tasks be executed simultaneously under the assumption that they are indeed trained? How much training is needed? These questions are not so much questions about relations anymore, but in fact questions about mechanisms such as "how does working memory capacity function?", "how is context switching executed by the human brain?", and "how do we concentrate (what *is* concentration)?". The experiment designs needed to study such questions are extremely complex, and very hard to grasp in terms of their consequences for the conclusions (e.g., didn't we forget to control for this or that phenomenon). This is what makes experimental psychology such a difficult and challenging scientific enterprise, for which strong research methods, many different theories and exact reporting of results are critical.

Fortunately (especially for computer science graduates with a strong interest in psychology in search for a topic for their PhD thesis), psychology has added a new type of experiment to their research weapon arsenal, a weapon specifically targeted at understanding mechanism: computer simulation. Computational models need to be specified at a detailed level. As such, in order for a model to execute, mechanism details have to be filled in. If this filling in is done based on a psychological theory, the model becomes a more detailed version of that theory. By executing a computational model, it can provide insights into possible mechanisms underlying the relations between constructs. More importantly, if a psychological theory already proposes potential mechanisms, the computational model can predict consequences of these mechanisms, thereby helping to refine the theory.

Interesting examples include neural network models of human working memory and attention (Dehaene, Sergent & Changeux, 2003), but also the many computational models of emotion based on cognitive appraisal theory that have been implemented in computer systems. Cognitive appraisal theory assumes that emotions result from an individual's cognitive evaluation of the current situation in terms of his or her goals and knowledge. Evaluation is often assumed to be symbol manipulation. As computers are good at such systematic symbol manipulation, this type of theory has been immensely popular as basis for computational models of emotion in (simulated) robots. The development of computational models based on cognitive appraisal theory advances cognitive appraisal theory by refining them (Broekens & DeGroot, 2006; Wehrle & Scherer, 2001). Assumptions in the theory need to be made explicit when used in a computer program.

On the one hand, computational modeling is useful to psychology, while on the other, as we will see now, psychology is useful to computer science, most notably to the field of artificial intelligence.

Broadly speaking, *Artificial Intelligence* (AI) (Russell & Norvig, 2003) studies how computer programs can solve problems, inspired by how nature (including animals, cells, molecules, etc.) solves problems. Intelligence in AI is a vast concept. It includes reactive behavior of autonomous robots aimed at solving concrete problems (e.g., simulated ants in the traveling salesman problem heuristic mentioned above), adaptive stock-price prediction software, and symbolic reasoning processes aimed at transport and military operations planning. In AI, a computer program (the mechanism used to simulate nature) is also defined in a broad way. A program in AI can range from a collection of preprogrammed algorithms that execute planning routines to find optimal planning solutions in advance (e.g., planning an optimal route for a transport company), to reward-based learning mechanisms that continuously adapt their input-output behavior such that the robot they are controlling is able to learn new tasks. So, AI is not exclusively about robots, nor is every robot intelligent. AI is not exclusively about putting loads of knowledge in a database and programming an algorithm that reasons over that knowledge, nor is every knowledge base intelligent. And, to do away with another common misconception: the grand aim of Artificial Intelligence is not about creating intelligence that is artificial as in "fake", "dumber than real", and "superficial", it is about studying the processes and mechanisms of intelligence using artificial means, such as digital computers. If there is a common grand "creational" aim then this would be to develop intelligent, autonomous systems that are able to think and act for themselves, in a way that reflects the wit and cunning of natural intelligence.

Many of the techniques used in AI directly come from other disciplines, such as neuroscience, psychology and biology. For example, artificial neural networks are based on the work by the neuropsychologist Donald Hebb (1904-1985), who described the learning process of neurons in terms of the correlation between pre- and post-synaptic firing, now called *Hebbian learning*. If two neurons are connected through a synapse, and both the pre-synaptic neuron A (exciting neuron B) and the post-synaptic neuron B (excited *by* A) activate (fire) at about the same time, the strength of the connection is increased, thereby increasing the probability that neuron A excites B in the future. This model underlies many of the learning mechanisms implemented in artificial neural networks, but also underlies connectionist learning models in general.

Another, more specific, example is the application of *Soar* in the area of computer games research as well as medical image analysis. Soar (originally for State, Operator And Result) is a cognitive architecture aimed at problem solving through rule matching. It is based upon the idea of a unified theory of cognition, proposed by Newell (1990), integrating theories of cognition from many different disciplines. Key elements of Soar are its ability to plan for, reason about and act upon a situation using rule matching in recursive thought cycles. In every cycle, all rules that apply to the current situation activate. The activation strength of a rule depends on how well the rule matches the current situation. The most strongly activated rules are allowed to propose new "facts", such as actions that can be executed by the robot controlled by the Soar program. If no rules activate based on the current situation, a new “problem” is created, and Soar tries to recursively solve this problem. Once the problem is solved, Soar creates a new rule for future use, solving that problem more efficiently should it pose itself again. This architecture, proposed as a symbolic theory of cognition, has been used to build intelligent computer game agents that predict what other agents (e.g., the user) will do (Laird, 2001). In the medical domain it is currently being used in image analysis software agents: specialized programs responsible for analyzing a specific type of information in an image to coordinate, e.g., analysis of coronary plaque images (Bovenkamp et al., 2003).

We have seen that computer science—specifically artificial intelligence—and psychology—specifically cognitive psychology—are fields that strongly influence each other in many ways. This influence dates from the very early 1950’s. Alan Turing’s (1950) well-known paper on machine intelligence was published in *Mind*, a psychological and philosophical journal, at about the same time as the seminal papers that started the cognitive revolution in psychology. Donald Hebb (1949) presented such a clear description of how brains learn that this opened up an information processing view of the mind. The mechanisms he described have by now been applied in robotics and AI many times.

Most important to this thesis are the concepts *affect* and *instrumental conditioning*. Instrumental conditioning underlies Reinforcement Learning (RL) (Sutton & Barto, 1998), a method that has proven to be critical for artificial task-learning. As we have used RL as a model for learning in our research, it is one of the cornerstones of our approach. We devote the next section to it. We use artificial affect to influence learning. Therefore, affect is the second cornerstone. We devote Section 1.4 and Chapter 2 to the latter topic.

1.3 Learning, Instrumental Conditioning, Reinforcement Learning.

Animals learn behavior in a variety of ways, such as by imitation, by play, and by trial and error. Instrumental conditioning is the more formal name for learning behavior by trial and error. For example, rats learn to push buttons or pull levers in order to receive food. To learn this behavior they have to try actions *before* they know the result of that action. It could be that pushing a button results in the rat being punished. As there is no way to know this beforehand, the rat has to try to push the button, at least for the first time. After pushing it, the rat either receives food, or some kind of punishment (e.g., a loud sound). The animal learns to repeat the actions that lead to food, and avoid actions that lead to punishment.

This is called instrumental conditioning (see Anderson, 1995): learning to repeat or avoid actions in a certain situation, based on reward and punishment.

Interestingly, many animals learn to execute sequences of actions. To take our rat example, the rat not only learns to push the button for food, it also learns to walk to the button *after* having looked around for the button *after* having entered the specific rat-maze room in which the button is located, etc. By reinforcing a certain situation-action couple, not only the last action is influenced, but also the sequence of environment-rat interactions leading to that reinforcement. Further, this sequence is better learned if it is repeated. So, repetition of a sequence of interactions ending with reinforcement enables the rat to learn that sequence better and better. The same mechanisms can account for many goal-directed behaviors of humans. We rarely do something without having received rewards, and by training we become better at it. Sometimes the reward is indirect, such as in the case of money. It is straightforward to argue that money has become a reinforcer by itself because humans have associated it with more natural reinforcers (Anderson, 1995), such as food (restaurants, candy), play (vacation, toys) and social interaction (having a drink with friends, going to the theatre or a rock concert, distributing candy at school). We learn to work (a long sequence of actions) for money, because money gives us naturally reinforcing stuff.

Finally, *discounting* is a concept of critical importance: rewards and punishments in the future are perceived as less important than in the here and now. Animals discount the value of reinforcement, dependent upon the time passed between administration of the reinforcement and the action to be reinforced. As a result, reinforcement most strongly influences the action executed just before receiving the reinforcement.

In this section we will see that the machine learning concept of Reinforcement Learning is a very good model for instrumental conditioning.

1.3.1 Reinforcement Learning

Strongly related to instrumental conditioning, there is a form of machine learning called *Reinforcement Learning*. Reinforcement Learning (RL) (Sutton & Barto, 1998) is a computational framework describing how in an environment appropriate actions can be learned purely based on exploration and reinforcement. Actions are appropriate if they maximize some signal from the environment, say a reward. As such, RL, is a particular computational model of instrumental conditioning[4]. A formal description of RL is the problem of learning a function

[4] Dayan (2001) and Kaelbling, Littman and Moore (1996) discuss some of its limitations.

that maps a state to an action, such that, given a certain history of state-action transitions, for all states this mapping results in an action that yields the highest cumulative future reward as predicted by that history of state-action transitions. In normal language this means that RL attempts to recognize the best possible action in a situation, given a certain amount of experience.

We have used Reinforcement Learning as a basis for learning in this thesis. The main reason for this choice is that RL maps very well to animal task learning (instrumental conditioning). The second reason is that RL has proven to be the most successful paradigm for the machine learning of tasks composed of multiple actions that are not known in advance. Other forms of learning, such as supervised learning, need a human observer. RL does not, it learns by trial and error, providing a clear benefit: a RL system learns autonomously. This is important for, e.g., robot learning. By investigating the relation between RL and affect, we hope to advance a well known machine learning paradigm as well as shed some light on the potential relation between affect and learning.

In essence, RL aims at solving the *credit assignment* problem (Kaelbling, Littman & Moore, 1996). That is, how much credit should an action get, based on its responsibility for receiving current and future rewards; in other words, how should an action in a certain situation be valued given its immediate reward as well as all rewards that might follow? Note that from now on we will talk about *reward* when we mean reinforcement. Reward can thus be positive and negative. A classical representation of a function that represents a solved credit assignment problem is a 2-dimensional table with cells representing the value of all actions in all possible states, rows representing states, and columns representing actions (Table 1.1). If this table is used for control, i.e., to select actions for execution by a simulated animal, the current observed state is used as row entry, and the action belonging to the cell with the highest value on that row is selected. For example, if the simulated animal would be in state *choice* (Figure 1.1), the best action to perform is *left*. When the action *left* is executed a state change occurs, and the next state is observed by the simulated animal; *food* in our case. Now the process of action-selection can be repeated.

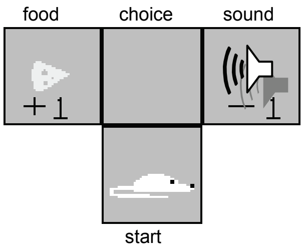

Figure 1.1 The maze solved by the function depicted in Table 1.1. The cheese has a reward of +1, while the loud sound has a reward of −1. States are called food, choice, sound, start for “mouse at cheese”, “mouse at junction”, “mouse in sound room”, and “mouse at start”. This is a four-state problem.

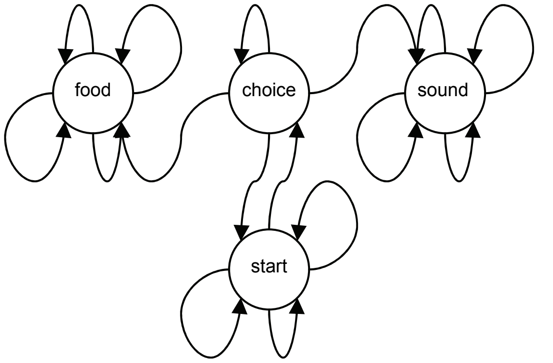

Figure 1.2 A state-transition diagram for the maze presented in Figure 1.1. Arrows denote move actions. Probabilities are assumed to be equal to 1 (i.e., choosing, for example, *up* always results in the state pointed to by the up-arrow). The states, *sound* and *food* are terminal states.

	left	*right*	*up*	*down*	*eat*
start	0.125	0.125	0.25	0.125	0
choice	0.5	−0.5	0.25	0.125	0
food	0	0	0	0	1
sound	−1	−1	−1	−1	−1

Table 1.1 The classical representation of a function that solves a specific credit assignment problem, in our case food-finding in a simple maze (Figure 1.1). The discount factor, γ, equals 0.5. So the importance of future rewards drops with a factor of 2 for every step in between an action and a reward. States are called *food*, *choice*, *sound*, *start* for “mouse at food”, “mouse at junction”, “mouse in sound room”, and “mouse at start” respectively. We assume that when the mouse arrives at food or sound, it can not exit that place by itself. We further assume that moving outside the maze does not result in a state change. This table presents the solution to our four-state problem.

At an architectural level, the RL problem can be formally described as follows. It consists of a set of states, *S*, a set of actions, *A* and a transition function $T : S \times A \times S \rightarrow [0,1]$ defining how the world changes under the influence of actions giving the probability *T*(*s*, *a*, *s'*) that action *a* in state *s* results in state *s'*, where the sum over all *s'* of *T*(*s*, *a*, *s'*) equals 1. Further, a reward function $R : S \times A \rightarrow \Re$ and a value function $V : S \rightarrow \Re$ are defined. The states *S* contain representations of the world perceived by the *agent*, such as a *start* state, a *food* state etc. Note that from now on we use the term *agent* to refer to a simulated animal or robot. The actions *A* contain all possible actions the agent can execute, such as *left*, *right*, *up*, *down* and *eat*. The transition function defines the probability of ending up in one state, assuming a current state, *s* and action, *a*. So, the agent’s world is probabilistic[5]. The reward function defines the reward for a certain action, *a*, when executed in state, *s*. The value function maps a state, *s*, to a cumulative future reward. So, if an agent knows *T* and *V* the optimal next action can be selected using:

[5] but stationary, i.e., the probabilities do not change (Kaelbling et al., 1996).

$$a^* = \arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s') \right), \text{ with } \gamma \text{ the discount factor (1.1)}$$

The best action a^* is the action with the highest sum of immediate reward *R*(*s*, *a*) and value predictions *V*(*s'*), over all possible next states *s'* resulting from action a^*, weighted according to their probability of occurrence *T*(*s* ,*a* ,*s'*). Note that the summation in formula (1.1) is needed as in a probabilistic world multiple states *s'* might result from action *a*. In our example, moving *up* in state *start* would be the best action, because *R*(*start*, *up*) + 0.5*T*(*start, up, choice*)*V*(*choice*) = 0 + 0.5*1*0.5 = 0.25, which is the highest value (we assume that the probability of ending up in state *choice* after executing *up* in state *start* equals 1, so in our case we only have one possible next state *s'* after executing action *up* in state *s*). However, to select this action we have to know both *V*(*choice*) and *T*(*start*, *up*, *choice*).

Solving the credit assignment problem has thus become a question of learning the value function *V*, together with the transition function *T*. The main question is, how? The short answer is: by trial and error; try actions in states, record the received reward and the resulting state, and update both *V* according to the reward, as well as *T* according to the probability of arriving in that new state. The longer, formal answer is: by value propagation according to the following formula:

$$V(s) \leftarrow \max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s') \right) \qquad (1.2)$$

which is equivalent to $V(s) \leftarrow val(a^*)$, with *val*(a^*) the value of action a^*

The formula updates the value for state *s* with the immediate reward *R*(*s*, *a*) and discounted future values *V*(*s'*) for all *s'* possibly resulting from action *a* weighted according to the probability *T*(*s*, *a*, *s'*) that transition *s*→*s'* occurs due to action *a*. Again, action *a* is chosen such that it is the best one possible. This enforces conversion of values to the highest possible value attainable by the agent.

By now, many different version of RL exist that all solve the credit assignment problem in a slightly different way (for a dated but excellently written overview, see Kaelbling et al., 1996). In general there are two different types of RL approaches; *model-based* and *model-free*. Model-based approaches have (or learn) a model of the world that consists of a (probabilistic) state transition structure (Figure 1.2). Model-based approaches thus have a function *T*. The research in Chapter 3 and 4 is based on model-based RL. Model-free approaches do not have such a world model. Model-free approaches thus need to learn *V* in a different way, as they do not possess the function *T* while this function is needed for value propagation as described in formula (1.2). The research in Chapter 6 is based on model-free RL.

In the model-free case, *V* can be learned in the following way. It can be shown (Singh, 1993) that the following formula converges to an optimal value function *V*, if a sufficient and unbiased amount of exploration occurs during learning, and the learning rate *α* is gradually decreased from 1 (in the beginning of learning) to 0 (at the end of learning):

$$V(s) \leftarrow V(s) + \alpha\big(r + \gamma V(s') - V(s)\big), \text{ with } r \text{ the reward} \qquad (1.3)$$

It is quite well possible to intuitively grasp this without proof. If an agent has an infinite amount of time to keep trying things in a world, it eventually bumps infinitely many times into all possible situations that exist in that world. This means that it will see the transitions *s*→*s'*,*s''*,… for all *s* many times. Every such transition updates *V*(*s*) a little bit, so together *V*(*s*) accumulates the results of all these transitions. It correctly estimates the value of *s* by sampling a representative number of transitions resulting from *s*. So, an agent (or real animal for that matter) has to explore—i.e., sample a representative number from all possible interactions with the environment—to be able to learn a useful value function. After exploration, the agent can use the learned value function to act, i.e., the agent can exploit its knowledge. The *exploration – exploitation tradeoff* is a very important issue in Reinforcement Learning (Sutton & Barto, 1998). Without a good mechanism to decide when to explore versus exploit, RL cannot learn an optimal value function.

It is important to note here that there are ways in which an artificial agent can learn an optimal interaction model (in terms of maximizing cumulative reward). One of these is to let the agent first explore a large amount of time, and then switch to an exploitation mode. However, this is not plausible from a natural point of view. No animal can afford to purely explore, as this is just too risky. In our learning models (Chapter 3 to 6), we take this into account. We have no separate exploration – exploitation phases; our agents learn the value and transition function while at the same time using these for action selection (called *certainty equivalence*, see Kaebling et al, 1996). Our agents thus assume that their world model is a correct estimation of the world they interact with.

1.3.2 Reinforcement Learning as a Model for Instrumental Conditioning

As mentioned before, one of the main reasons for using Reinforcement Learning (RL) as learning mechanism in studying the interplay between affect and learning is that RL very well models instrumental conditioning. RL models instrumental conditioning in at least three important ways.

- First, it associates rewards with the probability of execution of actions in a certain situation, as in instrumental conditioning. The simulated animal learns to repeat actions based on an association between reward and action.
- Second, by repetition the learned association becomes more accurate, and as such the probability to execute actions that result in reward becomes larger (positive reward) or smaller (no reward, or punishment).
- Third, the learned value for a situation can influence the execution of actions in earlier situations. We thus see that RL provides an answer to how sequences of actions can be learned by trial and error: propagate the reward through the sequence back to the beginning such that the right amount of credit is given to the individual actions in the sequence.

Recently, the mechanism of Reinforcement Learning has been tied to neural substrates involved in instrumental conditioning. For example, there are strong links between dopamine brain systems and RL (Dayan & Balleine, 2002; Montague, Hyman & Cohen, 2004; Schultz, Dayan & Montague, 1997). It seems that neurons in these regions encode for the RL *error signal*, i.e., the change to the expected value of a situation, $\Delta V(s)$. More recently Foster and Wilson (2006) showed that awake mice replay in reverse order behavioral sequences that led to a food location; a crucial finding for the above mentioned link. It suggests that mice can replay sequences backward from the goal location to the start location. This is a mechanism that would be needed to speed up value propagation back to the beginning, and is highly compatible with the RL concept of *eligibility traces* (Foster & Wilson, 2006). An eligibility trace (for details see Sutton & Barto, 1998) is a state sequence leading to a certain reward or punishment. In RL, eligibility traces can be used to speed up learning. The idea is to update the complete sequence based on that reward (such a sequence represents a trace of situations that is eligible for the resulting reward). In RL, updating the value of states in this trace can be done in any order. In nature, backwards is more plausible than forwards for the following reason. Assumed that the brain is a connectionist architecture primarily learning by means of Hebbian mechanisms, in order for two situation representations to transfer a characteristic (e.g., reward) between each other, both have to be active at the same time. If a state sequence is replayed backwards, pair-wise activation of two consecutive states, for all states in the sequence starting at the end, would in principle suffice to (partly) transfer the reward to the start of the sequence. However, for any other order to get the same value propagation result, it would need either massive repetition of activated pairs of representations or activation of all pairs at the same time. So, activation of the state sequence from the end, back to the beginning seems more efficient than any other order[6]. It is therefore interesting to see that mice seem to indeed replay *in reverse order* the "states" they visited while walking towards the food.

Finally, animal learning by trial and error closely matches RL in how experience of the world is built up: by means of a sufficient number of interaction samples to build up the value function. Trials are samples from all possible interactions with the environment; errors (rewards) change the value and reward functions learned by the animal. If an animal is a good explorer, it will be better at finding optimal solutions because it samples more possibilities from the environment, therefore the animal's resulting value function has more chance to better estimate the real value function. On the other hand, exploration is risky: if you don't know what the result will be, you could die. Animals that do not explore will stick to their current interaction pattern. This means that as long as the interaction pattern is appropriate for the environment they are in, they will do better than explorers: they don't waste time exploring useless options while they have a good option available. However, as soon as the environment changes, they will die because of the useless option and the lack of exploration. To learn a good value function, a sufficient amount of exploration is needed. So, also in real life, the tradeoff between exploration and exploitation is important. Actually it is much more important in real life, as one stupid action can result in death or illness, while in a simulated world it only results in a negative reward. A second difference is that in real life one can not afford to have a pure exploration phase: this would most certainly result in at least one very stupid action, hence death. As a result, the exploration – exploitation tradeoff is even more important. Both have to be in balance for an agent to survive. In Chapter 3 and 4 we explore to what extent artificial affect can be used to control the exploration - exploitation tradeoff. We have based these studies on how affect influences learning in humans, a topic introduced in the next section, and in more detail in Chapter 2. In order to stay consistent with nature, we do not separate exploration - exploitation phases.

Although from this description it seems that RL has been used primarily to simulate learning animals, this is not the case. RL has been widely used to learn computers to play games (e.g., Tesauro, 1994), to control cars to autonomously drive based on visual input (e.g., Krödel & Kuhnert, 2002) and to control robots (e.g., Theocharous, Rohanimanesh & Mahadevan, 2001).

[6] Interestingly, value propagation in RL is in the same direction, that is, backwards.

1.4 Emotion, Affect and Learning

In this thesis 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). Therefore we first explain the meaning we will use for these terms. 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 the set of phenomena, and—at least in humans—feelings and cognitive appraisal are too (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. For example, a car that passes you in a dangerous manner on the freeway, immediately (*time*) elicits a strongly negative and highly arousing (*affect*) emotional state that includes the expression of anger and fear, feelings of anger and fear, and intense cognitive appraisal about what could have gone wrong. On the contrary, learning that one has missed the opportunity to meet an old friend involves cognitive appraisal that can negatively influence (*affect*) a person's mood for a whole day (*time*), even though the associated emotion is not necessarily arousing (*affect*). Eating a piece of pie is a more positive and biochemical example. This is a bodily, emotion-eliciting event resulting in mid-term moderately-positive affect. Eating pie can make a person happy by, e.g., triggering fatty-substance and sugar-receptor cells in the mouth. The resulting positive feeling is not of particularly strong intensity and certainly does not involve particularly high or low arousal, but might last for several hours.

We use affect to denote the *positiveness* versus *negativeness* of a situation. In the studies reported upon in this thesis 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).

Emotion plays an important role in thinking, and evidence is abundantly available. 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 behaviors. Normal emotional functioning appears to be necessary for normal behavior.

Emotion[7] influences thought and behavior in many ways. Emotion can be a motivation for behavior. Emotion is related to the urge to act (e.g., Frijda & Mesquita, 2000): run away when in danger, fight when trapped, laugh and play when happy. Specific emotions trigger specific behaviors (e.g., fight or flight). So, emotion is not only related to the *urge* to act, some emotions—when strong enough—make us really act.

Emotion and feelings influence how we interpret stimuli, how we evaluate thoughts while solving a problem (Damasio, 1996) and how we remember things. A person's belief about something is updated according to emotions: 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). As shown by the “should I stay or should I go” scenario presented earlier in this introduction, we often decide to do something based on how that option feels to us.

Finally, 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; Gasper & Clore, 2002; Forgas, 2000; Phaf & Rotteveel, 2005). As a result, positive affect relates to a “forest” or goal-oriented look (we interpret what we see in the context of our existing knowledge), while negative affect relates to a “trees” or exploratory look (we critically examine incoming stimuli as they are).

Several psychological studies support that enhanced learning is related to positive affect (Dreisbach & Goschke, 2004). Others show that enhanced learning is related to neutral affect (Rose, Futterweit & Jankowski, 1999), or to both (Craig, Graesser, Sullins & Gholson, 2004). Although much research is currently being carried out, it is not yet clear how affect is related to learning in detail.

In this thesis we computationally address this issue: in what ways can affect influence learning. We do not model categories of emotions nor use emotions as

[7] An emotion is different from a feeling. A feeling is in essence your mental representation of yourself having the emotion.

information in symbolic-like reasoning. So the research goal has not been to investigate how agents can reason “emotionally”, such as in the work by Marsella and Gratch (2001), or interact emotionally with humans (Heylen et al, 2003).

1.5 Questions Addressed and Thesis Outline.

To study the influence of affect on learning, in a Reinforcement Learning setting, we first have to evaluate whether affect can be used in this context: we have to define affect in a Reinforcement Learning context. In Chapter 2 we define artificial affect in detail. In Chapter 3 to 6 we study three different ways in which affect can influence learning, where learning in each chapter is modeled using a different variation of RL.

In Chapter 3 we investigate how artificial affect can control exploration versus exploitation. As the amount of exploration strongly influences learning behavior, and as it has been found (e.g., in the studies mentioned earlier) that affect relates to broad (explore) versus narrow information (exploit, goal directed) processing, we have investigated how artificial affect can control exploration versus exploitation in agents. A simulated “mouse” in a grid-world maze can either search for “cheese” (eating cheese is its goal) by trying actions it does not know the consequences for (explore), or use its model of the environment it has built up so far in an attempt to walk to the cheese by trying actions it thinks it knows the consequences for (exploit). We couple artificial affect to exploration and exploitation in different ways, according to studies reported by Dreisbach & Goschke (2004) and Rose et al. (1999): positive affect increases exploration (and negative affect increases exploitation) and vice versa. In RL terms, we use artificial affect as meta-learning parameter (see also Doya, 2002) to control exploration versus exploitation by dynamically coupling it to the greediness of the *action-selection* function responsible for making this choice (the β parameter of the Boltzmann distribution, in our case). A meta-learning parameter is a parameter that influences learning, but does not contain information about the task to be learned per se, e.g., the choice to explore versus exploit, or the speed with which to forget knowledge you had acquired. We use a version of RL that is similar to *Sarsa* (Rummery & Niranjan, 1994; Sutton, 1996). The main findings are that (1) both negative affect and positive affect can be beneficial to learning, and (2) negative affect seems to be related to less selective decisions while positive affect is related to more selective decisions.

In Chapter 4, we investigate the influence of affect on thought. Instead of studying the influence of artificial affect on action-selection in a purely reactive agent, we now study the influence of artificial affect on “thought selection” in a more cognitive agent. In our study, we have defined thought as internal simulation of potential behavior, according to the S*imulation Hypothesis*, proposed by Hesslow (2002) and Cotterill (2001). This process of simulation uses the same brain mechanisms as those used for actual behavior. For example, if I consciously think of going home and play games, I, in a sense, go home and do so without moving my body. Simulating going home thus enables me to evaluate how I feel about going home by triggering the same brain areas and processes that would have been triggered if I went home and started playing. This again enables me to decide whether I should do it or not, showing that simulation could be useful for decision making and action selection. We have developed a variation to the model-based RL paradigm, called *Hierarchical State Reinforcement Learning*, which enables us to study this question. We computationally investigate, again using a grid-world setup, the influence on learning efficiency when artificial affect controls the amount of internal simulation. Artificial affect is dynamically coupled to the greediness of the *simulation-selection* mechanism responsible for selecting potential actions for internal simulation. As such we model affective modulation of the amount of thought during a learning process. The main findings are that (1) internal simulation has an adaptive benefit and (2) affective control reduces the amount of simulation needed for this benefit. This is specifically the case if 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. Thus, agents “feeling positive” can think ahead in a narrow sense and free-up working memory resources, while agents “feeling negative” are better off thinking ahead in a broad sense and maximize usage of working memory.

In Chapter 5 we discuss related and future work in the context of the studies presented in Chapter 3 and 4.

In Chapter 6, we investigate how affect can be used to influence behavior of others. Emotion and affect are important social phenomena. One way in which affect is important socially is that it enables effective parenting. Affect communicated by a parent can be seen as a reinforcement signal to a child. In this chapter we investigate the influence of affect communicated through facial expressions by a human observer on learning behavior of a simulated “child”. We thus investigate the effect of parenting a simulated robot using affective communication. Two important differences exist between the study in this chapter and those in Chapters 3 and 4. First, we use a continuous (non-discrete) grid-world setup, use real-time interaction between the robot and the human “parent”, and use a specifically developed neural-network approach to Reinforcement Learning applicable to this context. This has been done to match real-world learning problems more closely. Second, we use affect in a different way. In Chapter 3 and 4, we use artificial affect as defined in Chapter 2; i.e., a long-term signal originating from the simulated agent, used by the simulated agent to control its own learning-parameters. In contrast, in the experiments reported in Chapter 6, we use affect as a short-term signal related to emotion, originating from an observing "parent" agent, used to influence the reinforcement signal received by the simulated robot. The main finding is that the simulated robot indeed learns to solve its task significantly faster (measured quantitatively) when it is allowed to use the social reinforcement signal from the human observer. As such, this chapter presents objective support for the viability and potential of human-mediated robot-learning.

In Chapter 7, we take a theoretical approach towards computational modeling of emotion. We present a formal way in which emotion theories can be described and compared with the computational models based upon them. We apply this formal notation to cognitive appraisal theory, a family of cognitive theories of emotion, and show how the formal notation can help to advance appraisal theory and help to evaluate computational models based on cognitive appraisal theory: the main contributions of this chapter. Although this chapter is quite different from the others, it fits within the general approach: that is, the use of computational models to evaluate emotion theories.

1.6 Publications

A revised version of Chapter 3 has been published in (Broekens, Kosters & Verbeek, 2007). Parts of Chapter 4 have already been published earlier (Broekens, 2005; Broekens & Verbeek, 2005), while Chapter 4 is a slightly revised version of the article by Broekens, Kosters & Verbeek (in press). Chapter 6 has been published in (Broekens & Haazebroek, 2007), while an extended and revised version has been published as a book chapter in (Broekens, 2007). Earlier versions of the work in Chapter 7 have been published (Broekens & DeGroot, 2004c; Broekens & DeGroot, 2006), while a revised version of Chapter 7 is published in (Broekens, Kosters & DeGroot, in press).

2

Artificial Affect

In the Context of Reinforcement Learning

In this chapter we present the rationale for the concept of emotion used in the studies reported upon in Chapter 3 and 4, that is, positive and negative affect. We first review different findings on the interplay between emotion and cognition, after which we describe several ways in which affect influences learning, the main phenomenon investigated computationally in this thesis. Finally we introduce a measure for artificial affect, and argue for its validity in the context of Reinforcement Learning.

2.1 Emotion and Behavior Regulation

Emotion influences thought and behavior. For example, 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 "classical" cognitive and instrumental learning phenomena.

Emotion is related to the regulation of adaptive behavior and to information processing. 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, so at this level of information processing one type of emotional regulation of behavior has already been established, i.e., approach (rewarded behavior) versus avoidance (punished behavior).

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 related to the continuous checking of the environment for important stimuli. More resources are allocated to further evaluate the implications of an event, only if the stimulus appears important enough. 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 information processing strategy used.

Emotion also regulates behavior of others. Obvious in human development, expression (and subsequent recognition) of emotion is important to communicate (dis)approval of the actions of others. This is typically important in parent-child relations. Parents use emotional expression to guide behavior of infants. Emotional interaction is essential for learning. Striking examples are children with an autistic spectrum disorder, typically characterized by a restricted repertoire of behaviors and interests, as well as social and communicative impairments such as difficulty in joint attention, difficulty recognizing and expressing emotion, and lacking of a social smile (for review see Charman & Baird, 2002). Apparently, children suffering from this disorder have both a difficulty in building up a large set of complex behaviors *and* a difficulty understanding emotional expressions and giving the correct social responses to these. This disorder provides a clear example of the interplay between learning behaviors and the ability to process emotional cues

To summarize, emotion can be produced by low-level mechanisms of reward and punishment, and can influence information processing. As affect is a useful abstraction of emotion (see Section 1.4), these aspects inspired us to study (1) how artificial affect can result from an artificial adaptive agent's reinforcement signal, and (2) subsequently influence information processing in a way compatible with the psychological literature on affect and learning. In the next section we present some of the psychological findings related to the latter. In Section 2.3 we introduce the measure of artificial affect we have used in the studies reported upon in Chapter 3 and 4.

2.2 Learning is Influenced by Positive and Negative Affect

The influence of affect on learning is typically studied with the following psychological experiment. 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, affect induction (positive or negative change in a subject's affect due to, e.g., presented images) is assumed to be responsible for this effect, ergo; affect influences the execution of the cognitive task.

Some studies find that non-positive 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, neutral affect correlated with faster learning. Attention mediated this influence. Neutral affect related to more diverse attention, i.e., the babies' attention was "exploratory", and both neutral affect and diverse attention related to faster learning. Positive affect resulted in the opposite of neutral affect (i.e., slower learning and "less exploratory" attention). This relation suggests that positive affect relates to exploitation and neutral affect relates to exploration. Additionally, Hecker and von Meiser (2005) suggest that attention is more evenly spread when in a negative mood. This could indicate that negative affect is related to exploration.

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 press a button A or a button B 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. The authors 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 both positive and negative affective states can help learning but perhaps at different phases during the process, a point explicitly made by Craig et al. (2004). Chapter 3 and 4 of this thesis address exactly this issue. We use simulated adaptive agents to study the influence of artificial affect on learning performance, by controlling several learning parameters. In Chapter 3 artificial affect controls the amount of exploration versus exploitation used by the agent: affect controls the greediness of the action-selection mechanisms. In Chapter 4 artificial affect controls the greediness of its thoughts. In the latter study, an agent can internally simulate a number of interactions before actually executing these. Internal simulation can increase or decrease the likelihood of choosing a particular action, as it biases the value of the next actions (much like a person who imagines the potential results of a certain action, and who decides not to do it because of the imagined consequences). Some of these anticipated possibilities seem good, others do not. Affect is used to control the extent to which the selection of these simulated interactions is biased towards simulating only the positive ones (narrow, greedy "optimistic" thoughts) or towards simulating all anticipated possibilities (broad, evenly distributed thoughts).

2.3 Artificial Affect

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 reward *r*. Here we explain how our agent's artificial affect is linked to this reward *r*.

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 reward and affect follow the same rule: *if it's predicted it isn't important*.

The formula we use for artificial affect is:

$$e_p = (r_{star} - (r_{ltar} - f\sigma_{ltar}))/2f\sigma_{ltar} \qquad (2.1)$$

Here, e_p is the measure for affect. If e_p=0, we assume this means negative[1] 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 running-average reinforcement signal, r_{ltar}, with *ltar*

[1] Low and high values of e_p should not be interpreted as depressed and elated respectively. We assume that we model moderate levels of positive and negative affect, as induced by typical psychological affect-induction studies. Clinical depression and elatedness have different influences on behavior that are out of scope, and are too complex for our current modeling approach.

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 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.

By defining artificial affect purely in terms of rewards and punishments, one could argue that we interpret affect in a too narrow sense, thereby hollowing out the concept. 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 and punishment. Finally, as rewards and punishments define what behavior an artificial agent should pursue and avoid, reinforcement *is* the definition of good and bad for such agents. We therefore believe our measure for artificial affect is firmly grounded.

3

Affect and Exploration

Affect-Controlled Exploration is Beneficial to Learning

Recent studies show that affect influences and regulates learning. We report on a computational study investigating this. We simulate affect in a probabilistic learning agent and dynamically couple affect to its action-selection mechanism, effectively controlling exploration versus exploitation behavior. The agent's performance on two types of learning problems is measured. The first consists of learning to cope with two alternating goals. The second consists of learning to prefer a later larger reward (global optimum) to an earlier smaller one (local optimum). Results show that, compared to the non-affective control condition, coupling positive affect to exploitation and negative affect to exploration has several important benefits. In the Alternating-Goal task, it significantly reduces the agent's "goal-switch search peak". The agent finds its new goal faster. In the second task, artificial affect facilitates convergence to a global instead of a local optimum, while permitting to exploit that local optimum. Our results illuminate the process of affective influence on learning, and furthermore show that both negative affect and positive affect can be beneficial to learning. Further, our results provide evidence for the idea that negative affect is related to less selective decisions while positive affect is related to more selective decisions.

3.1 Introduction

As we have seen in Chapter 1 and 2, emotions influence thought and behavior in many ways. In this chapter we focus on the influence of affect on learning and adaptation. The main question we address here is: how is an agent's learning performance influenced if artificial affect is used to control exploration versus exploitation. Based on findings from the affect-cognition literature (Craig, Graesser, Sullins & Gholson, 2004; Dreisbach & Goschke, 2004; Rose, Futterweit & Jankowski, 1999) as discussed in Chapter 2, we hypothesize two types of relations between affect and exploration. The first type relates positive affect to exploitation, and negative affect to exploration. The second type uses the inverse relation of the first type, i.e., positive affect relates to exploration while negative affect relates to exploitation. We contrast these two dynamic settings to a non-affective control group of agents that use a static amount of exploration.

We investigate the relation between affect and learning with a self-adaptive agent in a simulated grid world. The agent acts in the grid world—in our case a simulated maze that represents a psychological task—and builds a model of that world based on perception of its surroundings and received rewards. Our agent autonomously influences its action-selection mechanism—the agent's mechanism that proposes next actions based on the learned model. The agent uses artificial affect, as defined in Chapter 2, to control the randomness of action selection. This enables the agent to autonomously vary between exploration and exploitation.

Our agent learns (adapts) using a simple form of Reinforcement Learning. The agent learns by constructing a Markov Decision Process (MDP), of which the state-value pairs are learned using a mechanism based on model-based Reinforcement Learning (Kaelbling, Littman & Moore, 1996). We investigate the hypothesized relations between affect and exploration using two different learning tasks (modeled as discrete grid worlds). In the first task the agent has to cope with a sudden switch from an old goal in one arm of a two-armed maze to a new goal in the other arm. We call this task the Alternating-Goal task. The second task consists of learning to prefer a later larger reward (global optimum) to an earlier smaller one (local optimum). We call the second task the "Candy task"; candy represents the local optimum being closest to the agent's starting position, while food represents the global optimum being farther away from its starting position.

From a learning and adaptation point of view, these tasks represent two significant problems for an agent. The Alternating-Goal task exposes an agent to a changing set of goals. The agent has to modify its behavior in order to reflect a change in this set of goals. It has to be flexible enough to give up on an old goal and learn a new one, while at the same time it has to be persistent enough to continue trying an active goal in order to actually learn the path to the goal (Dreisbach & Goschke, 2004). In other words, to cope with alternating goals, the agent has to decide when to explore its environment and when to exploit its knowledge; a.k.a. the exploration-exploitation problem or tradeoff (Kaelbling, Littman, & Moore, 1996). In our task, failure to solve this problem results in huge goal-switch cost (if the agent does not explore the environment after the goal-switch has taken place) and/or slow/unstable convergence (if, after exploration, the agent does not exploit its learned new model of the environment).

The Candy task represents searching for a global optimum, while exploiting a newly found local optimum. This ability is important for adaptive agents as it enables them to survive with the knowledge they have, while trying to find better alternatives. Failure to do so results in getting stuck in local optima or slow convergence. This again represents a tradeoff between persistence and flexibility, but different from the tradeoff in the first task. Now, the agent has to autonomously decide that the current goal *might* not be good enough and search for a better goal. In contrast, in the previous task the old goal attractor (high reward) is removed and the agent should react to this by searching for a new goal.

In this study we use artificial affect as defined in Chapter 2, that is, artificial affect is a measure for how well the agent is doing compared to what it is used to, based on an analysis of the difference between a long-term and a short-term reinforcement signal average. In the next section we explain our experimental method, i.e., how we implemented the two different relations between affect and action selection mentioned earlier, the grid-world setup, the tasks, the agent's learning mechanism and our experimental setup. In Section 3.3 we present experimental results. Section 3.4 discusses these results in a broader context.

3.2 Method

To investigate the influence of affect-controlled exploration, we did experiments in two different simulated mazes. Each maze represents a task, and we compared affect-controlled dynamic exploration to several control conditions with static amounts of exploration.

3.2.1 Learning Environment.

The first task is a two-armed maze with a potential goal at the end of each arm (Figure 3.1a). This maze is used for the Alternating-Goal task, i.e., coping with two alternating goals: find food or find water (only one goal is active during an individual trial, goal reward r = +2.0). The second maze has *two active* goal locations (Figure 3.1b). The nearest goal location is the location of the candy (i.e., a location with a reward r = +0.25), while the farthest goal location is the food location (r = +1.0). This maze is used for the Candy task. The walls in the mazes are "lava" patches, on which the agent can walk, but is discouraged to do so by a negative reinforcement (r = –1.0).

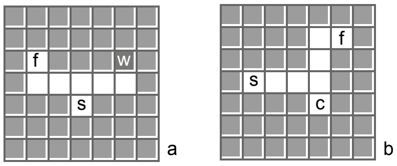

Figure 3.1. Mazes used in the experiments; (a) the Alternating-Goal task, (b) the Candy task; the 's' denotes the agent's starting position, 'f' is food, 'c' is candy and 'w' is water.

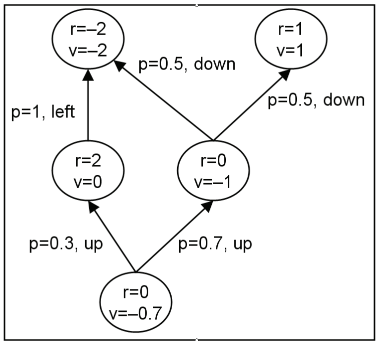

Figure 3.2. Example of a Markov Decision Process. Nodes are agent-environment states. Edges are actions with the probability *p* that executing the action results in the state to which the edge points. Nodes contain rewards (*r*; local reinforcement) and values (*v*; future reinforcement). In this example γ=1 (see text).

The agent learns by acting in the maze and by perceiving its direct environment using an 8-neighbor and center metric (i.e., it senses its eight neighboring locations and the location it is at). An agent that arrives at a goal location is again placed at its starting location. Agents learn a probabilistic model of the actions and their values possible in the world. Mathematical details of this process follow; however, the most important part of our method is explained in Section 3.2.2. Agents start with an empty model of the world and construct a Markov Decision Process (MDP) as usual (i.e., a perceived stimulus is a state *s* in the MDP, and an action *a* leading from state s_1 to s_2 is an edge in the MDP; see Figure 3.2; for details see Sutton & Barto, 1998). The agent counts how often it has seen a certain state *s*, *N*(*s*). It uses this statistic to learn the value function, *V*(*s*) (comparable to model-based Reinforcement Learning, see, e.g., Kaelbling et al., 1996). This function learns to predict a cumulative future reward for every observed state. This *V*(*s*) is learned in the following way. A reward function, *R*(*s*), learns to predict the local reward of a state:

$$R(s) \leftarrow R(s) + \alpha \cdot (r - R(s)) \qquad (3.1)$$

This learned reward is used in the value function *V*(*s*):

$$V(s) \leftarrow \gamma \sum_i \left(\frac{N(s_{a_i})}{\sum_j N(s_{a_j})} V(s_{a_i}) \right) + R(s) \qquad (3.2)$$

So, a state *s* has two reinforcement-related properties: a learned reward value *R*(*s*) and a value *V*(*s*) that incorporates predicted future reward. The *R*(*s*) value converges to the local reward for state *s* with a speed proportional to the learning rate α. The final value of *s*, *V*(*s*), is updated based on *R*(*s*) and the weighted predicted rewards of the next states reachable by actions a_i. In Reinforcement

Learning, the discount factor γ defines how important future versus current reward is in the construction of the value function, $V(s)$. If the discount factor, γ, is equal to 1, future reward is important (no discount), while $\gamma = 0$ means that only local reward is important for the construction of the value of a state as expressed by $V(s)$. In the Alternating-Goal task the learning rate α and discount factor γ are respectively 1.0 and 0.7, and in the Candy task respectively 1.0 and 0.8.

3.2.2 Modeling Action Selection.

Most relevant to the current study is that our agent uses the Boltzmann distribution to select actions based on learned values of predicted next states. This function is often used in Reinforcement Learning and is particularly useful as it enables both exploration and exploitation:

$$p(a) = \frac{\exp[\beta \times V(s_a)]}{\sum_{i=1}^{|A|} \exp[\beta \times V(s_{a_i})]} \qquad (3.3)$$

Here, $p(a)$ is the probability that the agent chooses action a, and $V(s_a)$ is the value of a next state predicted by action a. $|A|$ is the size of the set A containing the agent's potential actions[1]. Importantly, the *inverse* temperature parameter β determines the randomness of the distribution. The larger the β the more this distribution adopts a greedy selection strategy (thus little variation in deciding what action to perform in a certain state). If β is zero the distribution function adopts a uniform random selection strategy, regardless of the predicted reward values (thus high variation in deciding what next action to perform in a certain state).

De facto, the β parameter can be used to vary the adaptive agent's processing strategy between exploration and exploitation. Note that we define exploration as generating new learning experiences by selecting actions that are non-optimal according to the current model the agent has learned, while exploitation is defined as selecting optimal actions according to the currently learned model. Therefore, if we assume, for simplicity, that the model is a tree with the agent's starting state as root and edges as different actions to different next states, exploration generates different paths through the tree at different runs, while exploitation

[1] Note that for notational simplicity we assume that an action in one state leads to a determined next state, i.e., the world is deterministic and completely observable. However, our first world is not deterministic as we introduce a for the agent non-predictable goal-switch.

retries the same paths at different runs. In a lazy value propagation mechanism as ours, exploration is needed to find solutions, while exploitation is needed to internalize solutions. Exploitation thus models animal learning by repetition, while exploration models animal search.

Key in our study is that our agent uses its artificial affect e_p to control its β parameter. Affect directly and dynamically controls exploration versus exploitation. This approach is compatible with viewing emotion as a mechanism for meta-learning (Doya, 2000; Doya, 2002; Schweighofer & Doya, 2003).

3.2.3 Type-A: Positive Affect Relates to Exploitation

To investigate how affect can influence exploration versus exploitation, we hypothesize the following two relations. First, type-A agents model positive affect related to increased exploitation:

$$\beta = e_p \times (\beta_{\max} - \beta_{\min}) + \beta_{\min} \qquad (3.4)$$

If affect e_p increases to 1, β increases towards β_{max} and as e_p decreases to 0, β consequently decreases towards β_{min}. So positive affect results in more exploitation, while neutral and negative affect results in more exploration, as suggested by the study by Rose et al. (1999), detailed in Chapter 2. This is also compatible with the idea that positive mood relates to top-down processing (Gasper & Clore, 2002), i.e., in our case to the agent using its learned model to control its behavior. A selective mode of action selection uses this model to drive behavior, while a less selective mode could be said to use more diverse behaviors (whether or not this also models bottom-up processing is unclear).

3.2.4 Type-B: Negative Affect Relates to Exploitation

The second relation is the inverse of the first one. Type-B agents thus model positive affect related to increased exploration. Positive affect favors detaching actual behavior from existing goals (as suggested by the results of the study by Dreisbach and Goschke (2004):

$$\beta = (1 - e_p) \times (\beta_{\max} - \beta_{\min}) + \beta_{\min} \qquad (3.5)$$

As affect e_p increases to 1, β decreases towards β_{min} and as e_p decreases to 0, β consequently increases towards β_{max}. So, positive affect results in more exploration, while negative affect results in more exploitation.

Of course, cognitive set-switching and attention are not equivalent to learning. Both are a precursor to learning, specifically explorative learning. Divided attention and flexible set-switching enable an individual to faster react to novel situations by favoring processing of many external stimuli. So, in the study by Dreisbach and Goschke (2004) *positive* affect facilitated exploration, as it helped to remove bias towards solving the old task thereby enabling the subject to faster adapt to the new task. However, in the study by Rose, Futterweit and Jankowski (1999) neutral affect facilitated exploration as it related to defocused attention.

3.2.5 Experimental Procedure

To investigate the influence of affect-controlled exploration, our experiments are repeated with agents of type-A and type-B as well as a control condition of agents that use static levels of exploration versus exploitation (fixed β). In the Alternating-Goal task agents first have to learn goal one (food). After 200 trials the reinforcement for food is set at $r = 0.0$, while the reinforcement for water is set at $r = +2.0$. The water is now the active goal location (so an agent is only reset at its starting location if it reaches the water). This reflects a task-switch, of which the agent is unaware. It has to search for the new goal location. After 200 trials, the situation is set back; i.e., food becomes the active goal. This is repeated 2 times resulting in 5 phases, i.e., initial learning of food goal (phase 0), then water (phase 1), food (2), water (3), and finally food (4). This (5 phases, a total of 1000 trials) represents 1 run. We repeated runs to reach sufficient statistical power. All Alternating-Goal task results are based on 800 runs, while Candy task results are based on 400 runs. During a run, we measured the number of steps needed to get to the goal (steps needed to end one trial), resulting in a learning curve when averaged over the number of runs. We also measured the average β (resulting in an “exploration-exploitation” curve), and we measured the quality of life (QOL) of the agent (measured as the sum of the rewards received during one trial). The problem for the agent is to exploit the goal but at the same time “survive” a goal switch, i.e., keep the switch-cost as low as possible. So, the learning curve of the trials just after the task-switch indicate how flexible the agent is.

The setup of the Candy task experiment is simpler, and we measured the same (steps, β and QOL). The agent has to learn to optimize reward in the Candy maze. The problem for the agent is to (1) exploit the local reward (candy), but at the same time (2) explore and then exploit the global reward (food). This relates to opportunism, an important ability that should be provided by an action-selection mechanism (Tyrell, 1993). Average QOL curves will thus show to what extent an agent has learned to exploit the global reward.

Our independent variable is the type of exploration-exploitation control. We have several different settings of type-A ("*dyn*" in Figure 3.3-3.11) and type-B ("*dyn inv*" in Figure 3.3-3.11) affect-controlled exploration. For example, "AG dyn 3-6" means that the agent was tested in the Alternating-Goal task using affect controlled exploration of type-A (positive affect relates to exploitation) with exploration-exploitation varying respectively between β_{min}=3 and β_{max}=6 (see also Figure 3.3). The artificial affect parameters *star* and *ltar* defining the short-term period and the long-term period over which artificial affect is measured were set at 50 and 375 steps respectively. As a control condition we used agents with different static amounts of exploration ("*static*" in Figure 3.3-3.11). High static β values model low exploration and high exploitation while low values denote high exploration and low exploitation. The legend of Figure 3.7 shows all different agents used in the Alternating-Goal task. Figures 3.3-3.6 show relevant subsets of these agents. The legends of Figures 3.8-3.11 show all agents used in the Candy task, excluding static agents with β=5 and β=7. The results from these two agents did not add anything to the analysis and are therefore omitted.

3.3 Results

We now discuss the results of the experiments. A discussion in a broader context is presented in Section 3.4.

3.3.1 Experiment 1: Alternating-Goal Task

Our main finding is that type-A (positive affect relates to exploitation, negative to exploration) results in the lowest switch cost between different goals, as measured by the number of steps taken *at the trial in which the goal switch is made* (Figure 3.7). This is an important adaptation benefit. As shown, all goal-switch peaks (phases 1-4) of the 4 variations of type-A (i.e., dotted lines labeled AG dyn 3-6, 3-7, 3-9 and 2-8) are smaller than the peaks of the control (straight lines labeled AG static 3, 4, 5, 6 and 7) and type-B (i.e., striped lines labeled AG dyn inv 3-6, 3-7, 3-9 and 4-9). Initial learning (phase 0) is marginally influenced by affective feedback and by static β settings (Figure not shown). Closer investigation of the first goal switch (trial 200; phase 1; Figure 3.4) shows that the trials just after the goal-switch also benefit considerably from type-A. When we computed for all settings an average peak for trail 200, 201 and 202 together, and compared these averages statistically, we found that type-A performs significantly better (p<0.001 for all comparisons, Mann-Whitney, n=800). Closer investigation of the fourth goal-switch (trial 800, phase 4; Figure 3.5), reveals a different picture. Only the trial in which the goal is switched benefits significantly from type-A (p<0.001 for all comparisons except those mentioned shortly, Mann-Whitney, n=800).

Comparison between type-A (AG dyn 3-6, 3-7 and 3-9) and AG static β=6 showed significant smaller peaks for type-A with p<0.01, p<0.05, and p<0.01 respectively. So it seems that a high static amount of exploration performs slightly better at later goal switches but worse at earlier goal-switches as compared to affective control over exploration. One reason for this is that the agent has built up a very good model of both arms of the maze in these later phases. This means that in later phases, less exploration is needed anyway, because the agent only needs to relearn to take the right choice at the T-junction, but not learn the new arm in the maze. This limits the potential gain of affective control. This explanation is supported by the peak curves in Figure 3.7. Here, higher β values perform worse than lower at the peaks of earlier phases but better at the peaks of later phases. Note that for the first phase, this is also true, but as we plot only the first trial after the goals-switch in Figure 3.7 this is not shown (it *is* shown in Figure 3.4, where we detail the peak of the first phase, high β values show higher peaks than do low β values).

All other comparisons between peaks revealed significantly (p<0.001) smaller peaks for type-A. This effect is most clearly shown for the peaks of phase 3 and 4, where the peak-height difference between type-A peaks and static peaks is a factor 1.25 to 2. This means that the type-A model of affective control of action selection can result in up to a 2-fold decrease of search investment needed to find a new goal. As expected, the smallest difference between control and type-A is when β is small (3 or 4) in the control condition (small β = much exploration = less tied to old goal). However, small β's have a classical downside: less convergence (Figure 3.6). The agent is less able to exploit its model of the world and thus does not learn the solution well, while type-A curves in Figure 3.6 show that the agent does converge to the minimum number of steps needed to get to the goal (i.e., 4 steps).

For completeness we show the β curves for the complete phase 1 of the control group agents, one type-A agent and one type-B agent (Figure 3.3). These curves confirm the expected β dynamics. For type-A, the goal switch induces high exploration (β near β_{min}) due to the lack of reinforcement ("it is going worse than expected"), after which β quickly moves up to β_{max}, and then decays to average. For type-B this behavior is exactly the opposite.

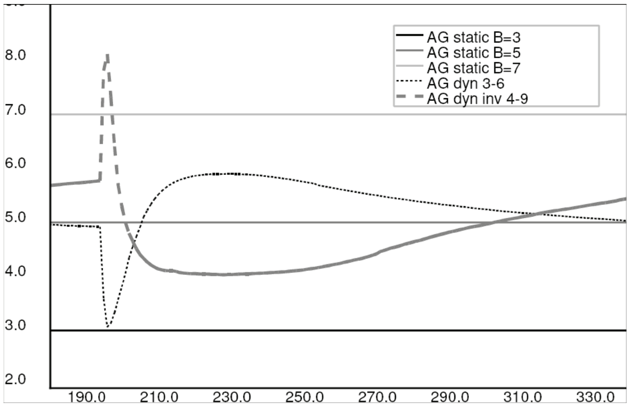

Figure 3.3. Alternating-Goal task; plot of the mean Boltzmann *β* for phase 1 (*n*=800). High *β* represents exploitation, low *β* represents exploration. The values of *β* for three static and two dynamic agents are shown. In all graphs, the trials are on *x*-axis, and means are based on the 5-95% percentile. Here, mean *β* is on the *y*-axis.

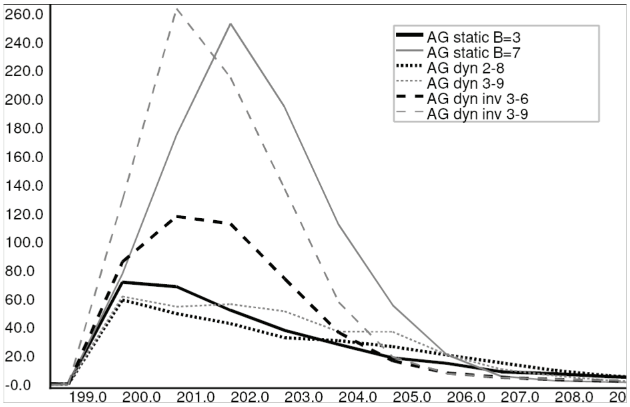

Figure 3.4. Alternating-Goal task; mean learning curves for phase 1 peak (*n*=800). The mean number of steps (*y*-axis) needed to find the goal is plotted per trial for two static, two dynamic and two inverse-dynamic agents (see text for explanation).

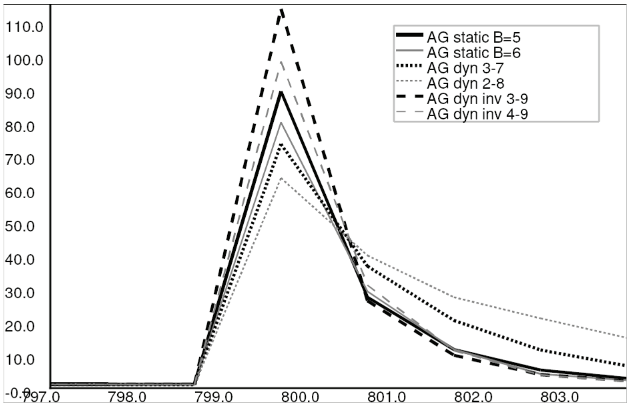

Figure 3.5. Alternating-Goal task; mean learning curves for phase 4 peak (*n*=800). The mean number of steps (*y*-axis) needed to find the goal is plotted per trial for two static, two dynamic and two inverse-dynamic agents (see text for explanation).

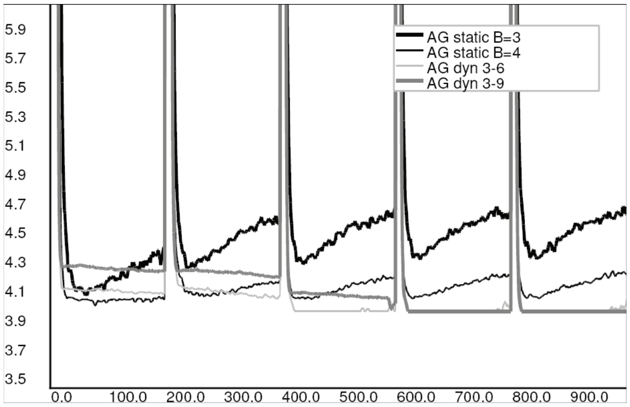

Figure 3.6. Alternating-Goal task; convergence plots of all learning phases (*n*=800), phases start at 0, 200, etc. 800. The mean number of steps (*y*-axis) needed to find the goal is plotted per trial for two static, two dynamic and two inverse-dynamic agents (see text for explanation).

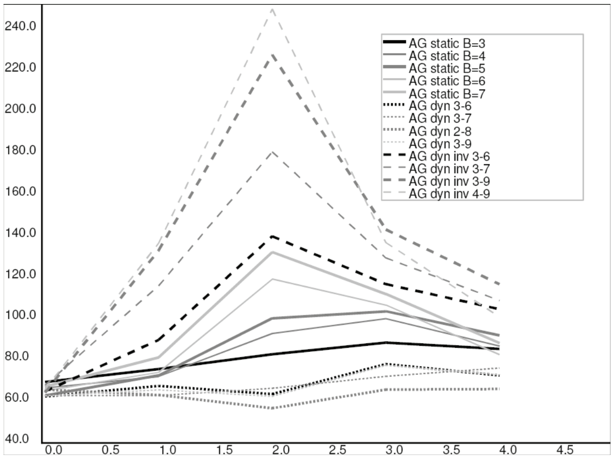

Figure 3.7. Alternating-Goal task; mean peaks of phases 0 to 4 (steps needed *at* respectively trail 0, 200, 400, 600 end 800) (*n*=800). Phase is on *x*-axis (only the integers); mean number of steps is on *y*-axis. The graph shows an overview for all agents of the mean number of steps needed to find the goal *at the goal switch*.

3.3.2 Experiment 2: Candy Task

Type-A agents have a considerable adaptation benefit compared to both control and type-B agents as shown by the following. In general, type-A agents have the same speed of finding the candy as exploiting agents (agents with a high static *β*), as shown by the learning curves of the complete task (Figure 3.8) and by the detailed learning curves of the start of the Candy task (Figure 3.10). In both figures the learning curves of *β*=6, and *β*=10 and dyn 2-8 overlap considerably. Interestingly, the quality of life curves show that in the beginning the QOL of the type-A agent quickly converges to the local optimum (candy, 0.25) comparable to that of the high *β* control agent (Figure 3.11, left “knee”). At the end of the task (later trials) the QOL of the type-A agent steadily increases towards the global optimum (food, +1.0; Figure 3.9). This shows that type-A affective feedback helps to first exploit a local optimum, while at a later stage explore for and exploit a global optimum. This is a major adaptation benefit resulting from type-A affective control of exploration. A playful way to think about this, is that the agent “gets bored” with the local optimum and as a result starts to search for other things, thereby increasing the chance of finding the global optimum.

The control agent with $\beta = 4$ does converge to the global optimum just like the type-A agent (Figure 3.9). However, due to continuous high randomness in this agents action-selection mechanism this agent consistently needs more steps to get to that global optimum as compared to the type-A agent (Figure 3.8). Also due to this high randomness this agent does not learn the local optimum consistently enough to quickly exploit it (Figure 3.11). High static exploration (smaller βs) results in a major delay in arriving at the same level of QOL as compared to the larger βs and the type-A agent (compare “candy static 4” curve with “candy dyn 2-8” curve in Figure 3.11). The type-B agent does not perform well at converging or at quickly exploiting the local optimum (Figure 3.8, 3.9, 3.10, 3.11).

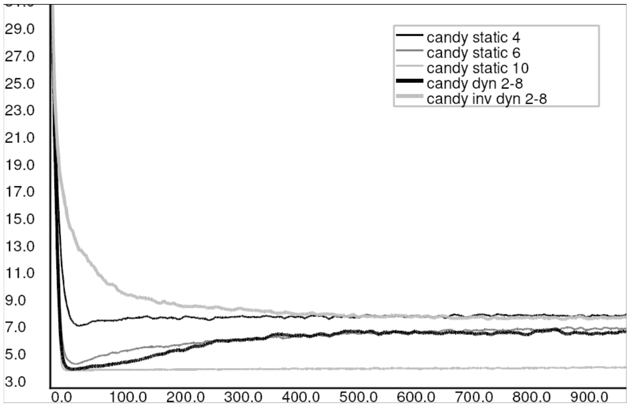

Figure 3.8. Candy task complete, mean learning curves (*n*=400). The mean number of steps (*y*-axis) needed to find the goal is plotted per trial for three static agents, one dynamic and one inverse-dynamic agent (see text for explanation).

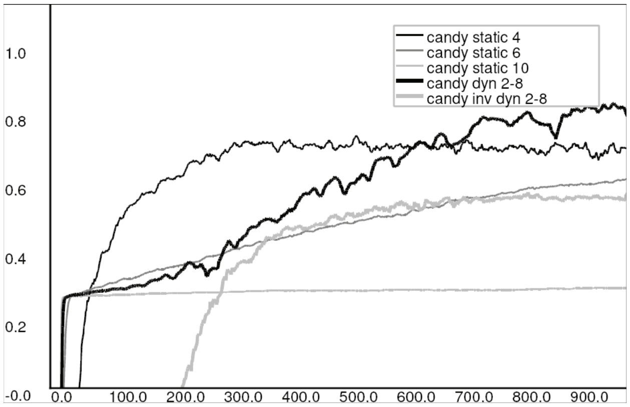

Figure 3.9. Candy task complete, mean Quality of Life curves (*n*=400). The mean QOL (*y*-axis) as it varies per trial is plotted for three static agents, one dynamic and one inverse-dynamic agent (see text for explanation).

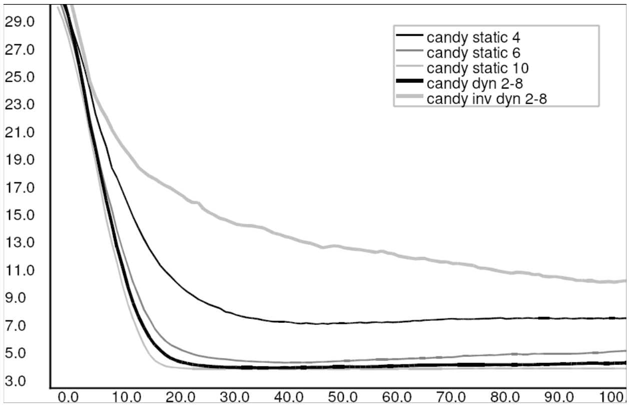

Figure 3.10. Candy task starts learning, mean learning curves (*n*=400). The mean number of steps (*y*-axis) needed to find the goal is plotted per trial for three static agents, one dynamic and one inverse-dynamic agent (see text for explanation).

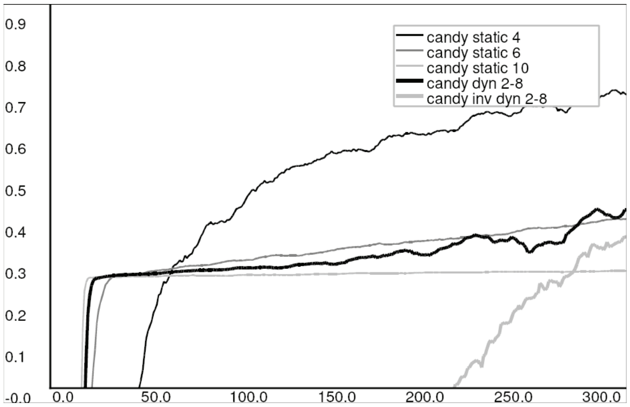

Figure 3.11. Candy task starts learning, mean Quality of Life curves (*n*=400). The mean QOL (*y*-axis) as it varies per trial is plotted for three static agents, one dynamic and one inverse-dynamic agent (see text for explanation).

3.4 General Discussion

Our results show that coupling positive affect to exploitation and negative affect to exploration provides two important benefits to learning and adaptation in the particular case of the grid worlds we have tested. Agents that use affect to control exploration in this way show significantly reduced task-switch cost *and* exploit a local optimum while being able to search for a global one.

3.4.1 Results Related to Other Learning Parameters

First, we briefly discuss the relation between learning and the proportion of local versus global optimum in the Candy task. If local and global optima are very similar, even a type-A agent cannot learn to prefer a global optimum, as the difference becomes very small. So, the candy and food reward have to be significantly different, such that the average β can exploit this difference once both options have been found. This has been confirmed in preliminary experiments we conducted, and is quite plausible: “you don’t walk a long way for a little gain.”

Second, we discuss the relation between discount factor and learning. This relates to the previous; a small γ results in discarding rewards in the future and therefore the agent is more prone to fall for the nearer local optimum. So, γ should be set such that the agent is at least theoretically able to prefer a larger later reward for a smaller earlier one, which is also the reason why we incremented γ to 0.8 in the Candy task, as compared to 0.7 in the Alternating-Goal task.

3.4.2 Results Related to Psychological Findings

Our results illuminate several psychological findings. First and foremost, they show that to understand the relation between affect and learning, the *process* of affective influence on learning is important. Only by coupling affect to exploration versus exploitation were we able to show that both positive and negative affect are useful for learning, but at different phases in the learning process. Negative affect induces exploration in those phases that need it, while positive affect induces exploitation of the learned model when needed. This is an important result providing empirical evidence (albeit simulated) for the idea that both negative and positive affect can relate to faster learning (Craig et al., 2004). It also provides evidence for the claim that some aspects of negative emotions are useful mechanisms for adaptation (Hecker & Meiser, 2005). More specifically, negative affect can defocus attention and thereby favor less selective decision making (Hecker & Meiser, 2005) (in our study modeled as a more random choice of action). Our results show that the dynamic coupling of affect and decision-making can increase adaptive potential of an agent if (1) negative affect relates to less selective decision making and (2) positive affect relates to more selective decision making.

Our results seem incompatible with the results by Dreisbach and Goschke (2004). They (and others) find that positive affect is related to more flexible, more distractible behaviors. In short, they argue that positive affect decreases selectivity (by increasing flexibility and distractibility) while negative affect increases selectivity (by decreasing flexibility and decreasing distractibility). However, closer investigation of their empirical results allows for a plausible alternative interpretation that relates to normal conditioning (Reinforcement Learning) effects. We discuss this is in detail, as our alternative explanation potentially is relevant to many affect induction tasks that measure reaction time and that allow for subjects to get accustomed to the task while it is being performed.

Dreisbach and Goschke (2004) measure the difference between reaction time (RT) before a task switch and after a task switch. This difference is interpreted as switch cost. So, if a task takes 600 ms at trials before a change in the characteristics of a task and 700 ms after that change, than this difference (100 ms) is the switch cost. The experimental setup is as follows. During a set of trials, subjects have to perform a simple cognitive task proposed in the target color (e.g., red). At the same time they see a different instance of the same cognitive task in the distracter color (e.g., blue). The subject's task is to react only to the task in the target color. Half way, there is a task switch. Now, two situations are possible, perseverance and learned irrelevance. In the perseverance condition, the target task is presented in a new color (e.g., yellow) and the distracter task is presented in the old target color. The subject's challenge is to *not* continue solving the task in the old target color. In the learned irrelevance condition the target is presented in the old distracter color (blue), while the distracter task is presented in a new color (yellow). The challenge here is not to be hindered by the novel color yellow or be inhibited by the old distracter color blue that has become the target color.

The main thrust for Dreisbach and Goschke's conclusion that positive affect reduces perseveration (= continuation on an old goal) but increases flexibility (= potential to switch to a new cognitive set) is (1) the relative lack of switch cost in the perseveration condition and (2) the increase of switch cost in the learned irrelevance condition. They argue that this is a specific effect of affect on perseveration versus flexibility. We will now present an alternative explanation based on standard learning and conditioning effects.

Affect can be interpreted as an unattributed reinforcement signal. First, it is generally accepted that floating (objectless) positive and negative affect is a signal to the organism defining the general goodness versus badness of the situation (e.g., Gasper & Clore, 2002). Second, we have argued and shown experimentally that reinforcement and affect are strongly related. Third, affect is coupled to the dopamine system (Ashby et al., 1999)—a system that is also highly related to Reinforcement Learning, a point explicitly made by, and one that underlies Dreisbach and Goschke's (2004) approach.

Therefore affect induction can alternatively be understood as unconscious reinforcement of trials. So in, e.g., the study by Dreisbach & Goschke (2004) positive affect induction can be seen as conditioning upon a certain task, specifically as the trials are repeated many times before the task switch is introduced. This means that subjects actually learn differently when affectively induced as compared to control or non-affective situations. This is an important point underlying our alterative interpretation.

Consider the following. When positive affect is induced, the subject is actually reinforced to respond to the task presented in the target color red and *not* to respond to the task presented in the distracter color blue. After the switch to the perseveration condition, the new color yellow is introduced (and the subject is explicitly made aware of this change). Now, there are two tasks. A new, neutral—non-reinforced—colored task and an old positively-reinforced colored task.

Consider the switch to the learned irrelevance condition. Again the subject is first reinforced on the target color red, and the task switch introduces the new color yellow. However, the distracter is presented in yellow, while the new target is presented in the old distracter color blue. This means that in the first condition the subject learns to react to a new stimulus (yellow), while in the second it has to perform reversal learning (blue meant no action, but now it means action). Reversal learning is generally considered more difficult than learning new behavior. According to this explanation, in the perseveration condition one would expect slightly better learning of the post-switch condition due to the generic effect of positive reinforcement during learning. In the learned irrelevance condition one would expect a much worse learning of the post-switch condition due to unlearning (reversal learning). This is almost exactly what has been found, *if the results are combined with a generic negative influence of positive affect on RT*. First, all positive affect situations have slightly higher RTs than the control (and pre-) tests, reflecting a negative influence of positive affect on performance on this specific task. Second, the perseveration condition has lower post-switch cost in the positive affect situation compared to the control (and pre-) test, reflecting enhanced learning due to positive affect. Third, the learned irrelevance condition has a major increase in switch cost as compared to the control (and pre-) tests, reflecting difficulty unlearning the previous association between distracter color and irrelevance.

This alternative explanation is plausible, albeit speculative. The main message of this elaborate discussion is that many affect induction studies could be measuring confounded dependent variables. The measured total effect can be a combination of both a learning-related effect (conditioning) and a top-down executive control effect that is not specifically related to learning (working memory, etc.). This is particularly important as these studies are done to measure the second effect. If part of the total effect attributed to top-down influences is in fact due to bottom-up influences, it is highly important to control for the bottom-up effect. The results of our—quite unusual—bottom-up approach to model a phenomenon that is typically considered top-down, shows that reasonably simple, and arguably low-level effects *can* be responsible for part of the flexibility effect. An additional experimental problem arises when attempting to separate these two effects, as both affect and reward seem to be mediated by the same dopamine system (Ashby et al., 1999). To summarize, our results cannot, at least not without further study, be considered as contrasting to results such as the ones discussed.

Current discussion on the Iowa Gambling Task (IGT) highly relates to our alternative explanation for the Dreisbach and Goschke study given here. The IGT (Bechara et al., 1997) measures the extent to which subjects learn to prefer to select cards from good decks versus bad decks. Good decks have many cards with low immediate monetary gain and some cards with low monetary loss. Bad decks have many cards with high immediate monetary gain but some cards with even higher loss. Overall, selecting cards from bad decks results in an average loss, while selecting cards from good decks results in an average gain. Subjects are unaware of the difference between decks and are asked to maximize gain by selecting cards from 4 decks (2 good, 2 bad).

In a sense, the IGT measures task-switching behavior. Up until the first bad card is selected from a bad card deck, these decks appear good, as they propose higher immediate monetary rewards than the good decks. After having selected the first bad card, subjects should re-evaluate (either consciously or unconsciously) their selection bias, ideally resulting in card-selection behavior directed at good decks. As subjects do not have any knowledge of the decks, we can easily interpret selecting the first bad card as a rule change that changes the current task. Prefrontal patients have difficulty learning to select cards from good decks instead of bad decks (Bechara et al., 1997). Alternatively, one could say that these patients are unable to switch to the new task of selecting from a good deck after having been reinforced to select from a bad deck. This interpretation suggests that prefrontal patients have difficulty switching to a new task in a Reinforcement Learning setting, which is quite plausible as the prefrontal cortex is often associated with executive control. Such control is needed for exactly this kind of task switches. This task-switch deficit might result from a lacking somatic marker signal (Damasio, 1994). However, in a recent review (Dunn, Dalgleish & Lawrence, 2006) it is argued that a reversal learning deficit can provide an alternative explanation. In a broad sense, this indicates that reversal learning is an important phenomenon to consider in all experiments that use (1) a learning task with potential involvement of reinforcement or affect, and (2) a, to the subject unknown, task switch due to a rule change. In a narrow sense, reversal learning is important in affect-induction cognitive-set switching experiments.

A comparison between the IGT and the Candy task is in place. The IGT has 4 decks of which 2 are good and 2 are bad. Every deck has a distribution of gain and loss cards. In terms of Reinforcement Learning one could say that a subject needs exploration to build a model of the average gain of the decks and subsequently needs exploitation to continue selecting cards from the good decks. Three main issues are thus involved in learning the IGT: (1) build a model of the goodness of a deck, (2) vary between decks such that all decks are covered, and (3) exploit the knowledge gained.

The Candy task is different (and simpler). There are no changing rewards. There is a local and a global maximum. The agent has to learn, through exploration, that a global maximum exists, and then exploit this maximum. Exploration-exploitation is controlled by affect in our studies. Key difference between the Candy task and the IGT thus is that the rewards in the Candy task are deterministic, i.e., once the agent has found the reward, it knows that this is the correct reward for that location in the maze. In the Candy task only the second and third issues are important (exploration-exploitation). Since the varying rewards in the IGT are a key characteristic of that task, our Candy task cannot be considered analogous to the IGT. Future work includes measuring the behavior of agents that use affect to control exploration-exploitation in a simulated IGT.

Of course alternative explanations for our experimental results are possible. Our model for affect could, for example, be interpreted as a model for *flow* (Csikszentmihalyi, 1990). If reward is consistently better than expected, we are in a state of flow and therefore continue to do what we do (model-based decisions). If reward is consistently worse than expected, we are out of flow, and engage in more random, search-like behavior.

However, our model of affect does seem to have face-validity, specifically in the context of adaptation. If things go well, don’t change. If things go bad, explore alternatives. This kind of underlying principle is quite plausible, but in stark contrast to the following: if positive affect indicates goodness, *we can afford to explore*, and if negative affect indicates badness, *we should be very selective regarding our behavior* (Dreisbach & Goschke, 2004). Which relation between affect and adaptation is right? Probably both, and the question is when and in what tasks? Only more elaborate process-oriented experimental and simulation studies will be able to show.

3.4.3 Results Related to Exploration and Exploitation in Machine Learning

The merit of using artificial affect as controller for exploration versus exploitation behavior has to be seen in light of adaptive agents in potentially changing environments. Such agents ideally decide autonomously when to explore versus exploit. It is in this context that we propose affect as signal to guide the learning process.

In contrast, standard methods exist that are far better at optimizing a solution to an arbitrary and *static* credit assignment problem. These methods stem from, e.g., operations research. Consider, for example, learning the optimal solution to the Candy task. This is merely a question of exploring enough in the beginning, and then gradually decreasing the amount of exploration; a process called *simulated annealing*. Given enough exploration and a smooth transition from exploration to exploitation, any RL mechanism is able to learn the optimal solution.

For an adaptive agent in a changing world, a gradual decrease in exploration is not what is needed mainly for two reasons. First, consider an autonomous robot that has to decide where to go. If that robot is purely exploring, it might choose actions that are lethal to it. In a simulated environment this is no problem, however, in a real environment this is. Second, consider a changing environment. In this case the problem is not static, and credit assignment can thus never reflect *the* optimal solution; it always reflects the *current* optimal solution. If a change occurs, the agent has to solve two problems that do not need to be solved for static problems. These are (a) how to detect the change, and (b) how to move back to exploration (in contrast to gradually moving from exploration to exploitation).

The problem we address with affect as meta-learning signal is not that of finding an optimal solution given an arbitrary problem. It is the problem of guiding the learning process such that the agent can autonomously decide *when* and *how* to explore versus exploit.

3.5 Conclusion

We have introduced a computational method of studying the relation between affect and probabilistic learning. Based on experimental results with learning agents in simulated grid worlds, we conclude that, at least in the task we have experimented with, coupling positive affect to exploitation and negative affect to exploration has two important adaptation-related benefits: 1) It significantly reduces the agent's "goal-switch search peak" when the agent learns to adapt to a new goal. The agent finds this new goal faster. 2) Artificial affect facilitates convergence to a global instead of a local optimum, while permitting to exploit that local optimum. Our results illuminate the process underlying the relation between affect and learning, and, we argue, is thereby a valuable addition to the existing affect-cognition literature. The results provide evidence for the idea that negative affect is related to less selective decisions while positive affect is related to more selective decisions. Further, our reinforcement-learning based analysis showed a potential problem with affect-induction techniques: the measured total effect of positive affect can be a combination of both a learning-related effect (conditioning) and a top-down executive control effect that is not specifically related to learning (working memory, etc.). However, as we have experimented with (only) two different types of worlds, our conclusions can not be generalized. More research is needed.

From a machine learning perspective, we have shown that in some cases artificial affect can be useful to guide exploration versus exploitation. However, more experiments should be done, specifically in different, and larger, worlds, using other RL models (for example, models that are able to cope with continuous environments).

4

Affect and Thought

Affect-Controlled Simulation Selection

In this chapter we study affective control of the amount of simulated anticipatory behavior in artificial adaptive agents. Artificial affect is positive when an agent is doing better than expected and negative when doing worse than expected, as defined in Chapter 2 and used in the study in Chapter 3. Our approach is based on model-based Reinforcement Learning (although we use a different model than the one used in Chapter 3) and inspired by the *Simulation Hypothesis* (Cotterill, 2001; Hesslow, 2002). In contrast to the research described in Chapter 3, where we used affect to control the exploration – exploitation rate directly, in an adaptive agent that has a purely reactive architecture (no internal simulation of interaction), 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 selects anticipatory behaviors for simulation from the agent's Reinforcement Learning model.

Based on experiments with adaptive agents in two nondeterministic partially observable grid worlds we conclude that (1) internal simulation has an adaptive benefit and (2) affective control reduces 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. 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.

4.1 Introduction

In this Chapter we study affective control of the amount of information processing in artificial adaptive agents. In order to model affective control of information processing, we use the measure for artificial affect, as defined in Chapter 2, which relates to an adaptive agent's relative performance on a learning task. Artificial affect measures how well the agent improves. Our adaptive agent learns by reward and punishment. Thus we define "wellness" based on averages over reinforcement signals. As such, 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 measure of artificial affect thus relates to natural affect in the sense that it characterizes the situation of the agent on a scale from good to bad. Further, as our measure is based on average reinforcement signals, it relates more to mood than emotion.

We have developed a variation to the model-based Reinforcement Learning (RL) paradigm (Sutton & Barto, 1998). This variation enables us to view 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 that evolutionary continuity is a critical aspect in studying behavior, emotions, consciousness and cognition. In this chapter, we refer to simulation as described by the Simulation Hypothesis.

An important current 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.

In this chapter 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.* Thus, the main contributions of this chapter to the affect-learning and Simulation Hypothesis literature are:

- The introduction of an affect-controlled mechanism for the selection of internally simulated behavior instead of actual behavior; we define this mechanism as *simulation selection*.
- A study into the influence of affect on learning, when 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 trade-off between internal versus external information processing effort (Aylett, 2006).

In Section 4.2 we review the relation between internal simulation and our approach in more detail. In Section 4.3 we present our computational model and how it implements artificial affect, internal simulation of behavior and learning. In Section 4.4 we describe our experimental setup. In Section 4.5 we present experimental results. In Section 4.6 we discuss our approach in a broader context.

4.2 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 internally simulate actions, perceptions and evaluations of action-perceptions in an off-line manner. 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 simulation process.

4.2.1 Thought and Internal Simulation of Interaction

In addition to being plausible, 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 on 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 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 crucial finding 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).

4.2.2 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 (exemplified by a computational neuronal model by Dehaene, Sergent and Changeux (2003)).

The "internal simulation thought process" would go like this. 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 (pushed away) 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 simulation-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. Moreover, the simulation-selection threshold in our model is dynamically controlled by artificial affect (Section 4.3.2, 4.3.3).

4.3 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 grid worlds. Figure 4.1 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). In the general discussion we explore some of the similarities and differences.

We first discuss the components of the model and the way it learns using RL principles. Then we explain how we have implemented the Simulation Hypothesis on top of our model. Subsequently we explain how artificial affect 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.

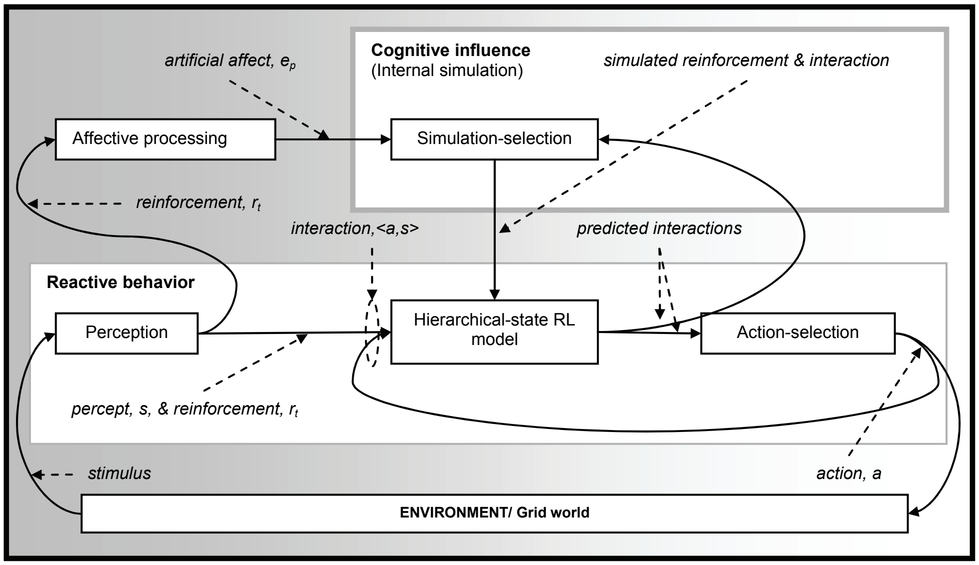

Figure 4.1. Overview of the different components in our model. Components are detailed below.

4.3.1 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. 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}>…<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 type 2 nodes, and the connectivity between type 1 and type 2 nodes 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 thus 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* $<a, s>$. Consider, for example, an agent that has chosen some action $\tilde{a}$ and experiences some state σ. Because its model does not yet contain a node that represents $<\tilde{a}, \sigma>$ it is created (e.g., s_1 in Figure 4.2a). 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 $s_2=<\tilde{a}', \sigma'>$, giving a new node that represents s_2 (Figure 4.2b). 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 a new type 2 node, defined as an *interactron* (sic!), connected to s_1 and s_2 with edges as shown in Figure 4.2c. This node 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. A type 2 node $(h_l, <a', s'>)$ 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 4.2c are active. An additional example is presented in Figure 4.2d and 4.2e. If situation s_2 is followed by a new situation s_3, the resulting memory structure is shown in Figure 4.2d, 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 4.2e, 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 type 2 nodes is k (for computational reasons k = 10 in this study; for more on k see below and Broekens & DeGroot, 2004b).

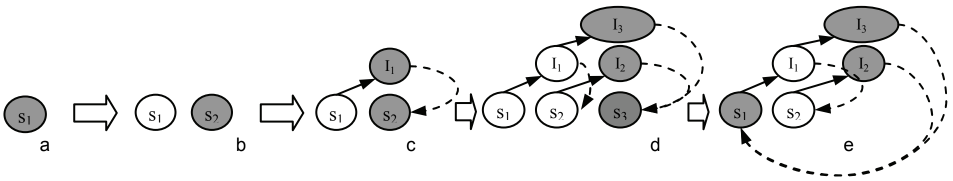

Figure 4.2a-e. Examples of the agent's memory structure

Every node $(h_l, <a', s'>)$ 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 $<a', s'>$. If at a later time the sequence of situations h_ls_i is *again* observed by the agent, then the statistic υ of the type 2 node 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:

$$p(x \mid y) = \upsilon_x \Big/ \sum_{i=1}^{|X_y|} \upsilon_{x_i} \ , \qquad (4.1)$$

where y is a node 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 node (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*> after a history of action-state pairs h_l using the most recent model of the world.

Furthermore, we define a global threshold called the *forgetting rate*, *θ*, representing the minimal "survival probability" for an interactron. If $p(x \mid y) < \theta$, the corresponding interactron *x* is forgotten and removed from the 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. The relation between interaction (de)stability and our learning model is explained in more detail in (Broekens & DeGroot, 2004b). 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:

$$v = r + \gamma v_{next} \text{, with } v \text{ maxed-out such that } \min(r, v_{next}) \le v \le \max(r, v_{next}) \quad (4.2)$$

where *r* is the learned reward for a certain interactron, *γ* the discount factor (equal to 1.0 in all our experiments, see below for why this does not pose a problem in our approach) 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*, its learned reward *r*(*y*) is adapted according to the formula:

$$r(y)^{t+1} = r(y)^t + \alpha(r_t - r(y)^t)\,, \qquad (4.3a)$$

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

$$v_{next}(y)^{t+1} = \sum_{i=1}^{|X_y|} v(x_i \mid y)^t \times p(x_i \mid y)^t \text{ ,} \quad (4.3b)$$

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.

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 kth interactron predicts potential next action-state pairs <*a*', *s*'> using a history of length 1 until k respectively (e.g., I_3 is a k=2 interactron with history s_1s_2). As such, this memory learns 1st...kth 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, with some of them predicting the future based on little history and some predicting the future based on a history of length at most k. The predictions consist of estimated future values for next action-state pairs, as usual. However, k of these MDPs are active at the same time, so action selection integrates not over the predictions of 1 such MDP but over the predictions of k such MDPs. How action selection integrates over these parallel predictions is explained in the section on action selection below. 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, partially 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 $s_0s_1h_{l-2}s_{t-1}$. Other limitations, experimental convergence results as well as several choices for the world model itself are discussed in more detail in Broekens & DeGroot (2004b).

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. In the ideal (policy unbiased) case, the value of every such node converges as is usual for *Q*-values in RL.

4.3.2 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} to converge *v* introduces a problem of cumulative prediction errors).

4. *Action selection*: select the next action using the mechanism explained in Section 4.3.4. 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 chapter, 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, and action selection could be severely compromised. 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*):

- 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).
- 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 4.3.3.

In essence, simulation selection is controlled by a simulation-selection threshold*,* t_s*,* of a t_s-Winner-Take-All (WTA) simulation selection ranging from infinite (no simulation) to zero (select and simulate all predicted action-state pairs). This threshold 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 *predicted best next interactions* that should be internally simulated (so in a sense it is an inverse threshold). If $t_s < 0$ (overly selective threshold), no simulation is done. If $t_s \geq 0$ (selective threshold) only the interaction with the highest predicted value is simulated, if $t_s \approx 1.0$ (non-selective threshold) all interactions are simulated. 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).

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

Using the measure for artificial affect, e_p, introduced in Chapter 2, it has now become straightforward to model affective control of the amount of internal simulation (i.e., affective control of working memory content), the basis of our study. Control can be modeled in several, equally plausible, 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. This reflects the hypothesis that positive affect decreases the amount of internal simulation favoring narrow, exploitative thoughts (i.e., only action-state pairs with a high value are internally simulated), while negative affect increases the amount of simulation favoring broad thoughts, including explorative ones (i.e., action-state pairs with low values are also simulated). This relates to results found by Rose et al. (1999). In our model this means that happy agents (i.e., performing better than expected) simulate positive thoughts, while a discontent agent simulates many thoughts including negative ones. So:

$$t_s = 1 - e_p \qquad (4.5)$$

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:

$$t_s = e_p \qquad (4.6)$$

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 ($e_p \approx 0$) or positive affect ($e_p \approx 1$) while not intense is neutral ($e_p \approx 0.5$). 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 thesis. The simulation-selection threshold is:

$$t_s = 2 \times abs(0.5 - e_p) \qquad (4.7)$$

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

$$t_s = 1 - 2 \times abs(0.5 - e_p) \qquad (4.8)$$

In Section 4.5 we report on the results of a systematic study that investigated the influence of internal simulation on the adaptiveness of artifical agents, when the amount of simulation is modulated by affect. Modulation is according to the hypotheses mentioned above.

4.3.4 Integrating Everything: Greedy Action Selection over Biased Value Predictions

In our approach, action selection must integrate over the predictions of at most k MDPs in parallel: 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” (see, 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:

$$l(a)^t = \sum_{i=1}^{k} \sum_{j=1}^{|X_{y_i}|} v(x^i{}_j \mid y_i)^t \times p(x^i{}_j \mid y_i)^t \ , \qquad (4.9)$$

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, $x^i{}_j = (h, <a, s>)$ with $h = h(y_i)s_{y_i}$ and $(h(y_i), s_{y_i}) = y_i$ and $s_{y_i} =< a^t, s^t >$. 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:

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

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 Winner-Take-All action-selection. As such, action selection uses a super-threshold greedy selection with sub-threshold linear weighted stochastic selection.

Further, 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 wrap up this section on the computational model consider the following. The number of thoughts that occupy working memory is often interpreted as an indicator of the intensity of information processing. As a thought equals an internally simulated behavior in our model, and 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 of information processing.

4.4 Method

To investigate the influence of affect-controlled anticipatory simulation of future action-state pairs, we have set up a grid-world environment consisting of walls, roadblocks, cues, food and empty spaces. We use two non-deterministic (i.e., changing), partially-observable grid worlds. Common to our two grid worlds 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 defined in the model (Section 4.3).

The first grid world 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 4.3). 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 in the following way. 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. As a result, both the world and the reward structure of that world change. The agent is, of course, 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 grid world the *switch-to-invest* grid world, 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.

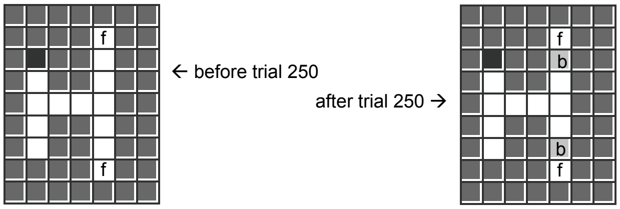

Figure 4.3. Switch-to-invest task. Potential start locations are alternated between the top-left and bottom-left arms, goal locations are alternated between the top-right and bottom-right arms.

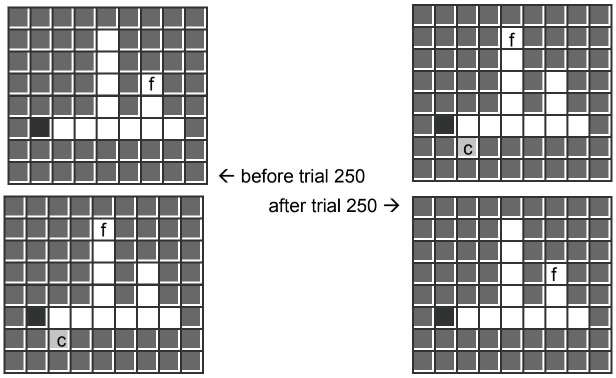

Figure 4.4. Cue-inversion world. The left and right pictures show the possible worlds before and after the cue inversion at trial 250 respectively; ’f’ is food, ‘c’ is cue, black square is the agent.

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 grid world, 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 4.4) 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 (e.g., when the agent has misinterpreted the cue). The non-goal location (empty arm) has a negative reinforcement of r=–0.5.

To test our three hypotheses, we vary the simulation-selection mechanism and analyze how an artificial agent copes with these two worlds. Our agent employs the learning and simulation mechanisms as described in Section 4.3. 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 4.3.3. 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. As mentioned earlier, our measure of affect has three parameters that define its behavior. We varied these three parameters, 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 grid-world experiments we varied these according to Table 1, resulting in 30 different affect-parameter settings. In our cue-inversion grid-world experiments we varied these only according to the f=1 column in Table 1, resulting in 10 different affect-parameter settings.

Further, 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 forgetting rate 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.

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 4.1. Possible *ltar*, *star*, and *f* combinations as they are used in the first set of experiments with the agent in the *switch-to-invest* task.

4.5 Experimental Results

We first describe the results obtained with the switch-to-invest grid world, after which we describe the results obtained with the cue-inversion grid world. Data was analyzed as follows. To investigate the effect of learning rate, α, forgetting rate, 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:

- 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 ($\alpha = 1$ and $\theta = 0$), 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., 20 steps. For a dynamic simulation mechanism the average is constructed in the same way using aggregated runs for every $<\alpha\ ,\theta>$ 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.
- 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 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.

4.5.1 Results of Experiment 1: Switch-to-invest Task

Results in this specific grid world show that simulation in general has a stable positive effect on learning. This trend is shown by the learning curves[1] in Figure

[1] 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.

4.5, and more formally in Figure 4.6 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 interactions, the better the learning result (*simall* costs less effort than *simbest*, *p*<0.05 for all settings except *α*=1 & *θ*∈{0, 0.01}, Figure 4.6). 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 *α*=1 & *θ*∈{0, 0.01}, Figure 4.6). 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 4.6). 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 4.7a, shown only for *α*=0.8 & *θ*=0.03). However, for all <*α* ,*θ*> 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 4.7b, shown only for *α*=0.8). Finally, results for α=0.9 are not shown, as these appeared to be an interpolation between the results for α=0.8 and α=1.0.

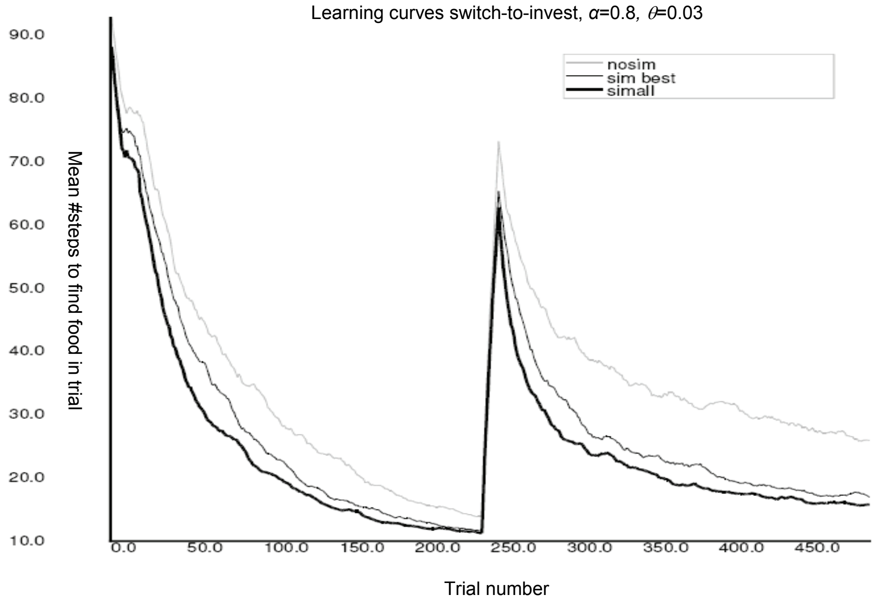

Figure 4.5. Learning curves (smoothed) of non-, best, and all-simulating agents in the switch-to-invest world for α=0.8, θ=0.03. Curves of other strategies are approximately in-between best and all.

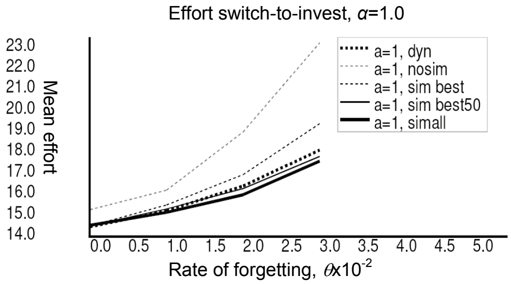

Figure 4.6a. Effort for different simulation strategies in the switch-to-invest task with a learning rate, α, equal to 1.0

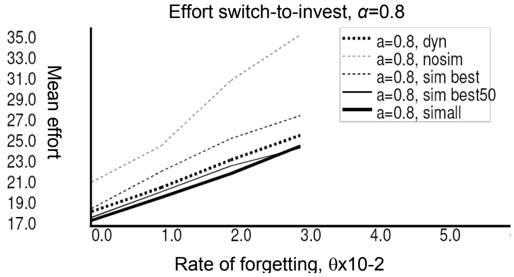

Figure 4.6b. Effort for different simulation strategies in the switch-to-invest task with a learning rate, α, equal to 0.8

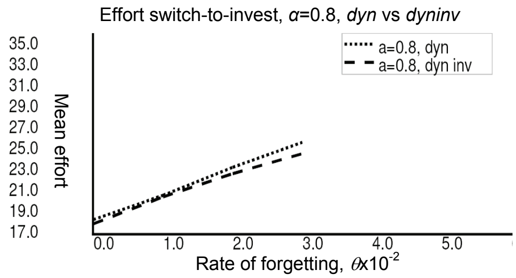

Figure 4.7a. Small difference in effort between dynamic and inverse-dyn simulation strategies.

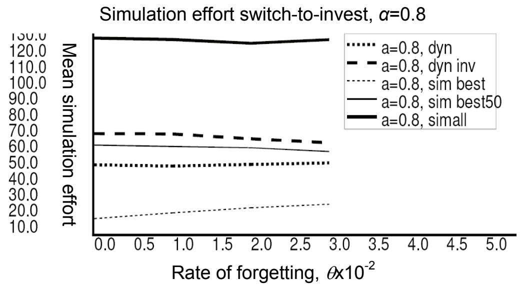

Figure 4.7b. Difference in simulation effort between simulation strategies.

4.5.2 Discussion of the Switch-to-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, forgetting rate 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 ($\alpha \approx 1.0$), 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 forgetting rate 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 4.7b) and seems to perform slightly better than the latter (Figure 4.7a). 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 grid world, i.e., the cue-inversion world.

4.5.3 Results of Experiment 2: Cue-inversion Task

Results in this grid world 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 the first experiment 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 $\theta=0$. This confirms our explanation of interaction effects between simulation, α and forgetting rate in the discussion of the previous experiment.

Again, dynamic strategies are quite close to the *simall* strategy in terms of learning performance (Figure 4.8a): 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 4.8b, 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 4.8a), 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 4.10). A small short term average (i.e., "how am I doing") results in a faster affective reaction to the cue-inversion (inset).

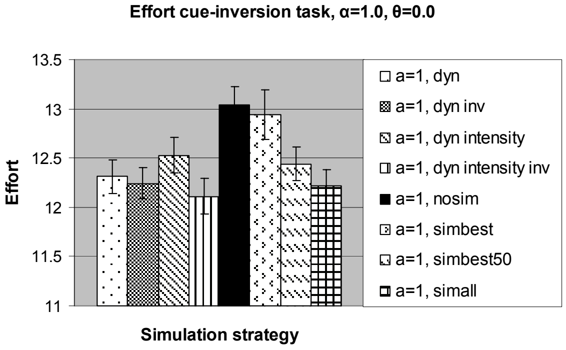

Figure 4.8a. Difference in effort between dynamic and static simulation strategies. Error bars show 95% confidence interval.

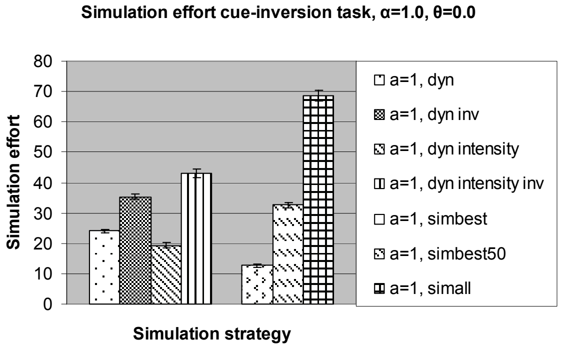

Figure 4.8b. Difference in simulation effort between static and dynamic strategies. Error bars show 95% confidence interval.

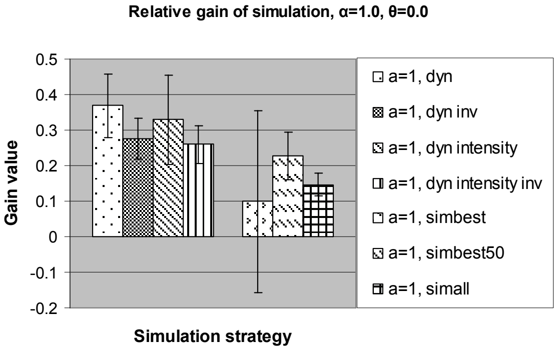

Figure 4.9. Gain of simulation strategies (details in text). Error bars show 95% confidence interval.

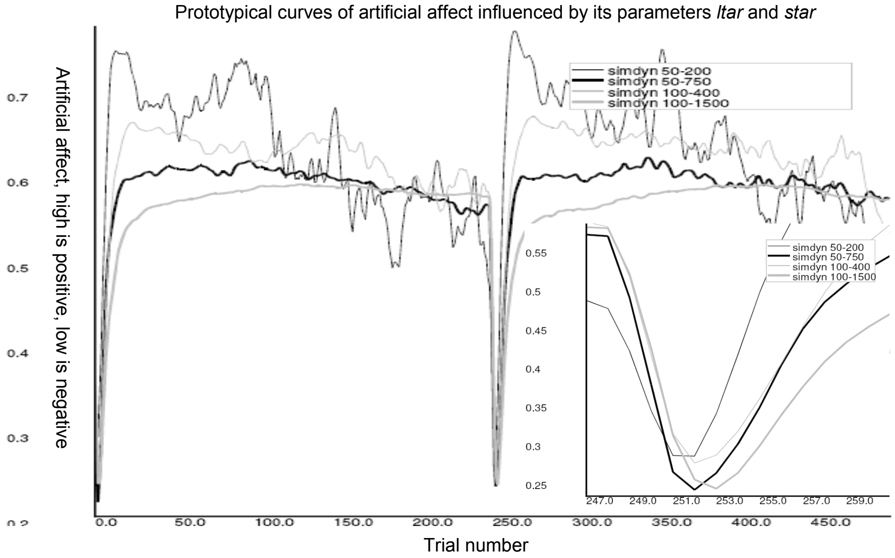

Figure 4.10. Depicted are affect curves for different settings (not smoothed). Inset is a detail of artificial affect at the cue inversion. Note that *star*=50 has the “dip” earlier than *star*=100.

4.5.4 Discussion of the Cue-inversion Task 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 control 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:

$$gain_i = \left(effort_{non} - effort_i\right)/\left(sim\_effort_i \,/\, effort_i\right) \qquad (4.11)$$

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 4.9). Simulating all is not very efficient compared to dynamic strategies. Interestingly, our original coupling of affect and amount of simulation seems most promising (as proposed, but not yet confirmed in 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*.

4.6 General Discussion

We now discuss our approach in a broader context. We first ground our approach more firmly, and relate our work to the work of others. Finally we present some directions for future research.

4.6.1 Model Grounding

Our findings are compatible with psychological findings that show that both positive and negative affect influence learning in a beneficial way (Craig et al., 2004; 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 neutral but not positive 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 Working Memory (WM) that depends on affect can be an adaptive strategy to cope with the changing world around us, without enforcing any capacity constraints.

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 simulation-selection threshold (cf. Section 4.3.3)—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 (see Step 1 – 5, Section 4.3.2).

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 therefore 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 results *only* from 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.

4.6.2 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 and 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), or explicitly used as input to the neural network controller to enable “’blindfolded’ corridor following behavior” based on these simulated next states (Ziemke). Although our action-state encoding and learning mechanism are different, our overall architectural approach is similar, especially to the work of van Dartel and colleagues. 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 influences action selection. The feedback from this copy to the input represents a simulated next state as predicted by the model (van 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 “blind-folded”. 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 a result, 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 broad simulation. This is the most important reason why simulating all potential next action-states is not really equivalent to exploration. 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 model 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 by biasing the predicted values of next states and thereby influencing action selection. In *Dyna*, planning can actually change the evaluation and policy functions.

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 of just one step ahead is beneficial to artificial adaptive agents, even if simulation does not alter the long-term knowledge of the agent. The influence of simulation is mediated by the action selection mechanism of the agent. Simulation introduces a temporary bias to the values predicted by the model. This approach is similar to the one proposed by Gadanho (2003). In her 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.

4.7 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 grid worlds we conclude that (1) anticipation 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 non-deterministic 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).

5

Affect and Modulation

Related and Future Work

In this chapter we discuss other approaches towards computational modeling of affect as well as directions for future work.

5.1 Related Work

The work described in the previous two chapters relates to emotion and motivation based control/action-selection. We explicitly define a role for emotion in biasing behavior-selection as do Avila-Garcia and Cañamero (2004), Cos-Aguilera and others (2005) and 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, either by biasing simulation selection (Chapter 4) or by biasing the greediness of action selection (Chapter 3).

A recent variation of this type of research has been presented by Blanchard and Cañamero (2006). In this study, artificial novelty and affect are coupled to exploration behavior of a robot that has to autonomously explore different possible distances to a box. Familiarity (non-novelty) modulated by positive affect is coupled to exploration. The authors argue that their study reproduces behaviors observed in nature. 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, which is also 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.

Two fairly different approaches—different from ours and different from each other—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 emotive episode (i.e., the short term onset and decay of an emotion) 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. There is no space here to do justice to this approach that is important to the process of emotion elicitation in adaptive agents in the spirit of, e.g., Rolls (2000). However, we do want to point out the main difference between Lahnstein's approach and ours, i.e., 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 (i.e., the form and elicitation of an emotive episode) with ours, such that our measure of long-term affect is based upon averages over the positive/negative aspect of Lahnteins short-term emotion.

Salichs and Malfaz (2006) introduce an interesting way in which affect can be embedded into the value function *Q* of a standard Reinforcement Learning method. They enhance *Q*-learning so that the reward is based on the happiness/sadness of the agent, where happiness and sadness are derived from the agent's wellbeing. Wellbeing is a function over the extent to which the agent's drives are met. So, the more drives met, the happier the agent. This means that their agent is intrinsically motivated by affect, and strives to "maximize happiness" (Salichs & Malfaz, 2006). They use fear, modeled as a parameter that dynamically modulates to what extent the agent chooses—in a world with a stochastic reward function—a risky but optimal policy versus a conservative policy. Fearless agents emphasize actions that are potentially good, while fearful agents more strongly consider the effect of actions that are potentially bad. Their approach thus differs from ours, but, again, both approaches could be integrated such that wellbeing based on drives provides the reward signal and thus our measure for artificial affect is based upon wellbeing averages.

Strongly related to our approach to affect-modulated exploration is research by McMahon et al. (2006), Morgado and Gaspar (2005) and Gadanho (1999; 2003). We discuss this work in more detail in the rest of this section.

McMahon et al. (2006) 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. This probability uses two measures: one derived from the accuracy of prediction for the upcoming reward as given by the learning mechanism that learns to predict values for future states; the other derived from the actual rewards received. As a result, the probability to explore is high when rewards are low and errors are made in the value prediction, while exploitation is high when rewards are high and prediction errors are low. In this manner they show that agents learn a grid-world problem faster when using this probability value to control exploration. 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 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).

In her PhD thesis, Gadanho (1999) shows an impressive collection of experiments that investigate the relation between affect and adaptive behavior. Here we will discuss several of them. First and foremost, she shows that affect and emotions can be embedded into adaptive agent architectures in a vast amount of ways. These include internally generated emotions as reinforcement to the agent, emotion as interrupting triggers that initiate alternative behaviors when needed, affect-based learning rates, and affect-based exploration/exploitation tradeoffs. Experimental results also vary from positive to negative (in terms of behavioral and learning efficiency). Here we will focus mainly on affect as meta-parameter setting, i.e., affect-based modulation of learning rate and affect-based control of exploration rate (Section 4.6 in Gadanho, 1999).

The experimental setting used is a grid world with an agent that has a neural network robot controller. Every action has a neural network that learns to predict the value of that action in a certain situation. The networks learn with a learning rate η. An action-selection module selects actions, using a temperate parameter T, based on the predictions of the neural networks. The grid world contains walls (avoid), lights (reinforced) and different starting locations (to vary where the robot starts). The goal of the agent is to optimize reward over time. The agent can have four potential emotions, *fear*, *happiness*, *sadness* and *anger*, based on the agents internal drives, *hunger*, *pain*, *body-temperature*, *restlessness* and *eating*. Emotions are elicited continuously during the behavior of the robot. For the current discussion it is not necessary to go into the details of the emotion elicitation model used.

In the emotion-modulates-learning-rate case, the intensity of the agent's dominant emotion is used as gain (multiplication) factor for the learning rate learning η. This gain is equal to 0 if there is no dominant emotion. Experimental results are difficult to interpret. There does not seem to be a generic learning benefit. This is obvious, as the agent can only learn when it experiences an emotion. Therefore learning will always be slower. However, the results have to be interpreted differently (Gadanho, 1999). One could say that the agent saves learning resources by only learning if its emotion indicates a significant situation. Indeed, in the long run, the agent does learn appropriate food finding behavior, so it is not hindered by slower learning. However, it might be that different tasks benefit more from affect-based learning rate modulation than the task used by Gadanho. Consider the following. In a task where the environment can suddenly change (unlike the task used by Gadanho), an increase in "fear" (e.g., due to running around and not eating) could trigger the start of an intensive learning period. Once the task has been learned again, fear would drop, and the agent would stop learning. While being "bored" (i.e., neutral emotion) the agent will not learn. This is good, as it therefore cannot unlearn previously learned behavior either. In this scenario, it is clear that affective modulation of the learning rate is also useful for adaptation in general, not just for saving learning resources. This thought experiment should, however, be investigated experimentally.

In the emotion-modulates-exploration-rate case, the intensity of the agent's dominant emotion is used to control the Boltzmann temperature. The best results in terms of learning performance are found when a negative dominant emotion is coupled with an increase in temperature. This means that when the agent is feeling bad, it starts to explore. This is exactly the same relation we have found in our studies, and as such there seems to be some convergence on the negative affect=exploration relation. Further, we have extended the studies by Gadanho in the following way. We have studied in more detail the exact behavior of this relation, including many alternative relations between affect and exploration. We have explicitly grounded the relations to the psychological affect and learning literature. Finally, we have explicitly developed several learning tasks (*candy task*, *switch task*) that enable to better investigate the potential of affective modulation.

Morgado and Gaspar (2005) take a slightly different approach. Theirs is more related to our work on affect-bounded thought (Chapter 4), instead of affect-based regulation of exploration in learning (Chapter 3; Gadanho, 1999; McMahon et al., 2006). They present a theoretical framework that explicitly defines a strong, dynamic relation between emotion and cognition. We focus on one aspect of their framework as this aspect relates strongly to our approach: affective bounding of cognitive effort. Affect is derived from how well the agent is doing, analogous to our approach. However, it is defined slightly different. Affect—*Emotional Disposition* as they call it—is defined as a point in a two dimensional space. The dimensions are *change in achievement potential* (δP, achievement potential is the degree to which an agent can change the current state of affairs in the direction of the goal) and *change in goal conduciveness* (δF, goal conduciveness is the degree of cooperation of the environment regarding goal achievement of the agent). This means that if an agent has a certain goal, and it is steadily moving towards it, affect will be slightly positive overall, as there are no changes in goal conduciveness but there is a constant positive change in achievement potential. Why? Because the agent gets closer and closer and therefore its potential to change the situation to achieve the goal gets larger and larger (assumed that certainty equates potential, hence being close to a goal means a high level of certainty about being able to get to your goal).

In their framework, behavior is understood as the reduction of the difference between the current situation (referred to as *observation*) and a goal situation (referred to as *motivator*). Both observation and motivator are defined as coordinates in a cognitive space, referred to as *cognitive elements*. The dimensions of this cognitive space are equal to the many different qualities a cognitive element can have. For example, dimensions could include “intensity of sunlight”, “outdoor temperature”, “sea and waves”, “having interesting books around” and “cocktail availability”. If the current situation is described by a cognitive element having low values on all of these dimensions, while a motivator cognitive element (a goal) exists that has high values on all of these dimensions, one needs a sun-and-beach vacation. The distance between the observation and motivator points thus indicates how much behavior is needed in order to get to a goal state.

Affect relates to cognition and behavior in the following way. Decrease of distance between the points defining current situation and goal state relates to an increase in achievement potential, and increase in speed with which the agent moves towards the goal state relates to increase in goal conduciveness. If I book my vacation and am on the road, my potential to achieve my goal increases, and there is a sudden increase of the goal-achieving speed (acceleration of my observation cognitive element in the cognitive space with respect to the motivator element). As such, positive affect relates to positive δP and δF, while negative affect relates to negative δP and δF.

Affect influences information processing by regulating attentional effort of the agent in two ways. Affective signals are integrated over time into a dynamic threshold ε. If a cognitive element has an activation value (let’s call this its saliency) that is higher than the threshold ε, it is considered in the thought process of the agent. Second, a similar threshold ω is used to control the amount of processing the agent has available before it needs to act. Affect thus controls what cognitive elements enter working memory, as well as how long these elements can stay there for active contemplation. It should be clear that there is a strong analogy with our approach. With regards to affect-based control of resources, the key differences are:

First, we have defined affect in terms of average reward signal changes, while Morgado and Gaspar (2005) exclusively define affect in terms of goal-orientation. This is rather limited, as it assumes a purely cognitive interpretation of affect elicitation and it needs a representation of a future goal. For their purpose this might be sufficient, but for modeling more down-to-earth effects on learning and adaptation (such as the influence of affect on exploration versus exploitation), defining affect in terms of goals is problematic. Representations of future goals might simply not be available.

Second, we specifically—and more elaborately—relate our measure of affect to psychological studies that relate to the influence of affect on learning and information processing. While it is clear that our measure of affect is simple (and in many aspect simplistic), it is strongly grounded in psychological findings. This cannot be said of affect as per Morgado and Gaspar (2005). For example, it is not clear in their model what it actually means if δP and δF do not have the same sign, nor is it very clear how emotions relate to the four possible quadrants of δP and δF.

Third, our approach focuses on the influence of affect-based control of the amount of processing (i.e., internal simulation) on the *learning effectiveness* of an adaptive agent, while Morgado and Gaspar (2005) focus on *problem solving effectiveness* by coupling their affect-based control mechanism to a standard planner. The planner was embedded into an agent that had to continuously plan routes to changing food location in order to maximize food intake. The main result of their experiment is that affect-based control can indeed make more efficient use of planning resources than non-affect-controlled planners.

Regardless of the differences, their results are promising and complementary to ours. They show that even when starting from a very different theoretical point of view, affect-based control of information processing can be a useful method to help resource-bounded agents adapt.

5.2 Future Work

The maximum total amount of simulation used in the setups in Chapter 4 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 exploration versus exploitation 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 chapter is to use different learning architectures, such as *Soar*, or ACT-R. Using the ACT-R architecture, Belavkin (2004) has shown that affect can be used to control the search through the solution space, which resulted in better problem-solving performance. Belavkin has an information-theoretic approach towards modeling affect that is related to the rule state of the ACT-R agent. A key difference is thus that our artificial affect is based on a comparison of reinforcement signal averages. Further we have explicitly modeled affect according to different theoretical views on the relation between affect and information processing and compared these different views experimentally. The “Salt” model by Botelho and Coelho (1998) relates to Belavkin’s approach in the sense that the agent's effort to search for a solution in its memory depends on, among other parameters, 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 the affect-based control techniques introduced in Chapter 3 and 4 in *Soar_RL* (Hogewoning et al., 2007).

Affective control should be investigated in other types of 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). Also, more complex and more realistic tasks should be used to test the affect-based mechanisms proposed in the previous two chapters.

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.

6

Affect as Reinforcement

Affective Expressions Facilitate Robot Learning

Until now, we have used affect as an abstraction for how well the agent is doing. This abstraction is a long-term signal, based on the reinforcement the agent receives during the process of learning. We have experimented with controlling meta-parameters by means of artificial affect (exploration versus exploitation, Chapter 3; broad versus narrow thoughts implemented via internal simulation of potential interaction with the grid world, Chapter 4). Artificial affect was thus related to mood (long timescale, not directed at a specific situation). Furthermore, artificial affect was a signal originating from the agent itself.

Affect can also be an abstraction for the positiveness versus negativeness of a *current* situation or object, as well as being elicited more directly by an external source (e.g., when used in affective communication). In this chapter we take such an approach. We thus part from the definition of affect introduced in Chapter 2. In this chapter, affect is a short-term signal communicated by a human observer to a learning simulated robot. The common part in this definition of affect and the one introduced in Chapter 2 is that affect still is an abstraction for positive versus negative.

In this chapter we briefly present *EARL*, our framework for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning. *EARL* is a framework, currently a prototype, that embodies many of the ways in which affect can influence learning, when learning is conceptualized as Reinforcement Learning (RL). *EARL* enables the study of, among other things, (a) affect as reinforcement to the robot (both internally generated as well as socially communicated; this chapter), (b) affect as perceptual feature to the robot (again internally generated and social), (c) affect resulting from reinforced robot behavior (see also Chapter 2), and (d) affect as meta-parameters for the robot's learning mechanism (Chapter 3 and 4). *EARL* can be seen as the concretization of the insights developed while researching the topics described in Chapters 2 to 5 of this thesis.

In this chapter, we focus on one aspect of *EARL*: the ability to model communicated affect by a human observer used as reinforcement by the robot. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. Emotional expression can therefore be a reinforcement signal to a child. We hypothesize that affective facial expressions facilitate robot learning, and compare a *social* setting with a *non-social* one to test this. The non-social setting consists of a simulated robot that learns to solve a typical RL task in a continuous grid-world environment. The social setting additionally consists of a human (parent) observing the simulated robot (child). The human's emotional expressions are analyzed in real time and converted to an additional reinforcement signal used by the robot; positive expressions result in reward, negative expressions in punishment. We quantitatively show that the "social robot" indeed learns to solve its task significantly faster than its "non-social sibling". We conclude that this presents strong evidence for the potential benefit of affective communication with humans in the Reinforcement Learning loop.

6.1 Introduction

In humans, emotion influences thought and behavior in many ways (Custers & Aarts, 2005; Damasio, 1994; Dreisbach & Goschke, 2004; Rolls, 1999). For example, emotion influences how humans process information by controlling the broadness versus the narrowness of attention (see also Chapter 3 and 4). Also, emotion functions as a social signal that communicates reinforcement of behavior in, e.g., parent-child relations. Computational modeling (including robot modeling) has proven to be a viable method of investigating the relation between emotion and learning (Broekens, Kosters & Verbeek, 2007; Gadanho, 2003), emotion and problem solving (Belavkin, 2004; Bothello & Coehlo, 1998), emotion and social robots (Breazeal, 2001; for review see Fong, Nourbakhsh & Dautenhahn, 2003), and emotion, motivation and behavior selection (Avila-Garcia & Cañamero, 2004; Blanchard and Cañamero, 2006; Cos-Aguilera et al., 2005; Velasquez, 1998). Although many approaches exist and much work has been done on computational modeling of emotional influences on thought and behavior, none explicitly targets the study of the relation between emotion and learning using a complete end-to-end framework in a Reinforcement Learning context[1]. By this we mean a framework that enables systematic *quantitative* study of the relation between affect and RL in a large variety of ways, including (a) affect as reinforcement to the robot (both internally generated as well as socially communicated), (b) affect as perceptual feature to the robot (again internally generated and social), (c) affect resulting from reinforced robot behavior, and (d) affect as meta-parameters for the robot's learning mechanism. In this chapter we present such a framework. We call our framework *EARL*, short for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning.

[1] Although the work by Gadanho (2003) is a partial exception as it explicitly addresses emotion in the context of RL. However, this work does not address social human input and social robot output.

Here we specifically focus on the influence of socially communicated emotion on learning in a Reinforcement Learning context. We show, using our framework *EARL*, that human emotional expressions can be used as an additional reinforcement signal used by a simulated robot.

The robot's task is to optimize food-finding behavior while navigating through a continuous grid-world environment. The grid world is not discrete, nor is an attempt made to define discrete states based on the continuous input. The grid world contains walls, path and food patches. The robot perceives its direct surroundings as they are. We have developed an action-based learning mechanism that learns to predict values of actions based on the current perception of the agent (note that in this chapter we use the terms agent and robot interchangeably). Every action has its own Multi-Layer Percepton network (see also Lin, 1993) that learns to predict a modified version of the *Q*-value (Sutton & Barto, 1998). We have used this setup so that observed robot behavior can be extrapolated to the real world; building the actual robot with appropriate sensors and actuators would, in theory, suffice to replicate the results. We explain our modeling method in more detail in Section 6.5.

As mentioned above, we study the effect of a human's emotional expression on the learning behavior of the robot. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. In this case, the emotional expression is used to setup an *affective communication channel* (Picard, 1997) and is used to communicate a reinforcement signal to a child. In this chapter we take *affect* to mean the positiveness versus the negativeness of a situation, object, etc. (see Rolls, 1999; Russell, 2003; and Broekens, Kosters & Verbeek, 2007, or Chapter 2 for a more detailed argumentation of this point of view). The human observes the simulated robot while it learns to find food, and affect in the human's facial expression is recognized by the robot in real time[2]. A smile is interpreted as communicating positive affect and therefore converted to a small additional reward (additional to the reinforcement the robot receives from its simulated environment). The expression of fear is interpreted as communicating negative affect and therefore converted to a small additional punishment. We call this the *social* setting. The non-social setting is a standard experimental Reinforcement Learning setup without human input.

[2] In this chapter, affect is thus a short-term signal elicited by an external source, as opposed to affect defined in Chapter 2 where it is a long-term signal elicited by mechanisms in the agent itself based on its learning performance.

We hypothesized that robot learning (in a RL context as described above) is facilitated by additional social reinforcement. Our experimental results support this hypothesis. We compared the learning performance of our simulated robot in the social and non-social settings, by analyzing averages of learning curves. The main contribution of this research is that it presents *quantitative* evidence of the fact that a human-in-the-loop can boost learning performance in real-time, in a plausible learning environment. We belief this is an important result. It provides a solid base for further study of human mediated robot learning in the context of real-world applicable Reinforcement Learning, using the communication protocol nature has provide for that purpose, i.e., emotional expression and recognition. Therefore, our results suggest that robots can be trained and their behaviors optimized using natural social cues. This facilitates human-robot interaction.

The rest of this chapter is structured as follows. In Section 6.2 we explain in some more detail our view of affect, emotion and how affect influences learning in humans. In Section 6.3 we briefly introduce *EARL*, our complete framework. In Section 6.4 we describe how communicated affect is linked to a social reinforcement signal. In Section 6.5, we explain our method of study (e.g., the grid world, the learning mechanism). Section 6.6 discusses the results and Section 6.7 discusses these in a broader context and presents concluding remarks and future work.

6.2 Affect as Reinforcement

As we have seen in the previous chapters, affect influences thought and behavior in a variety of ways. For example, a person’s mood influences processing style and attention, emotions influence how one thinks about objects, situations and persons, and emotion is related to learning behaviors as well as can be used to modify learning parameters in artificial learning agents. So, affect regulates behavior.

Affect also regulates behavior of others. Obvious in human development, expression (and subsequent recognition) of emotion is important to communicate (dis)approval of the actions of others. This is typically important in parent-child relations. Parents use emotional expression to guide behavior of infants. Emotional interaction is essential for learning. Striking examples are children with an autistic spectrum disorder, typically characterized by a restricted repertoire of behaviors and interests, as well as social and communicative impairments such as difficulty in joint attention, difficulty recognizing and expressing emotion, and lacking of a social smile (for review see Charman & Baird, 2002). Apparently, children suffering from this disorder have both a difficulty in building up a large set of complex behaviors *and* a difficulty understanding emotional expressions and giving the correct social responses to these. This disorder provides a clear example of the interplay between learning behaviors and being able to process emotional cues.

In this chapter we specifically focus on the influence of socially communicated affect on learning: we focus on the role of affect in guiding learning in a social human-robot setting. We use affect to denote the positiveness versus negativeness of a situation. We ignore the arousal a certain situation might bring. 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 *short term* timescale: i.e., to emotion. We hypothesize that affect communicated by a human observer can enhance robot learning. In our study we assume that the recognition of affect translates into a reinforcement signal. Thus, the robot uses a *social reinforcement* in addition to the reinforcement it receives from its environment while it is building a model of the environment using Reinforcement Learning mechanisms. In the following sections we first explain our framework after which we detail our method and discuss results and further work.

6.3 *EARL*: A Computational Framework to Study the Relation between Emotion, Adaptation and Reinforcement Learning.

To study the relation between emotion, adaptation and Reinforcement Learning, we have developed an end-to-end framework. The framework consists of four parts:

- An emotion recognition module, recognizing emotional facial expression in real time.
- A Reinforcement Learning agent to which the recognized emotion can be fed as input.
- An artificial emotion module slot, this slot can be used to plug in different models of emotion into the learning agent that produce the artificial emotion of the agent as output. The modules can use all of the information that is available to the agent (such as action repertoire, reward history, etc.). This emotion can be used by the agent as intrinsic reward, as metalearning parameter, or as input for the expression module.
- An expression module, consisting of a robot head with the following degrees of freedom: eyes moving up and down, ears moving up and down on the outside, lips moving up and down, eyelids moving up and down on the outside, and RGB eye colors.

Emotion recognition is based on quite a crude mechanism using the face tracking abilities of OpenCV[3]. It uses 9 points on the face each defined by a blue sticker: 1 on the tip of the nose, 2 above each eyebrow, 1 at each mouth corner and 1 on the upper and lower lip. The recognition module is configured to store multiple prototype point constellations. The user is prompted to express a certain emotion and press space while doing so. For every emotional expression (in the case of our experiment neutral, happy and afraid), the module records the positions of the 9 points relative to the nose. This is a prototype point vector. After configuration, to determine the current emotional expression in real time, the module calculates a weighted distance from the current point vector (read in real-time from a web-cam mounted on the computer screen) to the prototype vectors. Different points get different weights. This results in an error measure for every prototype expression. This error measure is the basis for a normalized vector of recognized emotion intensities. The recognition module sends this vector to the agent (e.g., neutral 0.3, happy 0.6, fear 0.1). Our choice of weights and features has been inspired by work of others (for review see Pantic & Rothkrantz, 2000). Of course the state of the art in emotion recognition is more advanced than our current approach. However, as our focus is affective learning and not the recognition process per se, we contented ourselves with a low fidelity solution (working almost perfectly for neutral, happy and afraid, when the user keeps the head in about the same position).

Note that we do not aim at generically recognizing emotional expressions. Instead, we tune the recognition module to the individual observer to accommodate his/her personal and natural facial expressions.

The Reinforcement Learning agent receives this recognized emotion and can use this in multiple ways: as reward, as information (additional state input), as metaparameter (e.g., to control learning rate), and as social input directly into its emotion model. In this chapter we focus on social reinforcement, in particular on the recognized emotion being used as additional reward or punishment. The agent, its learning mechanism and how it uses the recognized emotion as reinforcement are detailed in Sections 6.4 and 6.5.

The artificial emotion model slot enables us to plug in different emotion models based on different theories to study their behavior in the context of Reinforcement Learning. For example, we have developed a model based on the theory by Rolls (1999), who argues that many emotions can be related to reward and punishment and the lack thereof. This model enables us to see if the agent's situation results in a plausible (e.g., scored by a set of human observers) emotion

[3] http://www.intel.com/technology/computing/opencv/index.htm

emerging from the model. By scoring the plausibility of the resulting emotion, we can learn about the compatibility of, e.g., Rolls' emotion theory with Reinforcement Learning. However, in the current study we have not used this module, as we focus on affective input as social reward.

The emotion expression part is a physical robot head. The head can express an arbitrary emotion by mapping it to its facial features, again according to a certain theory. Currently our head expresses emotions according to the Pleasure Arousal Dominance (PAD) model by Mehrabian (1980). We have a continuous mapping from the 3-dimensional PAD space to the features of the robot face. As such we do not need to explicitly work with emotional categories or intensities of the categories. The mapping appears to work quite well, but is in need of validation study (again using human observers). We have not used the robot head for the studies reported upon in this chapter.

We now describe in detail how we coupled the recognized human emotion to the social reinforcement signal for the robot. Then we explain in detail our adapted Reinforcement Learning mechanism (such that it enabled learning in continuous environments), and our method of study as well as our results.

6.4 Emotional Expressions as Reinforcement Signal.

As mentioned earlier, emotional expressions and facial expressions in particular can be used as social cues for the desirability of a certain action. In other words, an emotional expression can express reward and punishment if directed at an individual. We focus on communicated affect, i.e., the positiveness versus negativeness of the expression. If the human expresses a smile (happy face) this is interpreted as positive affect. If the human expresses fear, this is interpreted as negative affect. We interpret a neutral face as affectless.

We have studied the mechanism of communicated affective feedback in a human-robot interaction setup. The human's face is analyzed (as explained above) and a vector of emotional expression intensities is fed to the learning agent. The agent takes the expression with the highest intensity as dominant, and equates this with a *social reward* of, e.g., 2 (happy), −2 (fear) and 0 (neutral). This is obviously a simplified setup, as the human face communicates much more subtle affective messages and at the very least is able to communicate the degree of reward and punishment. However, to investigate our hypothesis (affective human feedback increases robot learning performance), the just described mechanism is sufficient.

The social reward is simply added to the "normal" reward the agent receives from the environment. So, if the agent walks on a path somewhere in the grid world, it receives a reward (say 0), but when the user smiles, the resulting actual reward becomes 2, while if the user looks afraid, the resulting reward becomes −2.

6.5 Method

To study the impact of social reinforcement on robot learning, we have used our framework in the following experimental setup.

A simulated robot (agent) "lives" in a continuous grid-world environment consisting of wall, food and path patches (Figure 6.1). These are the features of the world observable by the agent. The agent cannot walk on walls, but can walk on path and food. Walls and path are neutral (have a reinforcement of 0.0), while food has a reinforcement of 10. One cell in the grid is assumed to be a 20 by 20 spatial unit object (let's say 20 x 20 centimeters). Even though wall, path and food are placed on a grid, the world is continuous in the following sense: the agent moves by turning or walking in a certain direction using an arbitrary speed (in our experiments set at 3 spatial units per time unit), and perceives its direct surroundings (within a radius of 20 spatial units) according to its looking direction (one out of 16 possible directions).

The agent uses a "relative eight-neighbor metric" meaning that it perceives features of the world at 8 points around it, with each point at a distance of 20 from the center point of the agent and each point at an interval of 1/4 PI radians, with the first point always being exactly in front of it (Figure 6.1).

The state perceived by the agent (its percept) is a real-valued vector of inputs between 0 and 1; each input is defined by the relative contribution of a certain feature in the agent-relative direction corresponding to the input. For example, if the agent sees a wall just in front of it (i.e., the center point of a wall object is exactly at a distance of 20 units as measured from the current agent location in its looking direction) the first value in its perceived state would be equal to 1. This value can be anywhere between 0 and 1 depending on the distance of that point to the feature. For the three types of features, the agent thus has 3x8=24 real-valued inputs between 0 and 1 as its perceived world state *s* (Figure 6.1). Therefore the agent can approach objects (e.g., a wall) from a large number of possible angles and positions, with every intermediate position being possible.

For all practical purposes, the learning environment can be considered continuous. States are not discretized to facilitate learning. Instead we chose to use the perceived state as is, to maximize compatibility of our experimental results with real-world robots. However, Reinforcement Learning in continuous environments introduces several important problems for standard RL techniques, such as Q-learning, mainly because a large number of potentially similar states exist as well as a very long path length between start and goal states can occur making value propagation difficult.

We now briefly explain our adapted RL mechanism. As RL in continuous environments is not specifically the topic of the chapter we have left out some of the rational for our choices.

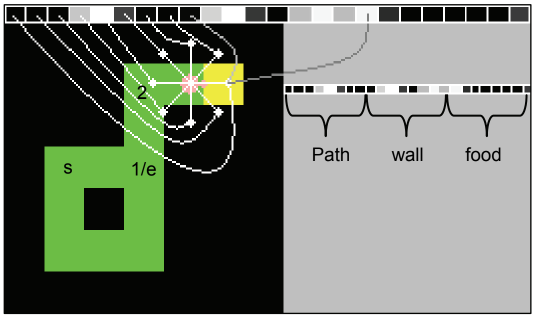

Figure 6.1. The experimental grid world. The agent is the "circle with nose" in the top right of the maze, In this figure the agent is looking to the right. The 8 white dots denote the points perceived by the agent. These points are connected to the elements of state s (neural input to the MLPs used by the agent) as depicted. This is repeated for all possible features, in our case: path (gray), wall (black), and food (light gray), in that order (as depicted in the smaller representation of the neural network). The "e" denotes the cell in which social reward can be administered through smiling or expression of fear, the "1" and "2" denote key locations at which the agent has to learn to differentiate its behavior, i.e., either turn left ("1") or right ("2"). The agent starts at "s". The task enforces a non-reactive best solution (by which we mean that there is no direct mapping from reward to action that enables the agent to find the shortest path to the food). If the agent would learn that turning right is good, it would keep walking in circles. If the agent learns that turning left is good, it would not get to the food

The agent learns to find the path to the food, and optimizes this path. At every step the agent takes, the agent updates its model of the expected benefit of a certain action as follows. It learns to predict the value of actions in a certain perceived state s, using an adapted form of Q-learning. The value function, $Q_a(s)$, is approximated using a multilayer perceptron (MLP), with 3x8=24 input, 24 hidden, and one output neuron(s), with s being the real-valued input to the MLP, a the action to which the network belongs, and the output neuron converging to $Q_a(s)$. As a result, every action of the agent (5 in total: forward, left, right, left and forward, right and forward) has its own network (see also Gadanho, 1999). The output of the action networks are used as action values in a standard Boltzmann action-selection function (Sutton & Barto, 1998). An action network is trained on the Q-value—i.e., $Q_a(\text{s}) \leftarrow Q_a(s)+\alpha(r+\gamma Q(s')-Q_a(s))$—where r is the reward resulting from action a in state s, s' is the resulting next state, $Q(s')$ the value of state s', α is the learning rate and γ the discount factor (Sutton & Barto, 1998).

The learning rate equals 1 in our experiments (because the learning rate of the MLP is used to control speed of learning, not α), and the discount factor equals 0.99. To cope with a continuous grid world, we adapted standard Q-learning in the following way:

First, the value $Q_a(s)$ used to train the MLP network for action a is topped such that $min(r,\ Q_a(s'))<=Q_a(s)<=max(r,\ Q(s'))$. As a result, individual $Q_a(s)$ values can never be larger or smaller than any of the rewards encountered in the world. This enables a discount factor close to or equal to 1, needed to efficiently propagate back the food's reward through a long sequence of steps. In continuous, cyclic, worlds, training the MLP on normal Q-values using a discount factor close to 1 can result in several problems not further discussed here.

Second, per step of the agent, we train the action-state networks not only on $Q_a(s)\leftarrow Q_a(s) +\alpha(r+\gamma Q(s') -Q_a(s))$ but also on $Q_a(s') \leftarrow Q_a(s')$. The latter seems unnecessary but is quite important. RL assumes that values are propagated *back*, but MLPs generalize while trained. As a result, training an MLP on $Q_a(s)$ also influences its value prediction for s' in the same direction, just because the inputs are very close. In effect, part of the value is actually propagated *forward*; credit is partly assigned to what comes next. This violates the RL assumption just mentioned. Note that the value $Q(s')$ is predicted using another MLP, called the value network, that is trained in the same way as the action networks using the topped-off value and forward propagation compensation.

Third, for the agent to better discriminate between situations that are perceptually similar, such as position "1" and "2" in Figure 1, for each action-network the agent also uses a second network trained on the value of *not* taking the action. This network is trained when other actions are taken but not when the action to which the "negation" network belongs is taken. In effect, the agent has two MLPs per action. This enables the agent to better learn that, e.g., "right" is good in situation "2" but *not* in situation "1". Without this "negation" network, the agent learns much less efficient (results not shown). To summarize, our agent has 5 actions, it has 11 MLPs in total: one to train $Q(s)$, 5 to train $Q_a(s)$ and 5 to train $Q_{-a}(s)$. All networks use forward propagation compensation and a topped-off value to train upon. The MLP predictions for $Q_a(s)$ and $Q_{-a}(s)$ are simply added, and the result is used for action selection.

To study the effect of communicated affect as social reward, we created the following setup. First an agent is trained without social reward. The agent repeatedly tries to find the food for 200 trials, i.e., one *run*. The agent continuously learns and acts during these trials. To facilitate learning, we use a common method to vary the MLP learning rate and the Boltzmann action selection β derived from simulated annealing. The Boltzmann β equals to 3+(*trial*/200)*(6−3), effectively varying from 3 (exploration) in the first trial to 6 (exploitation) in the last. The MLP learning rate equals 0.1−(*trial*/200)*(0.1−0.001) effectively varying from 0.1 in the first trial to 0.001 in the last. We repeated the experiment 200 times, resulting in 200 runs. Average learning curves are plotted for these 200 runs using a linear smoothing factor equal to 6 (Figure 6.2).

Second, a new agent is trained *with* social reinforcement, i.e., a human observer looking at the agent with his/her face analyzed by the agent, translating a smile to a social reward and a fearful expression to a social punishment. Again, average learning curves are plotted using a linear smoothing factor equal to 6, but now based on the average per trial over 15 runs (Figure 6.2). We experimented with three different social settings: (a) a moderate social reinforcement, r_{human}, from trial 20 to 30, where the social reinforcement is either −0.5 or 0.5 (happy vs. fearful, respectively); (b) a strong social reinforcement, r_{human}, from trial 20 to 25 where social reinforcement is either −2 or 2, i.e., more extreme social reinforcement but for a shorter period; (c) a social reinforcement, r_{human}, from trial 29 to 45 where social reinforcement is either −2 or 2 while (in addition to settings *a* and *b*) the agent trains an additional MLP to predict the direct social reinforcement, r_{human}, based on the current state s. The MLP is trained to learn $R_{social}(s)$ as given by the human reinforcement r_{human}. After trial 45, the direct social reinforcement from the observer, r_{human}, is replaced by the learned social reinforcement $R_{social}(s)$. So, during the critical period (the trial intervals mentioned) of social setting *a*, *b* and *c*, the total reinforcement is a composite reward equal to $R(s)+r_{human}$. Only in setting *c*, and only after the critical period until the end of the run, the composite reward equals $R(s)+R_{social}(s)$. In all other periods, the reinforcement is as usual, i.e., $R(s)$. As a result, in setting *c* the agent can continue using an additional social reinforcement signal that has been learned based on what its human tutor thinks about certain situations.

The process of giving affective feedback to a Reinforcement Learning agent appeared to be quite a long, intensive and attention absorbing experience. As a result, it was physically impossible to observe the agent during all runs and all trials in the entire grid world (after 2 hours of smiling to a computer screen one is exhausted *and* has burning eyes and painful facial muscles). To be able to test our hypothesis, we restricted social input to the cell indicated by ‘e’ (Figure 6.1). Only when the agent moves around in this cell (and is in a social input trial as defined by the social settings described above), the simulation speed of the experiment is set to one action per second enabling human affective feedback.

6.6 Results

The results clearly show that learning is facilitated by social reward. In all three social settings (Figure 6.2a, b and c) the agent needs fewer steps to find the food during the trials in which the observer provides assistance to the agent by expressing positive or negative affect. Interestingly, at the moment the observer stops giving social rewards, the agent gradually looses the learning benefit it had accumulated. This is independent of the size of the social reward (both social learning curves in Figure 6.2a and b show dips that eventually return to the non-social learning curve). This can be easily explained. The social reward was not given long enough for the agent to internalize the path to the food (i.e., propagate back the food's reward to the beginning of the path). As soon as the observer stops giving social rewards, the agent starts to forget these rewards, i.e., the MLPs are again trained to predict values as they are without social input. So, either the observer should continue to give social rewards until the agent has internalized the solution, or the agent needs to be able to build a representation of the social reward function and use it when actual social reward is not available. We have experimented with the second (social setting *c*): we enabled the agent to learn the social reward function. Now the agent uses actual social reward at the emotional input spot ('e', Figure 6.1) during the critical period, and uses its social reward prediction when social input stops. This is the third social setup. Results clearly show that the agent is now able to keep the benefit it had accumulated from using social rewards (Figure 6.2c). These results show that a combination of using social reward and learning a social reward function facilitates robot learning, by enabling the robot to quicker learn the optimal solution to the food due to the direct social reward as well as keep that solution by using its learned social reward function when social reward stops.

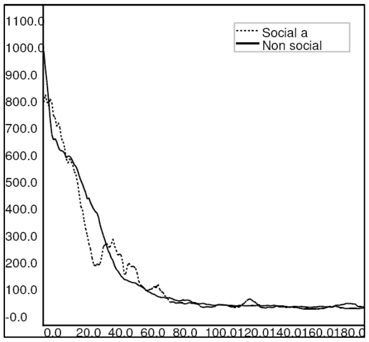

Figure 6.2a. Results of the learning experiment where the social setting *a* is compared with the non-social setting. In social setting *a*, social input is given between trial 20 and 30, where the social reward is either −0.5 or 0.5 (happy vs. fearful, respectively). On the *x*-axis the number of times the food is found is shown (trials); on the *y*-axis the average number of steps needed to find the food is shown.

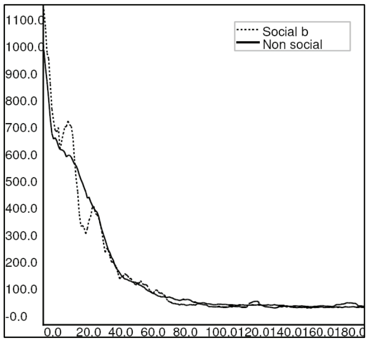

Figure 6.2b. Results of the learning experiment where the social setting *b* is compared with the non-social setting. In setting *b*, the social input is given between trial 20 and 25 where social reward is either −2 or 2, i.e., more extreme social rewards but for a shorter period. Axes are as in the previous figure.

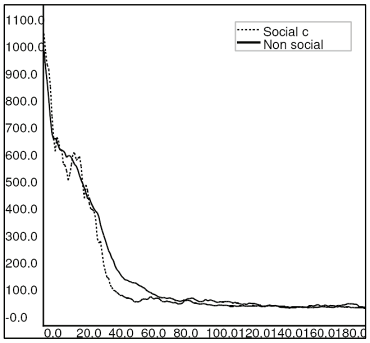

Figure 6.2c. Results of the learning experiment where the social setting *c* is compared with the non-social setting. In setting *c*, social input is given between trial 29 to 45, where social reward is either −2 or 2. The agent trains an additional MLP to predict the social reward. Axes are as in the previous figure.

6.7 Conclusion, Discussion and Further Work

Our results show that affective interaction in human-in-the-loop learning can provide significant benefit to the efficiency of a Reinforcement Learning robot in a continuous grid world. We believe our results are particularly important to human-robot interaction for the following reasons. First, advanced robots such as robot companions, robot workers, etc., will need to be able to adapt their behavior according to human feedback. For humans it is important to be able to give such feedback in a natural way, e.g., using emotional expression. Second, humans will not want to give feedback all the time, it is therefore important to be able to define critical learning periods as well as have an efficient social reward system. We have shown the feasibility of both. Social input during the critical learning periods was enough to show a learning benefit, and the relatively easy step of adding an MLP to learn the social reward function enabled the robot to use the social reward when the observer is away.

We have specifically used an experimental setup that is compatible with a real-world robot: we have used continuous inputs and MLP-based training of which it is known that it can cope with noise and generalize over training examples. We believe our results can be generalized to real-world robotics. However, this most certainly needs to be experimented with.

Many interesting computational approaches exist that study emotion in the context of robots and agents, of which we mention one explicitly here as it is particularly related to our work: the adaptive, social chatter bot *Cobot* (Isbell et al., 2001). Cobot learns the information preferences of its chat partners, by analyzing the chat messages for explicit and implicit reward signals. These signals are then used to adapt its model of providing information to that chat partner. So, Cobot effectively uses social feedback as reward, as does our simulated robot. However, there are several important differences. Cobot does not address the issue of a human observer parenting the robot using affective communication. Instead, it learns based on reinforcement extracted from words used by the user during the chat sessions in which Cobot is participating. Also, Cobot is not a real-time behaving robot, but a chat robot. As a consequence, time constraints related to the exact moment of administering reward or punishment are less important. Finally Cobot is restricted regarding its action-taking initiative, while our robot is continuously acting, with the observer reacting in real-time.

Future work includes a broader evaluation of the *EARL* framework including its ability to express emotions generated by an emotional model plugged into the RL agent. Further, it is interesting to experiment with controlling meta parameters (such as exploration/exploitation and learning rate) based on the agent's internal emotional state or social rewards, as has been done in the discrete grid-world case in Chapter 3 and 4. Currently we use simulated annealing-like mechanisms to control these parameters.

Further, the agent could try to learn what an emotional expression predicts. In this case, the agent would use the emotional expression of the human in a more pure form (e.g., as a real-valued vector of facial feature intensities as part of its perceived state *s*). This might enable the agent to learn what the emotional expression means for itself instead of simply using it as reward.

Finally, a somewhat futuristic possibility is actually quite close: affective Robot-Robot interaction. Using our setting, it is quite easy to train one robot in a certain environment (parent), make it observe an untrained robot in that same environment (child), and enable it to express its emotion as generated by its emotion model using its robot head, an expression recognized and translated into social rewards by the child robot. Apart from the fact that it is somewhat dubious if such a setup is actually useful (why not send the social reward as a value through a wireless connection to the child), it would enable robots to use the same communication protocol as humans.

Regarding the "usefulness" argument just put forward, it seems to apply to our experiment as well. Why didn't we just simulate affective feedback by pushing a button for positive reward and pushing another for negative reward (or even worse, by simulating a button press)? From the point of view of the robot this is entirely true, however, from the point of view of the human—and therefore the point of view of the human-robot interaction—not at all. Humans naturally communicate social signals using their face, not by pushing buttons. The process of expressing an emotion is quite different from the process of pushing a button, even if it was only for the fact that it takes more time and effort to initiate the expression and that the perception of an expression is the perception of a process and not of a discrete event (like a button press). In a real-world scenario with a mobile robot in front of you it would be quite awkward to have to push buttons instead of just smile when you are happy about its behavior. Further it would be quite useful if the robot could recognize you being happy or sad, and gradually learn to adapt its behavior even when you did not intentionally give it a reward or punishment. Abstracting away from the actual affective interaction patterns between the human and the robot in our experiment would have rendered the experiment almost completely trivial. Nobody would be surprised to see that the robot learns better if an intermediate reward is given halfway its route towards food. Our aim was to investigate if affective communication can enhance learning in a Reinforcement Learning setting. Taking out the affective part would have been quite strange indeed.

7

Affect and Formal Models

Formalizing Cognitive Appraisal Theory

In this chapter we take a theoretical approach towards computational modeling of emotion. Affect in this chapter is thus interpreted in a broader sense, as in *related to emotion*. This is different from the interpretations of affect presented in Chapter 2 to 6. To avoid any potential misunderstanding, in this chapter we use the term emotion, not affect. We present a formal way in which emotion theories can be described and compared with the computational models based upon them. We apply this formal notation to cognitive appraisal theory, a family of cognitive theories of emotion, and show how the formal notation can help to advance appraisal theory and help to evaluate computational models based on cognitive appraisal theory: the main contributions of this chapter. Although this chapter is quite different from the others, it fits within the general approach: that is, the use of computational models to evaluate emotion theories. As such it can be viewed as a high-level analysis of issues associated with computational modeling of emotion.

Cognitive appraisal theories (CATs) explain human emotions as a result of the subjective evaluation of events that occur in the environment. Recently, arguments have been put forward that discuss the need for formal descriptions in order to further advance the field of cognitive appraisal theory. Formal descriptions can provide detailed predictions and help to integrate different CATs by providing clear identification of the differences and similarities between theories. A computational model of emotion that is based on a CAT also needs formal descriptions specifying the theory on which it is based. In this chapter we propose a formal notation for the declarative semantics of the structure of appraisal. We claim that this formalism facilitates both integration of appraisal theories as well as the design and evaluation of computational models of emotion based on an appraisal theory. To support these claims we show how our formalism can be used in both ways: first we integrate two appraisal theories; second, we use this formal integrated model as basis for a computational model after identifying what declarative information is missing in the formal model. Finally, we embed the computational model in an emotional agent, and show how the formal specification helps to evaluate the computational model.

7.1 Introduction

Computational models of emotion are used in a wide variety of artificial emotional agents. In general, such a model is based on a cognitive appraisal theory (CAT) (note that the model of affect and affective feedback we have used in Chapter 2 to 5 are not based on cognitive appraisal theory). CATs explain human emotions as a result of the subjective evaluation of events. However, such theories typically lack the necessary detail to base a computational model upon (Gratch & Marcella, 2004). As a result, it is difficult to evaluate if the computational model correctly implements the theory.

Further, to advance the field of appraisal theory, it is essential that cognitive appraisal theories can be integrated and compared with each other. Thus, building computational models of emotion *and* advancing the field of appraisal theory are in need of a representation of appraisal theory that enables systematic analysis. This is the focus of our chapter.

More specific, we propose a formalism to describe the structure of appraisal. That is, we propose a formal notation for the behavior of processes that play a role in appraising a situation, how these processes are linked to each other, what the resulting emotions could be, etc. In this chapter we show that different cognitive appraisal theories can be described using the same formal notation, that such formal representations can be used to compare and integrate CATs and that the formal representation can be used to systematically analyze computational models of emotion.

Such formal description of a specific CAT can be used, for example, to prove that the happy expression on the face of a child, that just noticed it arrived at a large rollercoaster park with extremely exciting rollercoasters and a couple of flags, must be due to an appraisal of the situation that involves the expectancy of intrinsic pleasantness. If I would have a robot, the formalism can be used in approximately the same way. While developing the robot, I would use the formalism to understand why it shows a certain emotion. Assuming a specific CAT, the formalism can be used to decide whether its artificial emotion of fear is potentially correct after I have proposed to go to a rollercoaster park. At first, I might be tempted to start to debug the robot, but the formal description of the CAT on which its emotions are based can show me that its emotion might be genuine as it potentially results from a negative appraisal of the rain (reflecting its fear to rust).

This informal introduction gives some intuition for the need and use of formal representations of appraisal theory. In this chapter we propose a formalism to describe the structure of appraisal (Section 7.3) and we elaborate on two ways in which this formalism can be used: (1) we use it to integrate two different appraisal theories (Section 7.4), and (2) we use it to analyze a computational model of emotion we developed (Section 7.5). Before continuing the main line of this chapter, we first give a cognitive definition of emotion, some more detail on the development and use of artificial emotional agents, and a more detailed description of the problem we address.

7.1.1 Emotion

In cognitive psychology, emotion is often defined as a psychological state or process that functions in the management of goals, needs, desires and concerns of an individual (we refer to these four terms as *goals*). This state consists of physiological changes, feelings, expressive behaviors, cognitive activity and inclinations to act (e.g., Roseman & Smith, 2001). Emotion is elicited by the evaluation of an event in relation to the accomplishment of the agent's goals. Thus, an emotion is a heuristic that relates events to the agent's goals (Oatley, 1999). Additionally, emotions are used in non-verbal communication.

7.1.2 Artificial Emotions

Inspired by this heuristic and communicative aspect of emotion, computational models of emotion are embedded in a variety of intelligent agents. The development of artificial emotional agents is useful, and can be applied to a wide variety of domains. These domains include electronic tutors (Heylen et al., 2003), human-robot interaction (Breazeal, 2001; Chapter 5), virtual agents in VR training environments (Henninger et al., 2002), agents targeted at decision-making and planning (Coddington & Luck, 2003) and adaptive agents that use emotion or affect to control learning parameters (Belavkin, 2004; Chapter 3-4). For example, research shows that a robot's emotional expression influences human caretaking behavior (Breazeal, 2001), of which the following is a nice anecdote. When human subjects interacted with Kismet (the emotional robot) and Kismet reacted sad or distressed to the actions of the human, the subjects were visibly distressed and looked questioning to the researchers as if they wanted to say "am I doing something wrong?" A second example is a recent study by Partala and Surakka (2004) that shows that affective intervention in human-computer interaction has a positive effect on the human, both emotionally as well as in terms of the subject’s problem solving performance. Positive words resulted in smiling as well as better problem solving performance.

7.1.3 Cognitive Appraisal Theory

The majority of computational models of emotion embedded into intelligent agents are based on cognitive appraisal theory. Such theories of emotion attempt to explain why a certain event results in one emotional response rather than another and why a certain emotion can be elicited by different events. The key concept of most CATs is that the subjective cognitive evaluation of events in relation to the agent's goals is responsible for emotion (Roseman & Smith, 2001). More generically one can say that events have to be evaluated as having personal meaning or relevance (van Reekum, 2000). This evaluation is called *appraisal*. It is generally accepted that physiological changes and other non-cognitive factors can influence the actual appraisal of events. Although previously most appraisal theories assumed that appraisal was a necessary and sufficient condition for emotion (Roseman & Smith, 2001), currently it is seen as an important component of emotion.

7.1.4 How to Interpret Artificial Emotions in Relation to a CAT?

The “brain” of artificial intelligent agents is often based on a belief-desire-intention (BDI) architecture (Jennings, Sycara & Wooldridge, 1998). If cognitive evaluation of events in relation to the agent's goals is sufficient for emotion, then the addition of such an evaluation of events related to the beliefs, desires and intentions of an artificial agent is sufficient for computational emotions. This partly explains the current popularity of appraisal theories as basis for emotional agents.

However, appraisal theories are currently described in a way that is insufficiently precise as a specification for a computational model of emotion (Gratch & Marsella, 2004). As a result, many computational models are inspired by structural theories of appraisal—i.e., theories that describe the structural relations between events, appraisal processes and emotions—and implemented using artificial intelligence mechanisms. During implementation, designers are forced to make many assumptions about the exact mechanisms of appraisal. This results in a large gap between the structural theory of appraisal and the resulting computational model of emotion.

In addition to this, artificial agents have a more and more complex design. These agents are approaching a point at which inspection of the agent's program and internal state is no longer efficient to "debug" the agent’s design. We predict that in the future it will no longer be feasible to try to understand an agent's unexpected behavior by purely investigating its inner workings. Instead, a formal investigation of its behavior will be a necessary component of this process of understanding (Broekens & DeGroot, 2006), just like we need to ask a person about why he/she does something instead of *only* looking at neuroimaging data.

7.1.5 Advancing Appraisal Theory Needs Comparison and Integration

Apart from the problem of using appraisal theories as basis for computational models, another problem—directly related to appraisal theory—exists. Although most appraisal theories share the assumption that cognitive appraisal is an important part of emotion, many different appraisal theories exist (Reisenzein, 2001; Frijda & Mesquita, 2000; Smith and Kirby, 2000; Scherer 2001). Comparison between, and convergence of these theories is difficult, but important in order to advance the field of appraisal theory. Formalization of structural theories of appraisal can help to solve these problems in two different ways. First, formal descriptions facilitate comparison, convergence and integration of theories, because assumptions and relations between concepts are clarified (Wehrle & Scherer, 2001). Second, computational modeling of emotion is a powerful way of analyzing appraisal theories in a formal way (Wehrle & Scherer, 2001). Formal descriptions facilitate the evaluation of computational models, thereby contributing to the analysis of appraisal theories.

7.1.6 Aim and Scope of This Chapter

The main contribution of this chapter is an abstract-level, theory independent, set-based formalism that can be used to describe the structure of appraisal as describe by a cognitive appraisal theory. This formalism addresses the two issues introduced above.

- First, how can we advance cognitive appraisal theory? We argue that our formalism facilitates comparison and integration of CATs. We use our formalism as a tool to integrate the Stimulus Evaluation Check theory (Scherer, 2001) and Appraisal Detector Model (Smith & Kirby, 2000), two prominent and recent CATs. Our formalism can be used to describe the behavior of the processes involved in appraisal. It does not address the issue of how to formally describe and reason about what a certain emotion *is* in terms of *specific* beliefs, desires and intentions of a BDI agent (e.g., Meyer, 2004).
- Second, how to formally specify a structural appraisal theory, so that the resulting formal description can be used as basis for the specification and evaluation of the emotional behavior of an artificial agent? We argue that our formalism narrows the gap between appraisal theory and computational model, and we show how such a formal specification can be used as basis for a computational model of emotion we have developed. We also show how this specification helps to evaluate the computational model.

The structure of this chapter is as follows. First, we introduce the relation between computational models, structural theories of appraisal and process theories of appraisal. Then we introduce the actual formalism in Section 7.3. While the introduction of Section 7.3 is essential for understanding the rest of the chapter, the parts that detail the formalism are recommended to the mathematically oriented reader. Less mathematically oriented readers will find Section 7.4—showing how the formalism can be used as a tool to facilitate the integration of appraisal theories— as well as Section 7.5—demonstrating how a formal description of a structural model of appraisal can be used as basis for a computational model—more interesting. Section 7.6 discusses issues around formalization, and related approaches.

7.2 Appraisal Theory: Structure, Process and Computation

A common classification of CATs is based on a structural versus a process-based description (Roseman & Smith, 2001). Structural theories of appraisal (also called black-box models or structural models) describe the structural relations between:

- the environment of an agent and perception of this environment: *perception*;
- the agent's appraisal processes that interpret the perceived environment in terms of values on a set of subjective measures, called appraisal dimensions. An appraisal dimension influences emotion and can be considered as a variable—e.g., agency or valence—, used to express the result of the appraisal of a perceived object—e.g., a friend. This process of evaluation is called *appraisal*;
- the processes that relate these values to the agent's emotions: *mediation*.

Process theories of appraisal describe, in detail, the cognitive operations, mechanisms and dynamics by which the appraisals, as described by the structural theory, are made and how appraisal processes interact (Reisenzein, 2001). In other words, a structural theory of appraisal aims at describing the declarative semantics of appraisal, while a process theory of appraisal complements this description with procedural semantics. In this chapter we adopt the terms *structural model* and *process model* respectively, and use *appraisal theory/model* when referring to cognitive theories/models of appraisal in general.

A computational model is a model that is composed of operations that unambiguously control the behavior of a device. These operations may use available input data. If there is a sequence of such operations that maps a specific input to a specific behavior (output), an *algorithm* is said to exist for that mapping. The devices are essentially serial, but parallel execution can be either simulated in one such device using threading, or effectuated using multiple communicating devices. In this chapter, we define a *computational model* as a structured collection of interacting algorithms that operate serially or in parallel, with operations that are eventually reducible to the Turing machine level.

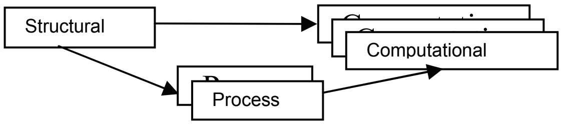

Figure 7.1. Three possible mappings between structural, process and computational models of emotion.

In Broekens and DeGroot (2006) we have analyzed the relation between cognitive appraisal theory and computation. We have argued that it is useful to have a theory-independent formal notation to describe structural appraisal theories (i.e., the behavior of processes that play a role in appraising a situation, how these processes are linked to each other, what the resulting emotions could be, etc.). For clarity, we summarize the conclusion here.

In general, there is a generic-to-specific relation between structural, process and computational models. Structural models are the basis of computational- and process models, and process models are also the basis of computational models. In this case "basis of" usually means that a model *A* that is the basis of a model *B* contains less details than model *A*, and therefore different model *B* instantiations are possible based on model *A* (Figure 7.1). Although this is true in general, in Broekens and DeGroot (2006) we have argued that the difference between a structural, process and computational description is also one of kind, not just of different degrees of detail; all three models are equally important for cognitive appraisal theory. We have also shown that a formal description of the structural model is needed for the following reasons:

- to advance appraisal theory. A formal description facilitates comparison, convergence and integration of appraisal theories, and the process of formalization helps theory refinement;
- to build computational models of emotion based on structural theories of appraisal. First, process models of appraisals should coexist with computational models, not take their place. Second, before designing computational models at the algorithmic level, declarative information is needed on the processes that are responsible for perception, appraisal and mediation as defined by the appraisal theory. Third, objective information is needed to evaluate the consistency between computational model and appraisal theory, and reuse of this information seems very useful. We need a declarative description of the processes that are responsible for an agent's emotion, in order to evaluate if the agent's *unexpected* emotion resulting from an experimental situation is due to a problem in the agent's architecture, or

due to a mismatch between our interpretation of the situation and the agent's interpretation.

Typically, a common formal notation should enable formal description of a structural model such that this description includes the following data (of which many are also relevant to process models; Reisenzein, 2001):

- What is the nature and level (van Reekum, 2000) of processes; deliberative, automatic, innate?
- What is the relation between (results of) perception and appraisal processes.
- When and how are these processes activated? Are there thresholds? Can activation be sub-threshold?
- What kind of input and output (representations) a certain process needs/produces?
- Does a process continuously output results or periodically (how often)?
- How many and what perception, appraisal and mediating processes exist?
- Is information activation binary or gradual? E.g., how strongly must a certain event be perceived for it to be input for a certain appraisal process?
- What is the number of different appraisal dimensions, their activation range and the responsible processes?

7.3 A Set-Based Formalism for the Structure of Appraisal

In this section we introduce the basic concepts of the formalism we propose to describe structural theories of appraisal. Later sections explain its use in some detail. Our formalism is set-based and built around sets of perception processes, appraisal processes and mediating processes (Figure 7.2). The notation used for these three types of processes and the accompanying terminology are borrowed from Reisenzein (2001). The external world, *W*, is the set of all events and objects that can respectively occur and reside in the environment. Perception processes, the set *P*, filter, select and translate information from the external world, and produce *mental objects*—representations of the external world suitable for appraisal. We define the set of mental objects produced by the perception processes, the set *O*, as the current content of working memory. Appraisal processes, the set *A*, evaluate the mental objects produced by the perception processes and assign a combination of appraisal dimension values, the set *V*, to these objects. Mediating processes relate appraisal information to emotions. Thus, mediating processes, the set *M*, relate appraisal dimension values to emotion-component intensities, the set *I*.

Perception processes also perceive the agent's current appraisal dimension values and current emotion components. These two kinds of information are translated to mental objects. Since in our formalism only perception processes can put information in working memory, the emotion-component intensities, *I*, and appraisal information, *V*, must be perceived before the agent is able to use these two kinds of information in appraisal. This is consistent with the idea that appraisal is a cognitive evaluation of perceived objects in working memory. Additionally, the separation between cognitive emotional information—i.e., *V* and *I* perceived by *P*—and non-cognitive emotional information—*I* influencing *A*—enables the specification of appraisal processes that are biased by a specific combination of emotional feedback (i.e., none, non-cognitive, cognitive, or both). This enables, for example, explicit specification of appraisal structures involved in coping, re-appraisal and strategic use of emotions. This ability is important for the completeness of our formalism.

To describe the *structural relations* between elements in the sets of perception, appraisal and mediating processes, our formalism allows the specification of process dependencies. For example, some process dependencies can be defined as excitatory relations, while others can be defined as inhibitory relations between processes.

The concepts of the formalism are detailed in the rest of this section. To facilitate understanding of the formalism, we demonstrate its use by showing how the static (hypothetical) appraisal structure of a baby can be defined. The baby can be exposed to a barking dog or its mother, resulting in different emotions.

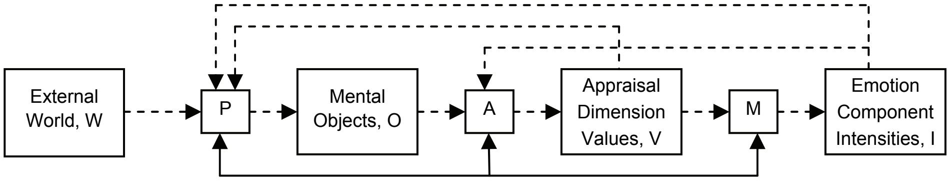
External World, W
P
Mental Objects, O
A
Appraisal Dimension Values, V
M
Emotion Component Intensities, I

Figure 7.2. Graphical overview of the assumed structure of appraisal underlying our formalism. Dotted arrows denote potential inputs for processes, while normal arrows denote potential process dependencies. The external world contains events that can be perceived. Perception processes perceive event, appraisals and emotion-component intensities and map these to mental representations (including beliefs, goals, etc.). Appraisal processes appraise these representations in the context of the current emotion-component intensities, by mapping them to appraisal dimension values (e.g., an object is moderately arousing and moderately goal conducive), which are again mapped to emotion-component intensities by mediating processes (e.g., the current set of appraisals results in a smile and a feeling of excitement). For details see text.

7.3.1 World, Perception Processes and Objects of Appraisal

Definition 7.1.1: $W=\{w_1,\ldots,w_n\}$ is the set of all observable objects and events in the environment of the agent[1].

Definition 7.1.2: $O \subseteq PO$ is the current content of working memory, assuming that $PO=\{po_1,\ldots,po_n\}$ is the set of all potential mental objects with $po_i=(t, any\_object\_name)$, $po_i \in PO$ and $t \in OT$, OT being the set of mental object types as defined in Definition 7.1.3.

Definition 7.1.3: $OT=\{t_1,\ldots,t_n\}$ is the set of type names—(O)bject (T)ypes—used to specify mental object types (e.g. belief, desire, goal, plan, etc.).

Definition 7.1.4: If we define V as the set of appraisal dimension values (see Definition 7.2.2) and I as the set of emotional-response-component intensities (see Definition 7.3.2) then $P=\{p_1,\ldots,p_n\}$ is the set of all perception processes available to the agent, with $p_i:\mathrm{P}(W \cup V \cup I) \rightarrow \mathrm{P}(PO)$, $p_i \in P$ such that $\forall o \in O\ \exists p \in P\ \exists x \in \mathrm{P}(W \cup V \cup I)$ with $o \in p(x)$. In words, a perception process p_i typically maps a portion of the agent's environment, several of the agent's current appraisal dimension values and several of its emotional-response-component intensities to one or more mental objects. These objects are the ones that can be in working memory[2]. Thus, we assume that if an object is in working memory then there must be a perception process producing it.

In our baby example the baby's world initially contains two objects: *mom* and *dog*, represented by two distinct noise levels *m* and *d*, $W=\{m, d\}$. The baby can perceive these objects with her only perception function called hear, p_h, that perceives noise levels *m* and *d*. $P=\{p_h\}$, with $p_h(\{m\})=\{po_m\}$, $p_h(\{d\})=\{po_d\}$, $p_h(\{m, d\})=\{po_m, po_d\}$ and for all other inputs x, $p_h(x)=\varnothing$. Thus p_h maps *m* and *d* to mental objects $PO=\{po_m, po_d\}$. The set OT contains one element, $OT=\{belief\}$, thus $po_m=(belief, mom)$ and $po_d=(belief, dog)$. The baby has two beliefs, *mom* is here and the *dog* is here. The set O is empty; we thus assume that the baby has not perceived anything.

[1] Note that we use *n* as a finite but arbitrary number to denote multiple elements in a set. When *i* is used as element index, we mean for any $1 \leq i \leq n$. Two sets both with *n* elements do not necessarily have the same number of elements. When they do, another subscript is used, e.g., *m*. Also, $\mathrm{P}(S)$ is used to denote the powerset of set *S*.

[2] Note that different perception processes could perceive the same object at the same time, even if they use different information. For example, an agent both smells and sees a person.

7.3.2 Appraisal Processes, Appraisal Dimensions and Values

Definition 7.2.1: $D=\{d_1,\ldots,d_n\}$ is the set of appraisal dimensions, containing elements like suddenness and pleasantness.

Definition 7.2.2: $V=\{v_1,\ldots,v_n\}$ is the set of current appraisal dimension values with $v_i=(o, d, r)$, $v_i \in V$, and $o \in O$, $d \in D$ and $r \in [-1,1]$. In words, v_i is a tuple of a one-dimensional appraisal result attributed to one mental object, or, v_i is the result of appraising an object in terms of one appraisal dimension.

Definition 7.2.3: $A=\{a_1,\ldots,a_n\}$ with $a_i{:}\mathrm{P}(O \cup I) \rightarrow \mathrm{P}(V)$, $a_i \in A$ such that $\forall v \in V\ \exists a \in A$ $\exists x \in \mathrm{P}(O \cup I)$ with $v \in a(x)$. Again in words, a_i is an appraisal process that interprets mental objects in the context of emotional-response-component intensities and attributes appraisal dimension values to other mental objects[3]. Appraisal can be biased by the current emotion, explaining I in the powerset of the input for the appraisal processes. Also, some appraisal processes may be relevant to emotion only through their relation with other appraisal processes. In this case these “indirect” appraisal processes assign only zero values to evaluated mental objects.

To continue our baby example, the baby has two appraisal processes, *pleasure* and *arousal*. Both assign tuples of values [-1,1] and appraisal dimensions to mental objects. There are two appraisal dimensions with almost the same name as the appraisal processes. Thus $A=\{a_p, a_a\}$ and $D=\{pleas, arous\}$. The dog produces noise, so the baby appraises the dog as arousing and unpleasant. So, $a_p(\{po_d\})=\{(po_d, pleas, \text{-}0.5)\}$ and $a_a(\{po_d\})=\{(po_d, arous, 0.5)\}$. For all other inputs x, $a_p(x)=\varnothing$ and $a_a(x)=\varnothing$. The set V currently is empty, as O is empty. Here, we ignore the formal description of the soothing voice of the baby’s mother, as such things tend to defy all attempts at formalization.

7.3.3 Formalizing the Mediating Processes

Definition 7.3.1: $E=\{e_1,\ldots,e_n\}$ is the set of possible components of the emotional response, like certain subjective feelings, facial expressions, physiological reactions and action tendencies.

[3] Note that the mental objects to which an appraisal value is attributed are not necessarily the same as the objects used in the appraisal process. Also note that the introduction of appraisal value *r* introduces a problem if different appraisal processes produce a result on the same appraisal dimension. For example, if two appraisal processes produce the same (*object*, *dimension*, *value*) tuple then only one is in the set *V* (per the definition of sets). However, this could mean that the total intensity of the appraisal dimension is invalid. Since the appraisal value is from the set of real numbers [-1, 1] we assume that this never happens, as it is always possible to pick a real number close enough to another one but different.

Definition 7.3.2: $I=\{i_1,\ldots,i_n\}$ is the set of emotional-response-component intensities with $i_i=(e, r)$, $i_i \in I$, $r \in [-1, 1]$ and $e \in E$ (note that we slightly overload notation here by using subscript *i* with variable *i*). In words, i_i is the intensity of one specific emotional-response component, e.g., a heart rate of 0.5 (on some scale). Appraisal theories typically assume that appraisal dimension values, not emotional-response-component intensities, are attributed to objects. This explains the lack of a mental object in i_i.

Definition 7.3.3: $M=\{m_1,\ldots,m_n\}$ with m_i: $\mathrm{P}(V) \rightarrow \mathrm{P}(I)$, $m_i \in M$ such that $\forall i \in I\ \exists m \in M\ \exists x \in \mathrm{P}(V)$ with $i \in m(x)$. In words, m_i produces emotional-response-component intensities based on appraisal dimension values[4]. Note that the definitions of m_i and a_i follow a common appraisal conception that appraisals are directed at objects, but emotions can be objectless.

Our baby has three emotions: *calm*, *distressed* and *neutral*, $E=\{calm, dis, neut\}$. The baby has one mediating process $M=\{m_e\}$ that relates *V* (the set of assigned appraisal dimension values) to *I* (the set of emotion component intensities) in the following way: $m_e(\{\ (o_d, pleas, -0.5), (o_d, arous, 0.5)\ \}) = \{\ (calm, 0), (dis, 0.5), (neut, 0)\ \}$. For all other inputs *x*, $m_e(x)=\varnothing$. This means that if and only if the baby appraises a situation as arousing and negative, the resulting emotion is distress with intensity 0.5. Again, *I* is empty as *V* is empty; we assume the baby is currently not appraising something.

7.3.4 Dependency between Processes

Our formalism represents processes connected to each other via different kinds of guarded dependencies. To be able to define the notation for dependency relations between processes, we first define guards and dependency types.

Definition 4.1: The set $G=\{g_1,\ldots,g_n\}$ of guards is a set of second-order predicates over the elements of the sets *P*, *O*, *A*, *D*, *V*, *M*, *E* and *I*, and over the variable *r*, being the actual value of elements in the set *V* and the intensity of the emotional response components of the set *I*. This allows the definition of conditional dependencies between processes.

Definition 4.2: The set $LT=\{n_1,\ldots,n_n\}$ is a set of dependency type names—(L)ink (T)ypes—used to identify the nature of the dependency between two processes (e.g., inhibitory, causal, correlation, information flow, parallelism, etc.). Again we slightly overload notation by using n_n.

[4] Same as previous note but for elements in *M*.

Definition 4.3: Let L be the set $L=\{l_1,\ldots,l_n\}$ with $L\subseteq PP\times PP\times G\times N$ and $PP=P\cup A\cup M$. The elements of L define dependencies—(L)inks—between processes constrained by the following. For a tuple (p, q, g, n), with $p, q\in PP$, $g\in G$ and $n\in N$, processing in q is influenced in the way described by n only if the guard g is true and process p is active itself[5].

For our baby, there are four dependencies $L=\{l_1, l_2 , l_3, l_4\}$ between the perception, appraisal and emotion generation processes. These dependencies define a causal activation relation:

- $l_1=(p_h, a_p, (\exists x\ x\in O), \mathit{activation})$,
- $l_2=(p_h, a_a, (\exists x\ x\in O), \mathit{activation})$,
- $l_3=(a_a, m_e, (\exists x\ x\in V \wedge x=(d, i) \wedge i<>0), \mathit{activation}), d\in D$
- $l_4=(a_p, m_e, (\exists x\ x\in V \wedge x=(d, i) \wedge i<>0), \mathit{activation}), d\in D$

These dependencies thus define that if and only if the baby hears something (has perceived an object, i.e., $\exists x\ x\in O$) the appraisal processes must be activated, after which mediating processes are again activated.

7.3.5 Data Constraints

The activation conditions of processes can be defined using the above mentioned dependencies and guards. To allow the specification of data constraints that must hold according to the theory, we define a set H of constraints, again containing second-order predicates. For example, if an appraisal intensity greater than 0.5 for the novelty dimension exists, there must be an emotional-response-component intensity greater than 0 for the orientation response. These constraints also allow formalization of what should happen when there are two appraisal values for the same appraisal dimension, e.g., the baby hears a large and a small dog, both appraised as arousing resulting in two appraisal values loading on the same appraisal variable. Now a data constraint can be used to specify that both values should be, e.g., added. These data constraints are global, and not attached to process dependencies, like the guards used to represent activation conditions.

[5] Note that when a structural theory only mentions the type of the dependencies between processes without mentioning any activation conditions, G can be defined as $G=\{\mathit{true}\}$, so that all dependencies have a guard that is always true and only the type of dependency is used. Second, although we could extend the formal notation by allowing multiple guards or types per dependency, this does not add expressive power to the notation itself since the sets N and G can be filled by an arbitrary number of conjunctions. When actually using the formalism to describe an appraisal theory, multiple guards and types per dependency are definitely allowed to simplify the resulting description of the model.

Definition 7.5.1: The set $H=\{h_1,\ldots,h_n\}$ of guards is a set of second-order predicates over the elements of sets P, O, A, D, V, M, E and I, and over the variable r, being the actual value of elements in the set V and the intensity of the emotional response components of the set I.

7.3.6 What Does the Baby Example Tell Us?

We have formally described the "structural theory" for our baby's hypothetical appraisal structure. For example, if we see the baby crying, we can prove that the baby must be appraising the situation as arousing and unpleasant. We can thus use the formal description to analyze structural relations between emotion processes of our baby. Now imagine a baby (or agent) with a much more complex appraisal structure. If we see it crying while we are trying to make cuddling noises, we might be surprised about this unexpected reaction. However, the formal appraisal structure could be used to, e.g., investigate an alternative possibility: our cuddling noises are appraised as unpleasant and arousing. This would mean, e.g., that the formal model predicts high skin conductance and increase in heartbeat. This is a verifiable hypothesis, and can now be tested. In short, we can use the formal description to evaluate, in a systematic way, whether an emotion is expected or not according to a certain structural theory.

Now, imagine that our theory actually cannot explain why the baby cries (e.g., because skin conductance is predicted to be high but is low in reality), and that a second theory exists that can. We can now formally compare these theories and make explicit the differences between both, so that we are able to explain why the second correctly explains the baby's crying. The sets of processes and dependencies of one theory can be systematically compared with those of another. This is a much more verifiable, understandable and repeatable process than comparing textual representations of structural theories. Comparing theories using our formal notation is the topic of the next section.

7.4 Using Formal Notation to Compare and Integrate Cognitive Appraisal Theories

To show that the formal notation presented above can be used as a tool to compare and integrate different appraisal theories, we present a more serious example than our hypothetical baby. We use our formalism to integrate Scherer's (2001) Stimulus Evaluation Checks (SEC) model and Smith and Kirby's (2000) Appraisal Detector Model (ADM) process model. We call this model the SSK model (*S*cherer, *S*mith and *K*irby). Our goal is to show the utility of formal notations in the domain of emotion theory and the power of our proposed notation in particular. We do not argue that the model we present in this section is the best integration of both theories. For the same reason we have limited ourselves to parts of both theories, the model we present here is not to be interpreted as a complete integration of all aspects of both theories.

7.4.1 Scherer's SEC Model

This model is based on the idea that appraisal processes evaluate stimuli in a certain sequence (for simplicity, in this chapter stimulus and event are assumed to be the same). Five different types of appraisal processes exist related to the evaluation of novelty, pleasantness, goal/need conduciveness, coping potential and norm/self compatibility. These appraisal processes exist at three levels, the sensory-motor level, the schematic level and the conceptual level. Appraisal processes take different forms depending on the level they operate on. An overview of these forms is given in Table 7.1. For the current integration we restrict ourselves by excluding norm/self compatibility.

	Novelty	***Pleasantness***	***Goal/Need conduciveness***	***Coping potential***
Sensory-Motor level (innate)	Sudden, intense stimulation	Innate preferences/ aversions	Basic needs	Available energy
Schematic level (automatic)	Familiarity: schema matching	Learned preferences or aversions	Acquired needs, motives	Body schema (automated knowledge of what the body can do, how it functions, etc.)
Conceptual level (deliberative)	Expectations: cause/effect, probability	Recalled, anticipated, or derived positive-negative estimates	Conscious goals, plans	Problem-solving ability

Table 7.1. Overview of the stimulus checks related to novelty, pleasantness and coping potential existing at the sensory-motor, schematic and conceptual level (Scherer, 2001).

In general, *sensory-motor* level appraisal processes are related to biological needs and drives and to biological mechanisms, and are mostly genetically determined. *Schematic-level* appraisals are based on learned knowledge organized into schemas. *Conceptual level* appraisal processes are based on propositional-symbolic, cortical mechanisms that require consciousness (Scherer, 2001). Higher levels are used to appraise the situation if lower levels seem inadequate to evaluate the stimuli.

As mentioned above, stimulus checks are sequential, and this sequence is roughly based upon the following steps (ignoring, again for simplicity, the last step related to normative significance). We also refer to these steps as *levels of processing*.

- Relevance detection: The stimulus is checked for novelty, innate pleasantness/unpleasantness and goal/need relevance. If it is found to be either novel, or pleasant/unpleasant or relevant to the current needs or goals of the agent, attention is directed to the stimulus (i.e., the orientation response; orienting towards the source of the stimulus) and further processing is initiated.
- Implication assessment: The stimulus is checked for its cause (what caused it), agency of the cause (who did it), its goal conduciveness (is it good for me), its discrepancy between what the agent expected and what actually happened and finally its urgency. This step needs considerable processing resources at the schematic and conceptual level, while the first step is largely operating at the sensory-motor level.
- Coping potential determination: The stimulus is checked to evaluate if the agent is able to control the stimulus or its consequences, and if the agent has enough power to actually effectuate this control (power can have many different sources like physical strength, money, friends, etc.). Finally, if coping potential is limited, the agent evaluates whether it can afford to adjust to the situation. Coping is a process that needs massive processing resources at all three levels.

Although these steps are inherently parallel and evaluated continuously, they are sequential in the sense that the later steps are only deployed to the maximum if earlier checks indicate that this is necessary. Later checks are fully activated only when earlier checks achieve “preliminary closure” (Scherer, 2001), that is, the check has to come to an intermediate stable conclusion about stimuli.

An important aspect of the SEC model is that a SEC is a continuous process that depends on, and changes the results of other SECs (including itself) and that the current state of all SECs is represented in appraisal registers (Scherer, 2001). We call this state the *appraisal state*. This state continuously synthesizes appraisal information from the SECs and is compatible with the concept of appraisal integration proposed by Smith and Kirby (2000). We do formalize the appraisal state, but we do not formalize all recursive connections between the SECs.

A second important aspect of the SEC model is that this appraisal state has a direct effect on all subsystems of the agent. For example, on information processing (the central nervous system), system regulation towards the novel situation (central nervous system, endocrine system and the autonomic nervous system) and action selection (the sensory-somatic nervous system). In the specification of the integration of both models we restrict ourselves to this appraisal state and do not go into the details of the effects of this state on the subsystems, therefore we do not formalize the action tendencies, physiological changes and expressive behaviors that are associated with the different appraisal states.

7.4.2 Smith and Kirby’s ADM

We present a short overview of Smith and Kirby's ADM. In this model the appraisal state (or appraisal integration) is produced by the *appraisal detectors*. The definition of such detectors is the central feature of the ADM (Smith & Kirby, 2000). These detectors continuously integrate the appraisal results originating from three different modes of processing: stimulus perception, associative processing and reasoning. These detectors do not appraise stimuli themselves. Stimulus perception outputs appraisal information to the detectors based on the evaluation of pain, intrinsic pleasure, and other biologically important survival information. In contrast, the latter two modes are considered to be cognitive modes. Associative processing outputs appraisal information based on learned combinations of information and appraisal results. Associative processing is fast, continuous, and autonomous. It can be unconscious and is based on spreading activation paradigms. Associative processing can use any kind of information (e.g., sensations, images, sounds, and emotions). The reasoning mode outputs appraisal information based on deliberative thought processes. These processes are costly and slow, but powerful and able to re-appraise remembered situations and reflect upon the current appraisal state. Reasoning actively generates appraisal information for the appraisal detectors and corresponds best to “active posing and evaluating of appraisal questions” (Smith & Kirby, 2000). Furthermore, the more cognitive the mode, the more resources it needs. It should be clear that these modes are compatible with the levels of appraisal as described in the SEC model. Furthermore, the appraisal integration is responsible for the emotional response, which is also compatible with the appraisal registers in the SEC model.

The ADM explicitly defines a feedback relation between the emotional response and the two different modes of cognitive processing. This feedback relation allows these modes to use emotional information for processing. Associative processing uses this information in learning and remembering, while reasoning uses this information to reflect upon and reappraise the situation.

7.4.3 Summary of Both Models

The ADM assumes three modes of information processing (stimulus perception, associative processing and reasoning). These modes generate appraisal information that is subsequently integrated into a "global" representation of the current appraisal state. This appraisal state is responsible for the emotional response. This state also feeds back to two of the three modes, namely the associative processing and reasoning modes, in order to use this emotional information for learning and reasoning respectively. The SEC model assumes three different levels of appraisal (sensory-motor, associative and conceptual) in which a large amount of different stimulus checks are present. These checks evaluate the stimuli in a specific order and depend on one another. The results of these checks are accumulated into appraisal registers, which—when the results are sufficiently stable—subsequently initiate next appraisal steps and the emotional response.

7.4.4 Formal Integration: the SSK Model

We now present the specification of a potential integration of the Stimulus Evaluation Check model and the Appraisal Detector Model, as an example of how our formalism can be used to integrate appraisal theories. For clarity, the specification is presented in a graphical form (Figure 7.3). To get an idea of the actual set notation, see the boxed text in Section 7.5.2 in which a simplified version of the SSK model is fully specified. This specification is used as basis for the computational model described in Section 7.5.

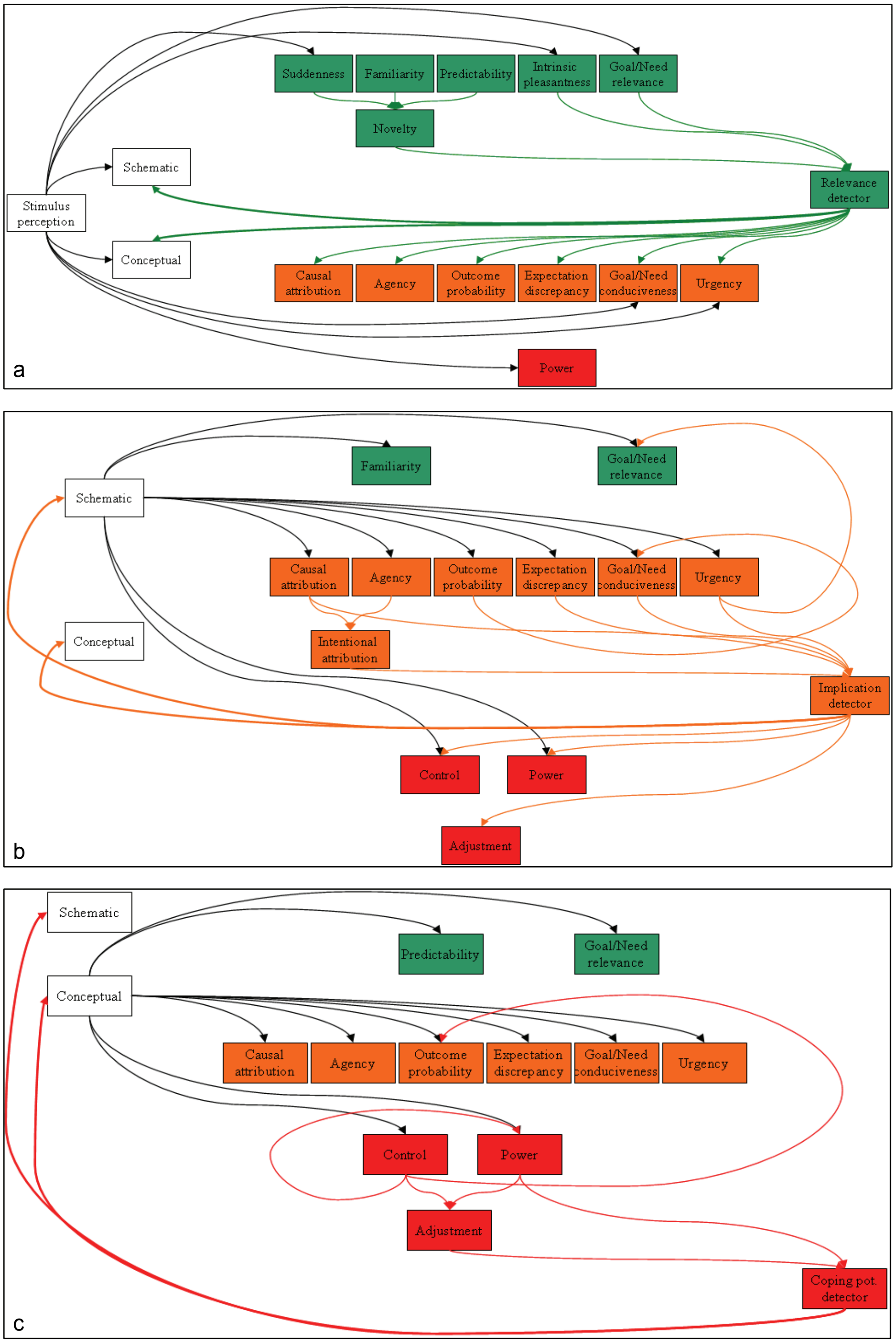
Suddenness
Familiarity
Predictability
Intrinsic pleasantness
Goal/Need relevance
Novelty
Schematic
Relevance detector
Stimulus perception
Conceptual
Causal attribution
Agency
Outcome probability
Expectation discrepancy
Goal/Need conduciveness
Urgency
Power
a
Familiarity
Goal/Need relevance
Schematic
Causal attribution
Agency
Outcome probability
Expectation discrepancy
Goal/Need conduciveness
Urgency
Conceptual
Intentional attribution
Implication detector
Control
Power
Adjustment
b
Schematic
Predictability
Goal/Need relevance
Conceptual
Causal attribution
Agency
Outcome probability
Expectation discrepancy
Goal/Need conduciveness
Urgency
Control
Power
Adjustment
Coping pot. detector
c

Figure 7.3. Graphical representation of the formal SKK model. See main text for explanation. Note that the boxes in the above figure denote processes. Connections between the boxes thus define process dependencies. Appraisal dimensions and emotional-response components are not represented in this figure (appraisal processes and mediating processes are, but in our formalism appraisal dimensions (*D*) and processes responsible for appraising on those dimensions (*A*) are not the same). Lastly, colors correspond to different appraisal steps (green: relevance; orange: implication; red: coping). Figure *a* represents all processes having incoming dependencies related to stimulus perception or outgoing dependencies related to the relevance check. Figure *b* represents the same but now for schematic reasoning or implication, while Figure *c* represents the same but now for conceptual reasoning or coping. Note that some appraisal processes receive input from all three types of processing, and as such are appraisal processes that can function on all three levels of processing (e.g., goal/need relevance).

Before describing the integrated model, some naming issues have to be resolved. When we use the term perception process, we refer to one of the three *processing modes* of the Appraisal Detector Model, to one of the three *levels of appraisal* in the SEC model and to an element $p_i \in P$ in our formal notation. When we use the term *appraisal process*, we refer to a single *stimulus check* in the SEC model and to an element $a_i \in A$ in our formal notation. When we use the term *mediating process* we refer to the appraisal detector/integrator in the Appraisal Detector Model, to the processes that check for *preliminary closure of the temporal appraisal result* in the SEC model and to an element $m_i \in M$ in our formal notation.

We base our integration on two common architectural concepts of the models: (1) the separation of appraisal into three distinct levels of information processing and (2) the appraisal registers/detectors. In our integration we focus on processes (perception, appraisal and mediation) and their dependencies.

We first formalize appraisal dimensions. For clarity, we limit ourselves to the strict minimum of data to be formally specified, in our case the set of appraisal dimensions. To demonstrate the use of dependency guards with second-order conditions relating to these dimensions, we need to include in our formal description at least these appraisal dimensions. The set of appraisal dimensions is defined based on the appraisal registers described in the SEC model, excluding those related to the norm/self compatibility check:

$$D=\{novelty\_dim, intrinsic\_dim, relevance\_dim, conduciveness\_dim,$$

$$urgency\_dim, control\_dim, power\_dim\}$$

We continue with the perception processes. Regarding perception processes, we first define the three processing levels as perception processes, and connect these perception processes to the appraisal processes as defined by the SEC model. This is consistent with both models. The set *P* is represented by the white boxes in Figure 7.3 and equals:

$$P=\{\textit{stimulus perception}, \textit{schematic}, \textit{conceptual}\}$$

Second, the SEC model assumes that certain checks have input from different levels of processing. For example *Goal/Need relevance*, *Urgency* and *Power* use input from all three levels of processing. The ADM specifically assumes that appraisal information can come from different levels. Although the models do not exactly define how the appraisal processes are distributed over the three levels of processing, together they give enough guidelines to formalize the connections between perception and appraisal. These connections are shown by the black arrows in the graphical representation of the specification. These connections define *excitatory* dependencies between the perception processes and appraisal processes. This connection topology thus defines the dependencies between modes of processing / levels of appraisal on the one hand and appraisal processes on the other. Additionally two *excitatory* dependencies are defined between the perception processes: one dependency between *stimulus perception* and *schematic*, the other between *stimulus perception* and *conceptual*. This reflects the general information processing architecture of the Appraisal Detector Model, which prescribes that perceived stimuli are processed further by the associative and reasoning mode. We do not define guard conditions for these dependencies, although several exemplary guards based on the SEC model are shown in Section 7.5.2 (boxed text).

An important characteristic of both models is that appraisal processes can evaluate continuously. In our model, continuous evaluation can be initiated by the perception processes, and is independent of the previous appraisal check. This aspect is represented by the dependencies between the perception processes and the appraisal processes in the three appraisal checks. Perception processes thus influence processing of appraisal processes directly, but only according to the structural relations defined in the SEC model.

Now we formalize the appraisal processes. The colored boxes represent appraisal processes (excluding the rightmost three boxes, to which we will return shortly). The green boxes represent those appraisal processes that are part of the first step of the stimulus checking process as defined by the SEC model. The yellow boxes represent the second step and the red boxes the third step (recall that we did not include the fourth, norm/self related step in our formal integration). The set of appraisal processes is thus defined as follows:

$$A=\{\textit{elements of the set of stimulus checks in the first 3 steps of the SEC model}\} \cup \{\textit{agency}, \textit{suddenness}, \textit{familiarity}, \textit{predictability}\}$$

We have included the appraisal process *agency*, because the SEC model, when determining whether the cause of an event is due to the action of an agent, implicitly assumes the existence of this process. Also, we included *suddenness*, *familiarity* and *predictability*, the three sub checks responsible for the result of the *novelty* check. We have explicitly included these sub checks as separate appraisal processes because in the SEC model each of them operates on a different level of appraisal. Therefore, these processes need to be formally connected to different perception processes.

Connections originating from appraisal processes define *excitatory* dependencies. The topology of these connections defines the structural dependencies between appraisal processes, consistent with the SEC model. For clarity, the color of a connection represents the appraisal step to which the dependency's originating appraisal process belongs. For instance, the green connection from *suddenness* to *novelty* represents an excitatory dependency originating from an appraisal process in the first appraisal step.

We continue with formalizing mediating processes. The three rightmost colored boxes represent mediating processes. The set *M* contains the following elements:

$$M=\{\textit{relevance detector}, \textit{implication detector}, \textit{coping potential detector}\}$$

These mediating processes are positioned between the different levels of appraisal. Mediating processes are activated by the appraisal processes of one level and activate appraisal processes of the next level, through *excitatory* dependencies. This connectivity explains their role: mediating processes detect when appraisal information is such that the next appraisal step should be activated in full glory. For example, if the *novelty* appraisal process outputs appraisal information that characterizes high novelty, the *relevance detector* will activate the appraisal processes to which its connections point.

Remember that all connections can be guarded, although for clarity we did not define most of the guards. In principle this allows connections to activate based on evaluation of second-order logic conditions. For example, we could define the following guard for the dependency between *novelty* and *relevance detector*:

$$(\exists x\, x \in V \wedge x=(o, d, i) \wedge i>t \wedge d=novelty\_dim),$$

with $novelty\_dim \in D$ and $t \in [0, 1]$ an arbitrary threshold. This guard checks the existence of a *novelty_dim* value greater than an arbitrary threshold *t*. Only if this value exists, the guard will be true, and thus the connection is active. Now the *novelty* appraisal process excites the *relevance detector*.

Finally we formalize process feedback. To formally represent the influence of mediating processes on processing modes, we have defined dependencies originating from the mediating processes ending at the *schematic* and *conceptual* perception processes. The influences are represented by six thick connections between the mediating processes and the perception processes. In the ADM, the emotional response feeds back to the associative and reasoning modes. The mediating processes in our formalism generate emotional response component intensities (elements in the set *I*). These component intensities formally represent the emotional response, and are available to all perception processes. Since the ADM defines this relation as data flow, perception processes are not activated through an *excitatory* dependency. We have defined a different type for these dependencies, called *information_available*. This means that when the guard of the dependency is true, the target process is informed of the fact that new information is available.

7.4.3 Summary

Integration and comparison are important reasons to formalize appraisal theories (Wehrle & Scherer, 2001). Therefore, a formalism for structural models should facilitate integration and comparison. In this section we have shown how our formalism can be used to integrate theories of appraisal. We have based our integration on two common architectural concepts of the models: (1) the separation of appraisal into three distinct levels of information processing and (2) the appraisal registers/detectors. We believe the integration was greatly facilitated by the formalism's ability to describe in detail the processes, their conditional dependencies based on second-order predicates and the appraisal dimensions.

7.5 Using Formal Notation to Develop and Evaluate a Computational Model of Emotion

To show the power of our formalism as basis for computational models of emotion, we describe a computational emotional agent that has been based on a simplified version of the SSK model. We have emotionally instrumented an existing version of the arcade game *PacMan*. This version was downloaded from the internet (Chow, 2003). We assume that the reader is familiar with the game of *PacMan*. First, we present the specification that was used as basis for the appraisal mechanisms implemented in *PacMan*. Then we show how this specification can be used to fill in missing declarative information that is critical to the development of a computational model. Finally, we show how our formal model helped us to debug our emotional *PacMan*-agent.

7.5.1 Why *PacMan*?

PacMan-like environments have been used in emotion research, both in the appraisal-theoretic domain (Wehrle & Scherer, 2001) as well as the virtual agent domain (Broekens & DeGroot, 2004a). Apart from being useful in the domain of emotion research (Wherle & Scherer, 2001), *PacMan* (Figure 7.5) is also a suitable environment to test emotional instrumentation for several reasons. First, *PacMan* provides a simple environment that allows for meaningful emotional instrumentation related to different levels of appraisal. This allows us to start with appraisal processes related to sensory-motor perception only (e.g., eating dots, being eaten by ghosts) and then extend this to appraisal processes related to the schematic level (e.g., eating fruit and ghosts related to the goal of collecting points). Second, *PacMan* is an environment enabling broad emotional coverage. Many different emotions make sense. Eating ghosts, eating dots, losing a life, being chased, chasing, etc. are all different situations imbuing different emotions in humans. Third, *PacMan* is an “action-packed” environment, which allows us to test the computational model’s appraisal behavior under continuous-time constraints. This facilitates studying the process of appraisal.

7.5.2 Generating a Formal Description for the Computational Model.

Before we introduce our formal description of *PacMan*’s appraisal structure we have to stress again that the point we want to make is that formal specifications of structural models are important for the development of computational models of emotion. More specific, the formal notation presented in this chapter is a powerful one. Consequently, the goal of this experiment was not to design a believable or “full-blown” emotional agent.

We have used a simplified version of the SSK model as basis for our computational emotional agent. First, we ignore the *conceptual* perception process since our *PacMan* agent is incapable of high-level cognitive processing. Second, several appraisal processes in the SSK model are ignored, because (1) these made no sense in light of the simplicity of the *PacMan* environment, or (2) because we could not design simple appraisal processes directly related to those mentioned in the formalism without providing the underlying mechanisms in more detail. Omitted processes are: *adjustment*, *expectation discrepancy*, *outcome probability check*, *predictability* and *attribution*. Third, since our *PacMan* agent is unable to use its emotions in any way, the feedback from the *mediating* processes to the *perception* processes is ignored. Note that our formal description of the SSK model enabled us to quickly evaluate what processes could or should be ignored in PacMan's case. This task would have been much more difficult without such description. The resulting processes and their dependencies are depicted in Figure 7.4.

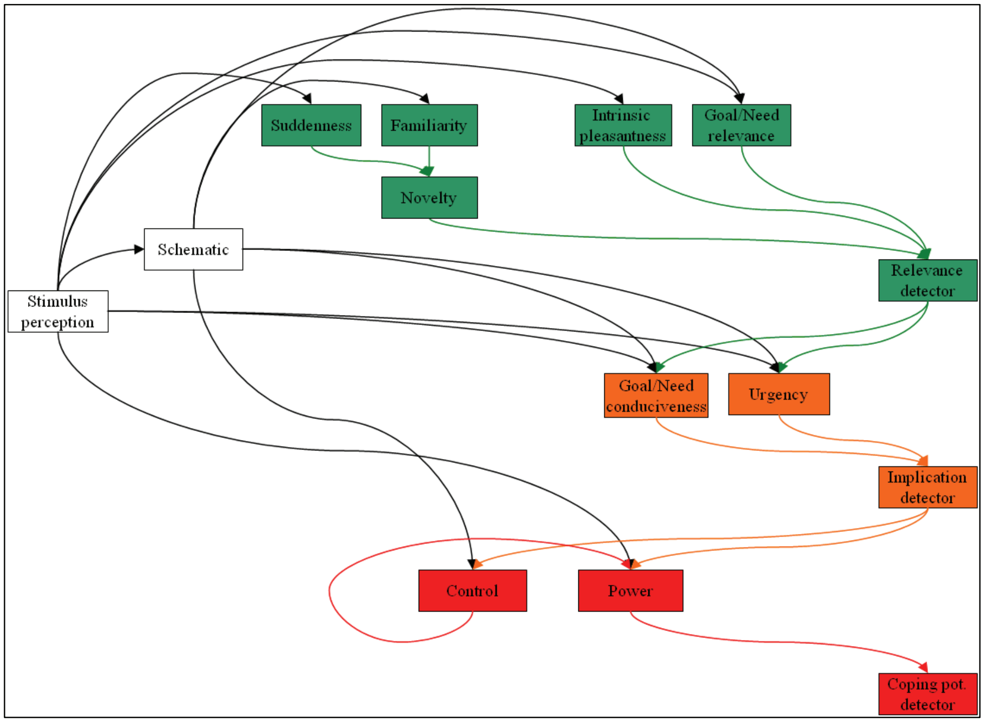

Figure 7.4. Graphical representation of the specification of PacMan's appraisal structure.

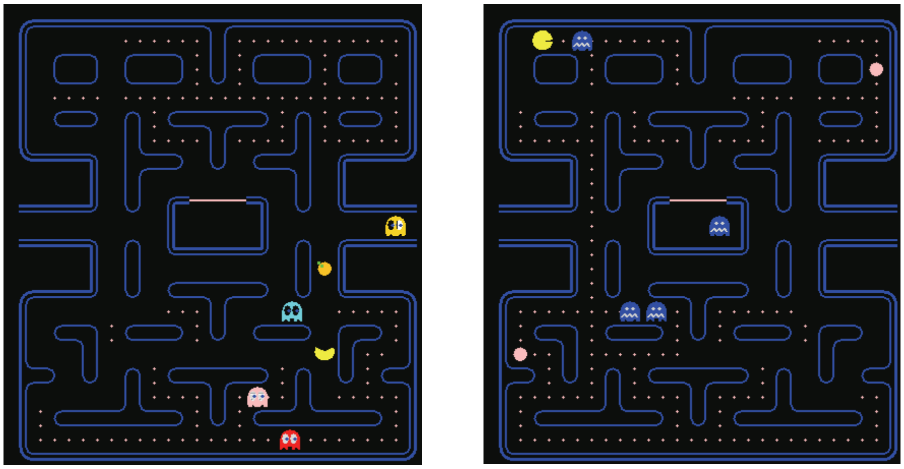

Figure 7.5. PacMan screen shots: chasing fruit (left), chasing an edible ghost (right).

The formal set notation of the simplified SSK model applied to *PacMan* is defined as follows. Perception, appraisal and mediating processes (just the processes, not the formal description of their input-output relations):

P = { stimulus_perception, schematic }
A = { suddenness, familiarity, novelty, intrinsic_pleasantness, relevance, conduciveness, urgency, control, power }
M = { relevance_detector,implication_detector, coping_potential_detector }

Mental object types, mental objects, appraisal dimensions and emotion components:

OT= { belief }
PO= {(see_ghost, belief), (lost_ghost, belief), (eaten_by_ghost, belief), (see_edible_ghost, belief), (lost_edible_ghost, belief), (eaten_ghost, belief), (see_power, belief), (eaten_power, belief), (see_dot, belief), (eaten_dot, belief), (see_fruit, belief), (lost_fruit, belief), (eaten_fruit, belief) }
D = { novelty_dim, intrinsic_pleasantness_dim, conduciveness_dim, relevance_dim, urgency_dim, control_dim, power_dim }
E = { }

Link types, guards, data constraints and dependencies:

LT= { activation }
G = { true, guard$_1$, guard$_2$, guard$_3$ } with:
guard$_1$ = ($\exists v_1,v_2,v_3\ v_1,v_2,v_3 \in V \wedge v_1=(o,d_1,i_1) \wedge v_2=(o,d_2,i_2) \wedge v_3=(o,d_3,i_3) \wedge (|i_1|+|i_2|+|i_3|)/3>0.15 \wedge d_1=novelty\_dim \wedge d_2=intrinsic\_dim \wedge d_3=relevance\_dim)$
guard$_2$ = ($\exists v_1,v_2\ v_1,v_2 \in V \wedge v_1=(o,d_1,i_1) \wedge v_2=(o,d_2,i_2) \wedge (|i_1|+|i_2|)/2>0.25 \wedge d_1=conduciveness\_dim \wedge d_2=urgency\_dim)$
guard$_3$ = ($\exists v_1,v_2\ v_1,v_2 \in V \wedge v_1=(o,d_1,i_1) \wedge v_2=(o,d_2,i_2) \wedge i_1 * i_2>0 \wedge d_x=control\_dim \wedge d_y=power\_dim)$
H = { c$_1$, c$_2$} with:
c$_1$ = $((\exists x)x \in V \wedge x=(y,d,i,t) \wedge i>0)$ if $((\exists y)y \in O \wedge y=(c,j,t') \wedge j>0 \wedge t=t')$, and
c$_2$ = $((\exists z)z \in I \wedge z=(e,i',t'') \wedge i'>0)$ if $((\exists x')x' \in V\ x'=(y',d,j',t''') \wedge j'>0 \wedge t''=t''')$}
L={ (stimulus_perception, suddenness, true, activation),
(stimulus_perception, intrinsic_pleasantness, true, activation),
(stimulus_perception, relevance, true, activation),
(stimulus_perception, conduciveness, true, activation),
(stimulus_perception, urgency, true, activation),
(stimulus_perception, power, true, activation),
(schematic, familiarity, true, activation),
(schematic, relevance, true, activation),
(schematic, conduciveness, true, activation),
(schematic, urgency, true, activation),
(schematic, control, true, activation),
(suddenness, novelty, true, activation),
(familiarity, novelty, true, activation),
(novelty, relevance__detector, true, activation),
(intrinsic_pleasantness, relevance_detector, true, activation),
(relevance, relevance_detector, true, activation),
(relevance_detector, conduciveness, guard$_1$, activation),
(relevance_detector, urgency, guard$_1$, activation),
(conduciveness, implication_detector, true, activation),
(urgency, implication_detector, true, activation),
(implication_detector, control, guard$_2$, activation),
(implication_detector, power, guard$_2$, activation),
(control, coping_potential_detector, guard$_3$, activation),
(power, coping_potential_detector, guard$_3$, activation) }

To construct a computational model that can execute, we have to fill in missing declarative information. We need to address several issues mentioned earlier in this chapter, issues that relate to computational aspects like process activation thresholds, process activity, and input/output constraints. Many of these questions are answered neither in the SEC model nor in the ADM. Consequently, answers are not available in the specification of the integration of both models. This is not intended as critique, but as an observation about the immediate applicability of appraisal theories as basis for computational models. This applicability is limited, as already mentioned by Gratch and Marsella (2004). Our observation lends formal support to this. We now describe how we added guards to fill in the missing details in a formal way.

First, two appraisal processes, *suddenness* and *familiarity* influence the appraisal dimension *novelty_dim*. How does the *novelty* process integrate this information? In Scherer's SEC model (Scherer, 2001), references are made to the mechanisms that could be responsible for *suddenness* and *familiarity*, but this information is not detailed enough for a computational implementation of the integration of the results of these mechanisms. To stay consistent with the SEC mode, we assume that both *suddenness* and *familiarity* appraise mental objects in terms of the *novelty_dim* dimension. Whenever one of these processes is active, the *novelty* check is activated and integrates these two results into one value by adding-up. Dependencies between *suddenness* and *familiarity* on the one side and *novelty* on the other are therefore without guard.

Second, what are the thresholds for the activation of the *relevance* and *implication detectors*? Or even more fundamentally, can we speak of a threshold? According to the SEC model, we can, since this model specifically mentions *preliminary closure*. However, no threshold or guideline for a threshold mechanism is given that is useful for an algorithmic approach (apart from the appraisal register values being *relatively stable*, which is about the same as *preliminary closure*).

Since we do not have a numerical guideline, we assume the following: the *relevance detector* is activated by either one of the three appraisal processes: *novelty*, *intrinsic pleasantness* and *need/goal relevance*. Every outgoing dependency from the *relevance detector* to an appraisal process of the next appraisal step has a guard equal to:

$$(\exists v_1, v_2, v_3 \text{ with } v_1, v_2, v_3 \in V \wedge v_1=(o, d_1, i_1) \wedge v_2=(o, d_2, i_2) \wedge v_3=(o, d_3, i_3) \wedge$$

$$(|i_1|+|i_2|+|i_3|)/3>0.15 \wedge d_1=\mathit{novelty\_dim} \wedge d_2=\mathit{intrinsic\_dim} \wedge d_3=\mathit{relevance\_dim})$$

We assume all three tuples v_1, v_2 and v_3 to exist. If not, we take their corresponding activation value to be equal to 0. Thus, this guard checks the value of the cumulative activation of the appraisal dimensions that are relevant to the *relevance* check. The value must be greater than an arbitrarily chosen threshold.

The next guard is related to the *implication detector*. The Goal/Need *conduciveness* and *urgency* processes activate this *implication detector*. Every outgoing dependency from the *implication detector* to an appraisal process of the next level of appraisal has a guard equal to:

$$(\exists v_1, v_2 \text{ with } v_1, v_2 \in V \wedge v_1=(o, d_1, i_1) \wedge v_2=(o, d_2, i_2) \wedge (|i_1|+|i_2|)/2>0.25 \wedge$$

$$d_1=\mathit{conduciveness\_dim} \wedge d_2=\mathit{urgency\_dim})$$

Again, we assume that the tuples v_1 and v_2 exist, and if they do not, we take their corresponding activation value to be equal to 0. Thus, this guard checks the value of the cumulative activation of the appraisal dimensions that are relevant to the *implication* check.

A third missing detail is the exact relation between control and power. Also, how do these appraisal processes together influence the *coping-potential detector*? Only a descriptive guideline is given in the SEC model, stating that the evaluation of power only makes sense if the situation is controllable. Complete lack of control or complete lack of power both result in lack of coping potential. High control results in coping potential fully dependent on power. Assuming that both dimensions cannot attain negative values, this can be interpreted as a multiplication of the appraisal dimension values for *power_dim* and *control_dim*. *Coping potential* is activated when the product between *power_dim* and *control_dim* is above a certain threshold. We defined the following guard attached to the dependency between the power appraisal process and *coping-potential detector*:

$$(\exists v_1, v_2 \text{ with } v_1, v_2 \in V \wedge v_1=(o, d_1, i_1) \wedge v_2=(o, d_2, i_2) \wedge i_1 \times i_2>0 \wedge$$

$$d_1=\mathit{control\_dim} \wedge d_2=\mathit{power\_dim})$$

Again we assume that both tuples v_1 and v_2 exists, and if one of them (or both) do not, we take their corresponding value to be equal to 0.

Fourth, what is, in the context of *PacMan*, a sensory-motor perception process and what is a schematic perception process? According to the Appraisal Detector Model the sensory-motor mode of processing reacts to inherently pleasant and painful stimuli or facial expressions and the SEC model states that this level of appraisal relates to stimuli having to do with basic needs, available energy and direct sensory processing—like sudden movements. Both models give a clear guideline, and we think that it is feasible to use this guideline in our domain. We have done this in the following way. The sensory-motor perception process reacts to events related to the survival of the *PacMan* agent. One can think of eating dots (*PacMan* is assumed to live of dots), being eaten by a ghost and perceiving dots and ghosts (see Table 7.2). The schematic perception process reacts to events that relate to the goal of collecting points (Table 7.3).

Appraisal process	*Dimension*	*Checking criteria*
Suddenness	`novelty_dim`	Moving objects (ghosts and fruit) are evaluated equally positive and more novel than non-moving objects (pills and dots).
Intrinsic pleasantness	`intrinsic_dim`	Eating a dot is positive, while being eaten by a ghost is negative.
Need relevance (survival)	`relevance_dim`	Events related to dots and non-edible ghosts respectively have values relative to the amount of hunger *PacMan* has and the amount of lives left (hunger is simulated based on the last time *PacMan* ate a dot).
Need conduciveness	`conduciveness_dim`	Based on all events related to non-edible ghosts and dots.
Urgency	`urgency_dim`	Based on whether the event implies a moving object. Seeing a non-edible ghost is urgent.
Power	`power_dim`	The power-pill time left is an indication of the amount of power left.

Table 7.2. *PacMan* appraisal related to survival need

Appraisal process	*Dimension*	*Checking criteria*
Familiarity	`novelty_dim`	Seeing a dot is more common than seeing a ghost, and seeing a ghost is more common than a power-pill which again is more common than fruit.
Goal relevance (points)	`relevance_dim`	All events related to fruit and eating ghosts are equally relevant.
Goal conduciveness (points)	`conduciveness_dim`	Seeing an edible ghost, eating a ghost, seeing and eating fruit are positive, while losing an edible ghost and losing a fruit are negative.
Urgency	`urgency_dim`	Based on whether the event implies a moving object. Seeing an edible ghost and a fruit both are equally urgent.
Control	`control_dim`	Based on whether the event allows to be controlled. All moving objects allow control to a certain degree, but fruit and edible ghosts allow for more control than non-edible ghosts. Seeing a power-pill also implies control.
Power	`power_dim`	Power is completely determined based on the power-pill time that is left.

Table 7.3. *PacMan* appraisal related to the goal of gathering points

7.5.3 Verification of the Computational Model

We have instrumented *PacMan* by building a simple system that generates mental objects based on the current game situation. The decision support system is based on the SSK model and has two processes, the *sensory-motor* perception process and the *schematic* perception process. Mental objects are appraised based on the appraisal processes and their relations as described in the SSK specification. These appraisal processes produce appraisal dimension values, as specified in Tables 7.2 and 7.3. These values are continuously integrated and the result is maintained in an appraisal state that is modeled as a vector with cardinality equal to the number of different appraisal dimensions (7 in our case, see boxed text in Section 7.5.2). This integration simply consists of adding appraisal values that belong to the same appraisal dimension and storing the result in the appraisal state.

The experiment itself consists of a human player controlling the instrumented *PacMan* agent who plays the first level of the *PacMan* game (by eating all dots), loses a life two times during the game, and eats several ghosts. When we ran the experiment, the result was contradictory. Although certain situations obviously should have a strong implication to the *PacMan* agent, the stimulus checks of the coping appraisal step were not activated, but should have according to the formal description. This lack of activation can be seen in Figure 7.6a, for $9000 \le t \le 13000$. In these situations *PacMan* was seeing a ghost and seeing and eating dots. However, the *implication* in this situation is below the arbitrarily defined threshold of 0.25, while other clearly less important situations are above this threshold (e.g., around *t*=27000 where *PacMan* only sees a ghost).

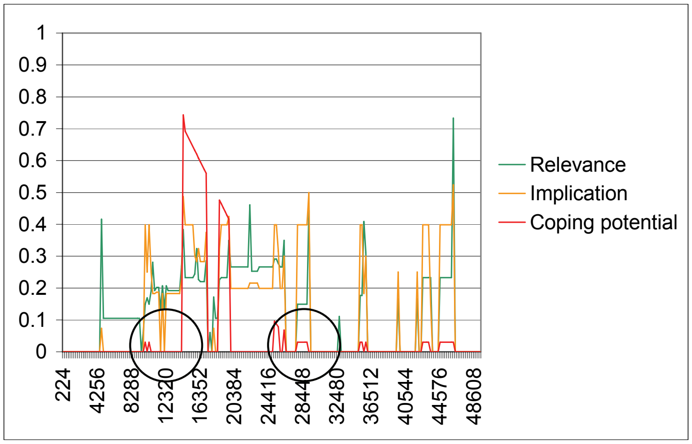

Figure 7.6a. PacMan using bi-polar variables. Time in milliseconds is on the x-axis. Appraisal dimension activation is on the y-axis. *Coping potential* is not activated around *t*=10000. The *implication detector* stays below its threshold.

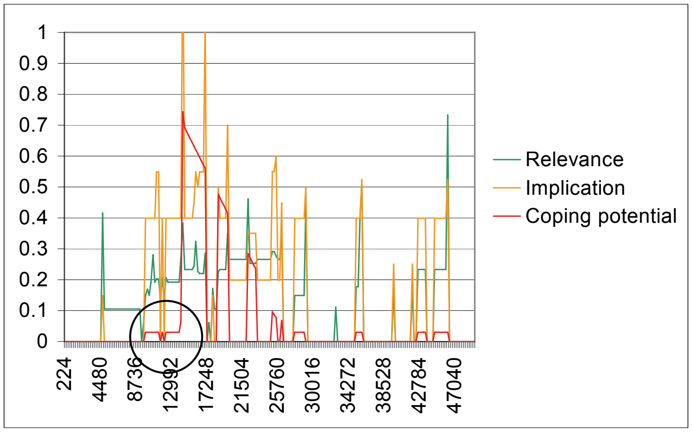

Figure 7.6b. PacMan without bi-polar variables. *Coping potential* is activated around *t*=10000, as a result of higher activation of the *implication detector.*

We can explain these contradictory results by examining the formal specification. The appraisal process *conduciveness* can produce both positive and negative appraisal values for the appraisal dimension *conduciveness_dim*. When these values are integrated by the *implication* mediating process, they cancel each other instead of together contributing to a high implication situation. Subsequently the guard of the *implication* mediating process is not true, so the next appraisal step (coping) is not activated, resulting in the contradictory result.

The underlying reason for this is that the above mentioned appraisal dimension is bi-polar (i.e., can have negative and positive values) and thus switches meaning when it switches sign. Consequently there is only a small difference between, for example, a situation in which highly conducive and non-conducive events happen and a situation in which nothing happens at all. In other words, this dimension cannot represent “mixed-emotions”. Because of the formal structural specification of the SSK model, we were able to exactly identify this issue. After the introduction of an extra appraisal dimension, an extra appraisal process that checks stimuli related to *non-conduciveness*, and a link between that process and *implication*, the new results are as expected. *Coping potential* is activated but low since *PacMan* has not eaten a power-pill recently (Figure 7.6b, between $9000 \leq t \leq 13000$).

7.5.4 Summary

Our formalism helped to develop a computational model based on the SSK model. It facilitated (1) filling in of computational details, and (2) making computational assumptions explicit. Further, the formal description helped us to verify and validate our computational model with respect to the SSK model. We could identify what was in our case a problem using bi-polar appraisal dimensions. Note that we do not claim anything about bi-polar appraisal dimensions per se. We claim that our formalism is useful for the specification and verification of a computational model.

7.6 Discussion

We first discuss several formalization issues. Then we discuss related and future work.

7.6.1 Some Drawbacks of Formalization.

Two warning remarks regarding formalization have to be made. First, the focus on strict definitions can be a disadvantage of formalization when used as a tool for psychological theory refinement. Formal modeling forces a theorist to commit to certain definitions for the concepts in a theory. In and of itself such commitment can be an advantage because it helps to refine and clarify theories (Mallery, 1988). However, such commitment can also be a disadvantage when unclear bounds of the concept to be formalized result in either a too strictly formalized concept—producing a formal representation that does not cover all of the concept—, or a too loosely formalized concept—producing a formal representation that is not better than the non-formal representation. It could be argued that this is not a disadvantage of formalization, but a lack of specificity of the theory. The theory lacks clear definitions. However, appraisal theories—like many theories of psychological processes—generally include concepts with such open bounds for good reasons.

A second, more important, disadvantage is that formal specifications risk living their own lives. This is all right if the probability is high that a formal specification covers everything the theory describes. As discussed above, exactly *this* is far from certain. However, as formal notations have many benefits (clarity, preciseness, etc.) the formal description of a cognitive appraisal theory might (by some) be interpreted as a substitute for the actual theory. This could result in overly strict interpretations of that theory, eventually leading to wrongly rejecting a phenomenon as consistent with the theory, based on results from an experiment with a computational model that is based on a formal specification. Rejecting a phenomenon based on a formal description of a psychological theory should thus always be done with care. The inverse, the acceptance of a phenomenon as supporting a theory, is less problematic since the formal specification of the theory generally is stricter than the theory itself.

7.6.2 Related Work.

We briefly discuss four approaches to the formalization of emotion theory. The choice for these four examples is not arbitrary; they each represent a different way in which formalization can be used in this context.

First, Gmytrasiewicz and Lisetti (2002) have defined a formalism to describe how emotions can influence agent decision making. Their formalism defines emotions as different modes of decision making. Their formalism allows the definition of personalities of others, where a personality can be seen as the potential transitions between emotional states. This approach is different and in a way complementary to ours. While their approach takes the emotion as a given and formalizes the influence this emotion has on decision making, our approach formalizes the structure of appraisal in order to, for example, describe the interactions between perception, appraisal and emotion mediating processes that generate the emotion in the first place.

Second, Meyer (2004) proposes a formalism based on modal logic to formally describe how specific emotions relate to the belief, desire and intention structure of an agent. This approach differs from ours in the sense that it tries to formalize an emotion in terms of specific sets of beliefs, desires and intentions, while our approach tries to formalize the appraisal theory on which the computational model is based by describing the processes and their structural relations.

Third, the GATE environment is a black-box modeling environment aimed at theory comparison (Wehrle & Scherer, 2001). This tool allows researchers to specify the theoretical relation between appraisal dimension intensities and emotional-response components—using mathematical formulas and parameters—and quickly compare the results of experiments with the theoretical predictions. A large database is attached to the tool, in which experimental results are stored. The database can be filled automatically with the results of questionnaires that are filled in by subjects. Data from this database can be used to compare experimental data with theoretical predictions derived from various theories. GATE contains a large set of analysis functionality to facilitate this comparison. The main differences between GATE and our approach are our theory independent, set-based formalism and our focus on the specification and verification of computational models. Our formalism allows the definition of the declarative semantics of the different processes, their inputs, outputs and interactions. If time is introduced (see future work) in our formalism it enables specification of the relation between the sub-processes involved in appraisal and specification of evolution of the structure of appraisal during development of an agent. Since we use a set-based notation, a formal specification developed with it can be systematically and automatically evaluated for consistency with a computational model or appraisal theory.

Fourth, Reisenzein (2000) proposes a meta-level formal representations for the emotion theory of Wundt. His approach is very similar to ours, in that it attempts to formalize the emotion theory at a structural level using a set-theoretic notation. Important differences are that his approach is more systematically based upon the structuralist approach (Westmeyer, 1989), and that our formal notation has explicitly been developed to also facilitate development of computational models. However, a closer comparison of both approaches is needed in the future. This is specifically interesting as the structuralist approach towards formalization is by no means restricted to the formalization of cognitive theories. This would indicate that our approach could be extended to less cognitively-oriented theories of emotion.

7.6.3 Future Work.

Our current version of the formal notation describes the static structure of appraisal. Future work should include time. Time is needed in order to model the evolution of a structural model. For example, we might want to formalize the relation between different developmental stages from child to parent (Lewis, 2001), or formalize the evolution of an appraisal over a shorter time period.

Further, to formalize the difference between conscious and unconscious influences (Zajonc, 2000), we need to separate the mental objects, our set O, in subsets of objects. Every subset now contains objects with different activation strength. This strength represents whether an object is conscious or not.

Also, future work includes the addition of long term memory to our formalism. It is difficult to formalize reappraisal (Levine, Prohaska, Burgess, Rice & Laulhere, 2001) or coping (Lazarus, 2001), without the LTM construct.

Finally, a comparison between the structuralist approach towards theory formalization and our approach is planned.

7.7 Conclusion

Integration of appraisal theories is important for the advancement of appraisal theory (Wehrle and Scherer, 2001). We have proposed a formal notation for the declarative semantics of the structure of appraisal, and argued for the need to have such a formalism. We have shown that this formalism facilitates integration between appraisal theories. We have illustrated this by integrating (in a simplified way) two appraisal theories; the Stimulus Evaluation Check model by Scherer (2001), and the Appraisal Detector Model by Smith and Kirby (2000) into one model, the "SSK model" (Section 7.4). The process of integration was greatly facilitated by the ability provided by the formalism to specify in detail the perception, appraisal and mediating processes, their conditional dependencies based on second-order logic and the appraisal dimensions.

We have shown that our formalism is a first step to narrow the gap between structural models of appraisal and computational models. To this end we have used our formalism as intermediate specification of structure and completed the translation process from appraisal theory to computational model by developing a computational model of emotion based on the "SSK model". We have shown that our formalism helped development in the following way (Section 7.5): filling in of computational details, and making computational assumptions explicit was greatly facilitated by the formal description of the "SSK model". Moreover, it helped us to verify and validate our computational model with respect to the "SSK model".

To summarize, our formalism for the structure of appraisal can be used to further advance cognitive appraisal theory as well as to facilitate development and evaluation of computational models of emotion based on cognitive appraisal theory.

8

Summary and Conclusion

Here we give a concise overview of the results presented in the different chapters of this thesis, and relate these to each other.

8.1 Affect, Mood and Information Processing

Action selection has been defined as the problem of continuously deciding what next action to select in order to optimize survival (Tyrell, 1993). In a Reinforcement Learning (RL) context, action selection is the process of selecting the next action from a set of actions proposed by a model of interaction with the world, such that the model can both be *learned* by means of interaction and be *used to optimize* received reward. In our case the RL mechanism is a model-based Reinforcement Learning method (see Kaelbling et al., 1996). An agent can select actions in a variety of ways, such as greedy (take the best proposed action) or random (take any action). This is an important issue in robot learning: when should the action-selection mechanism explore versus exploit. We have shown (Chapter 3) that this action-selection trade-off can be partly controlled by artificial affect, when artificial affect is defined as a measure that keeps track of how well the agent is performing compared to what the agent is used to. If the agent performs well, artificial affect is positive and action selection can be greedy, reflecting the relation "good performance—keep doing what you do" (exploit). If the agent performs badly, artificial affect is negative and action selection must be more random, reflecting the relation "bad performance—try new stuff" (explore). By doing so, we have shown that computational modeling can give insights into the possible relations between affect and learning on a meta-level. Artificial affect can be used to control learning parameters.

The type of agents used in the experiments just mentioned is reactive. Agents do not have an abstraction of thought. They behave using an input-output mapping: states go in; actions come out (via the value function that learns value-state-action mappings and the action-selection mechanism that subsequently selects one action from the set of proposed actions). However, thoughts, like actions, have to be selected in some way too. We have shown that thought-selection can also be controlled up to a certain extent by artificial affect (Chapter 4). Thought in this case is interpreted as internal simulation of behavior (Hesslow, 2002). The agent can internally simulate potential interactions with the world. In our experiments, simulation is bounded to one imaginary step ahead, however multiple possibilities exist for that one step (different actions are possible, and different states could result from those actions). The agent has to choose between selecting only several good thoughts (the agent is in a good mood and thinks greedy) or a lot of diverse thoughts including thoughts that are evaluated as bad (bad mood, the agent thinks "explorative"). Again, there is a trade-off between exploration (internally simulate all potential next interactions) and exploitation (internally simulate the option that is perceived as best). Note that in our computational studies, the agent can not really explore mentally, as it can only think of things it has already encountered, explaining the quotes around exploration.

We have shown that internal simulation in this sense is beneficial to the learning performance of an agent, and that artificial affect can be used to control thought selection. Internal simulation of all possible options results, in all experiments, in the best learning performance when compared to no simulation, some simulation, or affectively controlled simulation. However, internal simulation of all options every step is a waste of effort. Sometimes, the learned world model and value function are very good; simulation is actually not needed and the agent can just use a purely reactive mode of operation. In other cases, the learned model is bad; the agent should try to look ahead in a broad sense in order to predict possible consequences of its actions. Artificial affect can control this trade-off. When positive artificial affect is coupled to less, but greedy, internal simulation and negative artificial affect is couple to more, "explorative", internal simulation, the resulting amount of internal simulation that is needed for a learning performance comparable to one resulting from simulation of all options every step is reduced. To be more precise, coupling artificial affect to internal simulation, in the way just mentioned, enables a learning agent to have about the same learning performance gain compared to an agent that simulates all possible interactions every step, but using considerably less internal simulation. This means that agents that "feel good" can think ahead in the narrow sense freeing mental resources for other things, while agents that "feel bad" should think ahead in a broad sense fully using mental resources to plan ahead. This is compatible with the psychological literature on human mood, as discussed in Chapter 3 and 4.

An interesting issue that has not been discussed yet is that the most beneficial relation between artificial affect and action selection on the one hand, and artificial affect and simulation selection on the other is the same, i.e., positive relates to narrow and negative relates to broad. This is important for the Simulation Hypothesis (Hesslow, 2002). One of the cornerstones of this hypothesis is evolutionary continuity (Hesslow, 2002). It must be possible to move, in the evolutionary process, from agents that act reactively to agents that think and act. Our finding that the direction of the most beneficial relation between artificial affect and thought selection and artificial affect and action selection is the same is an indication that at the level of behavior modulation this continuity exists. However, one has to be very careful with such conclusions, as computational models are complex, large structures containing many choices. We return to this point, made in Chapter 7, in Section 8.3.

In the studies reported upon in Chapter 3 and 4, we have used a definition of affect that relates to the positiveness versus negativeness of *mood*. It is a long-term signal originating from the relative success of the agent. As such, we have used artificial affect as a meta-level signal: artificial affect is used to control learning parameters, not as reward. However, the latter is certainly possible (Chapter 6 and Section 8.2).

8.2 Affect, Emotion and Reinforcement

When affect is related to the positiveness versus negativeness of a situation in a short-term, object/situation-directed sense, it relates more to *emotion* than mood. As such, artificial affect can be used to tell the learning agent something about the current situation, instead of about its general situation. Further, affect can be elicited by external factors, such as communicated emotional expressions, instead of originate from the agent itself. We have taken this approach in Chapter 6, and we have shown that communicated affect can help learn an artificial agent. More precise, a human observer reacts affectively (by means of emotion recognition from facial expressions) to a simulated robot while that robot learns. This affective reaction is translated to a positive or negative reward. The reward is used by the robot in addition to the rewards it gets from interacting with the world. This interaction helps the robot to learn a grid-world task.

The main conclusion to be drawn from this study is that affective interaction facilitates robot learning: we have quantitatively shown that a simulated robot learns quicker with social reward than without social reward. However, for this beneficial effect to last, the robot has to learn an additional social reward function that predicts, based on world-state input, the social reward given by the human. If not, the agent simply forgets the social reward when the human stops giving it. This is not a problem if the task has been learned completely, because now the agent already has an optimal model. However it is a problem if the agent is left over to itself after a short social training period. The latter situation is the more plausible and more desirable one. It is more efficient (the human has to observe the robot less often), and it is better related to parent-child interaction: children are not monitored all the time, but in phases. In a non-monitored phase, the child has to try to find out for itself what to do with the guidance given during a monitored phase.

8.3 Formal Models and Computational Limitations

It is interesting to note that affect defined as the positiveness versus the negativeness of a situation (e.g., Gasper & Clore, 2002) is actually a very useful abstraction in the context of Reinforcement Learning. It can be used in many ways, as has been shown in this thesis. However, we have to be careful, again, about conclusions drawn from computational experiments, specifically related to the meaning of the modeled concepts. It can be argued that the way we model affect is quite limited, which is most certainly the case considering the wide variety of emotions and moods that exist in humans. In relation to Reinforcement Learning this definition (and our derived definition of artificial affect) might be adequate, but this does not mean that we have modeled affect in its full glory, or that we can conclude anything about affect in general. Therefore, our psychology-related claims and conclusions have to be interpreted in the context of Reinforcement Learning and instrumental conditioning. Our conclusions are about existence proofs of relations, for they appear beneficial to artificial agents that learn based on different computational models of instrumental conditioning (the versions of RL used in Chapters 3, 4 and 6). As such, they are relevant to experimental psychology. Experimental psychology has difficulties explaining the mechanisms behind relations. In this context, the mechanisms presented in this research are potential candidates that support relations between affect and learning found in the psychological literature. The conclusions should not be carried further than that.

Concrete computer science related results include the control of learning parameters in artificial learning methods by means of abstractions of concepts borrowed from psychology. More specific, artificial affect has successfully been used to control exploration versus exploitation, and affect has been used as reinforcement in an interactive learning setup with a human in the loop. It is very well possible to use affect in a broader sense than the one studied in this thesis. For example, it is interesting to research how affect can be used to control the search through a solution space, as this is also a process of exploration (random jumps, multiple start positions) versus exploitation (hill-climbing). Further, *arousal*, the part of affect that defines the activity or action readiness of the organism—a part we have ignored completely in this thesis—can be modeled and then used to control other parameters. These parameters could be related to the amount of energy available to the agent. Such parameters include the likelihood of acting in the first place and the depth of the thought process.

As mentioned in the previous paragraphs, computational models are limited in their ability to conclude about natural phenomena. This issue has been dealt with related to emotion modeling in Chapter 7. We have shown that it is useful, in fact critical, to use formal models of emotion at an architectural level to advance emotion theory. The analysis has been focused on cognitive appraisal theory, explaining emotions as a result of the subjective evaluation of events in the context of beliefs, desires, and intentions of an agent (being natural or artificial). Our analysis showed that with the formal notation we developed it becomes easier to evaluate whether unexpected behavior resulting from a computational model is due to errors in the computational model or errors in the theory. This is an important issue, as computational models of emotion tend to get very complex and are inspired by many psychological theories (see, e.g., the impressive agent models by Gratch and Marsella, 2004 or Baars and Franklin, 2003). We have further shown that the formal notation can be used to integrate different cognitive appraisal theories, an important issue in the advancement of appraisal theory (Wehrle & Scherer, 2001).

A very valid argument that could be put forward at this point is that we haven't formally described the affect-learning relations studied in Chapter 3 to 6, and as such can not really draw strong conclusions from these studies. We can say two things about this.

First, we did not formally represent the relations studied, and it would be interesting to find out if this is possible using the formalism developed in Chapter 7. However, as argued in Chapter 7 and (Broekens & DeGroot, 2006), emotion psychologist have to also formally annotate the data resulting from, and proposed mechanisms derived from emotion studies. Without this annotation, the computer model can not be evaluated other than in ways done in this thesis or in the work by many other modelers. So, formal modeling by computer scientists is only half the solution, and in this case, half a solution is no solution as there is nothing formal to compare the computer scientist's formal model with. More importantly, the formalism proposed in Chapter 7 is targeted towards cognitive appraisal theory, which is not used as underlying theory for the research in Chapter 3 to 6. We have taken this direction because the number of computational models of emotion based on cognitive appraisal theory is vast, and consequently a formalism targeted at this family of models and theories could have a larger impact.

Second, we can definitely draw conclusions related to psychology from our studies, given that we extensively argued why we modeled affect in the ways we did, as well as how we used it to influence learning. Further, our conclusions should be interpreted as *mechanism existence proofs* than can inspire psychological research, just as psychological research has inspired the modeling work in this thesis. Research can be done in many ways; sometimes the conclusions are clear-cut logical results, sometimes they are hypotheses made plausible. Our conclusions regarding computational results, such as better learning performance, fall into the first category: whatever the underlying mechanisms are or are not based upon, the result is objectively measurable. Conclusions related to the psychological implications of the studies presented in this thesis fall into the second category: given the computational results, the relations and mechanisms we have modeled become more plausible psychologically, although never an exclusive truth.

Samenvatting (Dutch)

Hier zal een korte samenvatting gegeven worden van de resultaten van het onderzoek gepresenteerd in dit proefschrift.

Affect, Gemoedstoestand en Informatieverwerking

Om te overleven moet elk wezen acties selecteren ("wat ga ik nu doen…"). *Actieselectie* is gedefinieerd als het probleem om te beslissen wat de volgende actie zal zijn zodanig dat de kans op overleving wordt geoptimaliseerd. *Reinforcement Learning* (RL) is een manier om leerprocessen op basis van positieve en negatieve terugkoppeling te modelleren door middel van computationele modellen. Een agent leert welke actie in welke situatie welke verwachte waarde heeft. Deze waarde wordt bepaald door de terugkoppelingen die er in de omgeving bestaan en door de ervaringen die de agent met deze omgeving heeft. In RL moet een mobiele agent dus steeds kiezen welke actie gedaan moet worden op basis van de waardes van de verschillende acties. Uiteindelijk wil die agent positieve terugkoppeling maximaliseren en negatieve terugkoppeling minimaliseren.

Een probleem dat de lerende agent hierbij moet oplossen is *exploratie* versus *exploitatie*. Exploratie (trial and error) is het proberen van nieuwe acties en het leren van wat er goed en niet goed aan is. Exploitatie is het kiezen van acties die volgens wat je geleerd hebt de beste zijn. Deze twee processen moeten bij een lerende agent afgewisseld worden. Neem als voorbeeld boodschappen doen in een nieuwe stad. Eerst zoek je naar de kortste weg naar een supermarkt (exploratie), en nadat je dit een aantal keer hebt gedaan denk je te weten wat de kortste weg is. Vervolgens neem je altijd de route waarvan je denkt dat deze het kortst is (exploitatie).

In Hoofdstuk 3 en 4 van dit proefschrift is er onderzocht hoe gemoedstoestand de keuze voor exploratie versus exploitatie kan beïnvloeden. Er is gebruik gemaakt van gesimuleerde robotjes. De robots moeten leren hoe ze door een doolhof kunnen navigeren. Ze moeten zo goed mogelijk de weg naar het doel leren. Je zou kunnen zeggen dat een gesimuleerde muis in een doolhof op zoek is naar kaas.

In Hoofdstuk 2 is een model van gemoedstoestand (affect) voor dit soort lerende robotjes ontwikkeld. Dit model gaat ervan uit dat de robotjes een stemming kunnen hebben die varieert van goed tot slecht afhankelijk van hoe goed het met ze gaat. Hoe goed het gaat hangt af van de gemiddelde hoeveelheid straf en beloning die ze krijgen, ten opzichte van wat ze gewend zijn. Dus, als ze steeds beter weten waar de kaas is (of, als wij steeds beter weten waar de supermarkt is), gaat hun stemming vooruit. Als ze echter steeds meer tijd nodig hebben om de kaas te vinden, of steeds vaker tegen de muur aan lopen (straf) terwijl ze door de doolhof bewegen, dan gaat hun stemming achteruit.

In Hoofdstuk 3 hebben we dit model van gemoedstoestand gekoppeld aan het exploratiegedrag van de gesimuleerde robotjes. Uit dit onderzoek blijkt dat robotjes die gaan exploreren als ze zich slecht voelen en gaan exploiteren als ze zich goed voelen sneller het beste pad naar hun doel leren. Dit is niet altijd zo, maar vooral als het doel (de kaas) plotseling naar een andere plek in de doolhof wordt verplaatst. Robotjes gaan zich dan minder goed voelen (ze vinden het doel immers niet meer), en gaan daardoor exploreren (nieuwe dingen proberen). Hierdoor vinden ze het nieuwe doel sneller dan robotjes die gewoon door blijven lopen op het oude pad naar het doel. Als de affectieve robotjes de nieuwe plek hebben gevonden, gaan ze zich langzaam beter voelen. Hierdoor gaan ze meer exploiteren (goede acties kiezen op basis van wat ze geleerd hebben). Daardoor doen ze minder probeeracties (exploratie), waardoor ze sneller het beste pad leren dan robotjes die geen affectgestuurd exploratiegedrag hebben.

In Hoofdstuk 4 hebben we verder gekeken naar een andere manier om gemoedstoestand te koppelen aan leergedrag. In dit onderzoek is de stemming van de lerende robot gekoppeld aan hoe de robot vooruit denkt. Vooruit denken is in dit geval het anticiperen op de mogelijke gevolgen van een actie, voordat de actie is uitgevoerd. Het is dus een soort van intern simuleren van gedrag om te voorspellen wat er zou kunnen gebeuren. De stemming (weer variërend van goed tot slecht) wordt nu gekoppeld aan de hoeveelheid positieve gedachten die de robot heeft. Als de stemming goed is, denkt deze alleen aan positieve gedachten; als de stemming slecht is, denkt de robot aan zoveel mogelijk. Dus, goed voelen betekent goed denken, en slecht voelen betekent breed denken. Uit dit onderzoek blijkt dat er geen positief effect is van affectgestuurd denken op de leersnelheid van de robot. Deze leert dus niet sneller waar het voedsel is. Wel hoeft de robot *minder* te denken om hetzelfde resultaat te behalen. Hieruit zou dus geconcludeerd kunnen worden dat het gunstig is voor de totale hoeveelheid benodigde denkinzet tijdens een leerproces om vooral breed over mogelijke consequenties na te denken als het minder goed gaat, maar vooral over positieve mogelijke consequenties na te denken als het goed gaat.

Er is een grote kanttekening bij deze resultaten: het zijn computationele modellen! Ook zijn het modellen die getest zijn in simpele omgevingen: kleine doolhofjes met maar een paar verschillenden objecten. Het is dus niet mogelijk om deze conclusies definitief te veralgemeniseren naar bijvoorbeeld menselijk gedrag. Wat wel gezegd kan worden is het volgende. Ten eerste, gemoedstoestand lijkt een nuttige toevoeging voor lerende robots: ze kunnen in sommige situaties op gunstige wijze gebruik maken van hun stemming. Ten tweede, de relaties die onderzocht zijn tussen gemoedstoestand en leren laten aan de cognitieve psychologie zien *hoe* (het mechanisme) gemoedstoestand en leren zouden kunnen samenhangen. Hier kan vervolgens weer verder onderzoek naar gedaan worden.

Affect en Beloning

In Hoofdstuk 6 is een iets andere aanpak gekozen voor de koppeling tussen affect en leren. In de eerdere hoofdstukken was affect een signaal dat door de robot zelf werd gemaakt, en dat samenhing met gemoedstoestand: “hoe gaat het nu met me ten opzichte van wat ik gewend ben”. Dit is een lange termijn interpretatie van affect. In Hoofdstuk 6 is onderzocht hoe affect als signaal door een ander gecommuniceerd wordt aan een lerende robot. Het is voor mensen heel belangrijk dat ze emoties kunnen herkennen van anderen. Deze kunnen je bijvoorbeeld vertellen dat je iets niet meer moet doen, of juist wel. Dit principe van leren door middel van emotionele uitdrukking is onderzocht, maar dan tussen mensen en robots.

Er is onderzocht of een lerende robot beter leert als hij ook gebruik kan maken van een menselijke “ouder”. De ouder kijkt naar de gesimuleerde robot (weer in een doolhof, waar kaas in is verstopt) en de robot kijkt naar de ouder door middel van een *webcam*. De webcam vertaalt de gelaatsuitdrukkingen van de ouder (bijvoorbeeld een onderzoeker) naar een positief of negatief signaal. Dit signaal kan door de robot gebruikt worden als terugkoppeling. Het blijkt dat de robot beter leert als er een observerende ouder bij zit. De robot leert sneller wat de beste weg naar het doel is. Dit signaal werkt het best als de robot ook leert *wanneer* de ouder lacht of boos kijkt (de belangrijkste signalen die in ons model vertaald worden naar positieve of negatieve terugkoppeling). Dus, als de robot het signaal van de ouder alleen gebruikt om zijn gedrag aan te passen maar niet om ook een model op te bouwen van wat die ouder waarover vindt, dan helpt het signaal niet goed. Ook hier geldt weer: dit is een computationeel model, dus oppassen met de conclusies.

Formele Modellen.

In het laatste hoofdstuk (Hoofdstuk 7) is er een compleet andere benadering gekozen om inzicht te krijgen in emotionele processen. Er is voor een bepaald type theorie van emotie, de *appraisal theorie*, een formele taal ontwikkeld die gebruikt kan worden om verschillende appraisal theorieën in op te schrijven. Appraisal theorieën gaan ervan uit dat emoties ontstaan door het vergelijken van een huidige situatie met toekomstige doelen en de actoren en hun rollen daarin. Als iets goed is voor mijn doelen, word ik blij, en andersom. Als iemand anders dat voor elkaar heeft gekregen ben ik die persoon dankbaar. Als ik het zelf heb gedaan ben ik trots. Als iemand anders iets doet dat slecht voor mij is word ik boos. Als er niemand verantwoordelijk voor is word ik verdrietig, etc. Het gaat hier te ver om precies uit te leggen hoe dit in zijn werk gaat; in Hoofdstuk 7 staan vele referenties naar de verschillende theorieën die er bestaan.

Het idee achter de ontwikkelde formele taal is dat verschillende theorieën allemaal beschreven kunnen worden in dezelfde taal. Hierdoor wordt het veel makkelijker om ze met elkaar te vergelijken. Ook kan de formele beschrijving van een dergelijke theorie beter gebruikt worden als basis voor het maken van computationele modellen van emotie. Waarom? Omdat een computationeel model heldere, duidelijke definities nodig heeft, en een formele beschrijving van een theorie duidelijkere definities heeft dan een in taal opgeschreven theorie. In Hoofdstuk 7 wordt laten zien hoe twee verschillende theorieën samengevoegd kunnen worden nadat ze beschreven zijn in de ontwikkelde formele taal. Vervolgens wordt er een computationeel model gemaakt op basis van deze samengestelde theorie.

Glossary

A*ction selection*. The process of selecting actions from a set of actions proposed by some model of the world in order to optimize survival. In Reinforcement Learning, actions have action values resulting from a combination of the reward function and the value function, and the action-selection process selects actions based on these action values. See Tyrell (1993).

Affect. In psychology, affect is usually defined as a two-dimensional abstraction of emotion in terms of arousal and valence (pleasure). Arousal defines the individual's activity level (e.g., physical readiness for action), while valence defines how positive versus negative a situation/object is to the individual. On the long-term timescale, affect relates to mood. On the short-term timescale affect relates to emotion. See Russell (2003), Gasper and Clore (2002), Ashby et al. (1999).

Affect induction. In psychology, the process of experimental manipulation of affect, aimed at inducing experimental subjects with affect (often positive, negative or neutral) in order to measure, e.g., the influence of affect on cognition. See Ashby et al. (1999), Custers and Aarts (2005), Dreisbach and Goschke (2004).

Agent. In this thesis, an autonomous learning entity being, e.g., an adaptive robot or simulated animal. See Jennings et al. (1998).

Alternating-goal task. A specific grid world used in the experiments in this thesis, aimed at testing the ability of an agent to switch from one goal to a second, after interaction with the grid world has been optimized at finding the first goal. See Chapter 3.

Anticipatory simulation. See, *simulation selection.*

Appraisal. In *cognitive appraisal theory*, appraisal refers to the evaluation of a situation in terms of personal meaning or relevance. The result of this evaluation is often described in terms of appraisal dimensions. See van Reekum (2000).

Appraisal dimension. An appraisal dimension influences emotion and can be considered as a variable—e.g., agency, relevance or valence—used to express the result of the appraisal of a perceived object or person.

Arousal. See *affect.*

Artificial affect. A computational model for affect. In this thesis, artificial affect describes how well an agent is doing compared to what it is used to. Arousal is ignored in this model. See Chapter 2.

Artificial emotion. See *computational model of emotion.*

Boltzmann. See *temperature.*

Bottom up. Information processing focused on incoming stimuli.

Candy task. A specific grid world used in the experiments in this thesis, aimed at testing the ability of an agent to first exploit a local optimum and then explore and exploit a global optimum. See Chapter 3.

(Cognitive) appraisal theory. A cognitive theory of emotion assuming that emotion primarily results from an individual's cognitive evaluation of a situation in terms of that individual's believes, desires and intentions. See Arnold (1960), Scherer (2001), Frijda and Mesquita (2000), Smith and Kirby (2000), and many others.

Computational model. An abstraction of a system/phenomenon/theory described in terms of a collection of algorithms such that the resulting description is executable on a computer (i.e., reducible to operations at the Turing machine level).

Computational model of emotion. A model of emotion executable by a computer, based on a psychological or neurobiological theory of emotion. Often such a model is embedded into an artificial agent (learning and non-learning agents), resulting in an artificial emotional agent.

Conditioning. Learning to associate a reinforcement with a situation. See also *instrumental conditioning.*

Credit assignment. In Reinforcement Learning, credit assignment is the problem of associating the right values to (sequences of) actions leading to rewarding or punished outcomes. Credit assignment is often seen as an optimization problem. The solution to this problem is a credit distribution over individual actions and states that, when actions with the highest credit according to the distribution are chosen, reflects the optimal behavior for an agent when the goal of that agent is to maximize cumulative monetary reward. See Sutton and Barto (1998).

Cue inversion. In psychology, used to test, e.g., behavioral flexibility by switching the meaning of a cue in a cued situation with the meaning of a non-cued situation. In this thesis the Cue-inversion-task measures how well an agent is able to cope with such a switch (Chapter 4).

Discounting. In learning, attributing less importance to future reinforcement as compared to current reinforcement. See Sutton and Barto (1998).

Dynamic selection. The use of an action-selection strategy that is controlled or influenced by the agent itself, e.g., a Boltzmann β that is controlled by the agent's artificial affect. See Chapter 3 and 4.

Effort. The behavioral investment needed to complete a certain task. In our experiments effort is used for the number of steps the agent needs to finish one experimental run. *Mental effort* is the simulation investment, e.g., the number of internally simulated steps during one run. See Chapter 4.

Emotion. Hard to define, but when forced, emotion refers to a set of—in animals—naturally co-occurring phenomena including facial expression, motivation, emotional actions such as fight or flight, a tendency to act, and at least in humans but possible in other animals as well, feelings and cognitive appraisal. An emotion is intense, short and directed at something. See Scherer (2001).

Exploitation. Action selection (behavior selection) that is optimal (in terms of some criterion, e.g., cumulative reward) according to the knowledge the agent/animal/robot has of the world it interacts with.

Exploration. Action selection (behavior selection) that is aimed at learning new knowledge about the world an agent/animal/robot interacts with. Explorative action selection is thus often not compatible with exploitative action-selection.

Food. The abstraction of a goal often used in grid-world experiments. Food is thus a positively reinforced location in a grid world.

Forgetting rate. In this thesis, the rate with which an agent forgets learned knowledge of the world due to not using the knowledge. See Chapter 4.

Formalism of appraisal. In the context of Chapter 7, a formal description of a structural theory of appraisal. For all specific formalism-related definitions see Section 6.3.

Gain. In Chapter 4, the amount of effort reduction a simulation strategy gives relative to no simulation, weighted by the amount of simulation effort needed by that strategy. Gain is a measure for the usefulness of a simulation strategy while controlling for the amount of simulation used by that strategy.

Greedy. In action selection, the selection of the best action out of a set of actions. *Non greedy* refers to a selection process whereby not specifically the best action is selected, but actions are selected by means of a stochastic process based on action values.

Grid world. A rectangular grid containing features at grid locations, such as food, walls, and water, with which a simulated robot or agent interacts. Features are perceived by agents inhabiting the grid world. In the discrete case agents can move from one grid position to another by executing actions. In the continuous case, an agent moves a certain amount of simulated distance in a certain direction.

Human-robot interaction. As a subfield of human-computer interaction (HCI), human-robot interaction (HRI) studies the (potential) interaction patterns between humans and robots, as well as resulting (potential) societal changes.

Instrumental conditioning. A natural learning mechanisms by which animals learn behavior by trying actions in context and associating reward and punishment with these actions. As a result, rewarding actions are repeated, while punished actions are avoided. The instrumental part refers to the reinforcement being contingent upon an animals action.

Interactron. In the learning model in Chapter 4, a node that models that a current state-action pair follows a history of state-action pairs.

Learning rate. In Reinforcement Learning, a numerical value that defines how quickly values of states and actions are updated according to new reward experiences. See Sutton and Barto (1998).

Maze. In instrumental conditioning, maze often refers to a typical maze-like experimental environment for an animal (such as rats in a T-maze, a maze in the shape of a T in which the rat has to learn to make a choice). In Reinforcement Learning, see *grid world.*

Markov decision process. A state-transition process that is described by states and state transitions, whereby the probability for any state transition is entirely defined by the previous state.

Mental effort. See *effort*.

Meta-learning. The process of learning about learning parameters, e.g., how should an agent adapt its learning rate, action-selection process, etc.

Model-based Reinforcement Learning. See *Reinforcement Learning*.

Model-free Reinforcement Learning. See *Reinforcement Learning*.

Mood. Mood shares many characteristics with emotion, but in contrast to emotion is not intense, is of long duration and is not specifically directed at an object or person.

Process model. A theory of appraisal that describes, in detail, how the process of appraisal evolves over time, what appraisal processes are activated and when, how the flow of information is between processes, etc. A process model describes the cognitive operations, mechanisms and dynamics by which appraisals, as described by a structural model, are made and how appraisal processes interact. See Reisenzein (2001), and Chapter 7.

Probabilistic learning. See *Reinforcement Learning.*

Punishment. A discouragement for action, a negative reward.

Q learning. A specific model-free version of Reinforcement Learning (see Kaelbling et al., 1996).

Reinforcement (signal). See reward and *punishment*.

Reinforcement learning. A computer model for task learning that solves the credit assignment problem by propagating reinforcement back from the end to the beginning of action sequences. This process is called *value propagation*. *Model-based* Reinforcement Learning uses a world model to propagated values, while *model free* does not use such a model but uses sampling. See Sutton and Barto (1998).

Reversal learning. In instrumental conditioning, the process of unlearning a previously learned behavior.

Reward. An encouragement for action. In Reinforcement Learning the reward of an action and/or state refers to the immediate reward/punishment of that action and/or state.

Reward function. In Reinforcement Learning, a given function that maps states/actions to their rewards.

Run. In Reinforcement Learning experiments, a sequence of trials, usually long enough to conclude convergence (i.e., the agent has learned a certain solution to a certain problem and does not improve any further with more training).

Sampling. In Reinforcement Learning, the process of acquiring a sufficient amount of experiences with the environment in order to build up a balanced value function, without the need for a world model. See Kaelbing et al. (1996).

Simulation. In computer science simulation refers to modeling a system/phenomenon/theory by means of a computational model (a program). The program runs and predicts potential outcomes and behaviors of the system. In the context of the Simulation Hypothesis, see *Simulation Hypothesis.*

Simulation Hypothesis. The Simulation Hypothesis proposes that conscious thought is internal simulation of behavior (i.e., an organism's simulation of interaction between that organism and the environment). See Hesslow (2002).

Simulation selection. In the context of the Simulation Hypothesis, the process of continuously selecting potential next behavior for internal simulation such that action selection is assisted not hindered. See Chapter 4.

Simulation strategy. A certain implementation for *simulation selection*, such as, simulate only the predicted state-action pair that has the highest value (greedy simulation selection). See Chapter 4.

Social reward. In this thesis, used for the reward administered to a robot, deduced from the emotional expression of an observing tutor. The term is used to discriminated between reward resulting from behavior in the grid world and reward resulting from the interpretation of an emotional expression. See Chapter 6.

Somatic Marker (Hypothesis). A somatic marker is a bodily signal that functions as a value signal to the organism. The Somatic Marker Hypothesis states that decision making is influenced by these markers, enabling the organism to quickly prune through a large space of potential next behaviors. See Damasio (1996).

State. A mathematical abstraction for a situation in which an agent can be.

Static selection. The use of a fixed strategy for selecting actions or simulations, such as a using a fixed Boltzmann β value for action selection. See Chapter 4.

Stochastic selection. See *Greedy.*

Structural model. In the context of appraisal theory, a theory of appraisal describing the declarative semantics of appraisal, i.e., the type of processes involved in appraisal, the relation between the processes, the appraisal variables, etc. See Reisenzein (2001), and Chapter 7.

Switch-cost. An agent's goal-switch cost defined in terms of search effort associated with a forced switch from a well-known goal to a new goal. See Chapter 3.

Switch-to-invest task. A grid world used in the experiments in this thesis that tests how an agent copes with a sudden change in the world, where the change consists of the placement of a negatively reinforced "roadblock" just before the food spots. The world thus tests the ability of the agent to suddenly make an investment. See Chapter 4.

Task learning. See *instrumental conditioning.*

Temperature. In action selection, a numerical factor that defines the greediness of action selection by controlling the landscape of the Boltzmann distribution over the action values. The term temperature refers to the fact that the temperate parameter in the Boltzmann distribution models the amount of noise in the distribution. See Chapter 3.

Top-down. Goal oriented or cognitively filtered information processing.

Transition function. In Reinforcement Learning, a probabilistic function that maps state-action pairs to next states. The next state is a potential result from executing the action in the state denoted by the state-action pair. As such, the transition function is a model for (the behavior of) the world under influence of the actions of the agent. See Kaelbling et al. (1998).

Trial. In Reinforcement Learning experiments, the period between the agent's start location and the goal location. A trial thus varies in the amount of steps needed. At the start of the learning process, trials are long, while at the end of the learning process trials are short (if convergence is reached).

Valence. See *affect.*

Value. In Reinforcement Learning the value of an action and/or state refers to the cumulative future reward a certain action and/or state predicts.

Value function. A function, learned using Reinforcement Learning mechanisms, that maps states/actions to their values (in contrast to rewards, which are usually *given* by the reward function for a certain task).

Value propagation. See *Reinforcement Learning*.

Winner-Take-All. A selection process in which the final selection is based on the option with the highest value (or the strongest representation).

Working memory. A capacity-limited short-term memory in which active maintenance of perceived or remembered situations and features is needed. See Baddeley (2000).

World model. See *transition function*.

References

Anderson, J. R. (1995). *Learning and Memory: An Integrated Approach*. John Wiley & Sons, Inc.

Arnold, M. B. (1960). *Emotion and Personality. Vol. I: Psychological Aspects*. Columbia University Press.

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). MIT Press.

Baars, B.J., & Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Sciences, 7*(4), 166-172.

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

Bechara, A., Damasio, H., & Damasio, A. R. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex, 10* (March), 295-307.

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. *JETAI, 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*.

Bovenkamp, E. G. P., Dijkstra, J., Bosch, J. G., & Reiber, J. H. C. (2004). Multi-agent segmentation of IVUS images. *Pattern Recognition, 37*(4), 647-663.

Breazeal, C. (2001). Affective interaction between humans and robots. In: J. Keleman and P. Sosik (eds), *Proc. of the ECAL 2001, LNAI 2159* (pp. 582-591). Springer.

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

Broekens, J. (2007). Emotion and reinforcement: Affective facial expressions facilitate robot learning. *LNAI Spec. Vol. on AI for Human Computing, LNAI 4451* (pp. 113-132). Springer.

Broekens, J., & DeGroot, D. (2004a). Scalable and flexible appraisal models for virtual agents. In: Q. Mehdi and N. Gough (eds.), *Proc. of the 5th Game-On International Conference* (pp. 208-215).

Broekens, J., & DeGroot, D. (2004b). 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., & DeGroot, D. (2004c). Emotional agents need formal models of emotion. In: *Proc. of the 16th Belgian-Dutch Conference on Artificial Intelligence* (pp. 195-202).

Broekens, J., & DeGroot, D. (2006). Formalizing cognitive appraisal: From theory to computation. In: R. Trappl (ed.), *Proc. of the 19th European Meeting on Cybernetics and Systems Research* (pp.595-600).

Broekens, J., & Haazebroek, P. (2007). Emotion and reinforcement: Affective facial expressions facilitate robot learning. *In: Proc. of the IJCAI Workshop on Human Factors in Computing* (pp. 47-54).

Broekens, J., Kosters, W.A., & Verbeek, F. J. (in press). Affect, anticipation and adaptation: Investigating the potential of affect-controlled selection of anticipatory simulation in artificial adaptive agents. *Adaptive Behavior*.

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.

Broekens, J., Kosters W. A., & DeGroot, D. (in press). Formal models of appraisal: Theory, specification, and computational model. *Cognitive Systems Research*, doi:10.1016/j.cogsys.2007.06.007.

Broekens, J., & Verbeek F. J. (2005). Simulation, emotion and information processing: Computational investigations of the regulative role of pleasure in adaptive behavior. In: *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.

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

Charman, T., & Baird, G. (2002). Practitioner review: Diagnosis of autism spectrum disorder in 2- and 3-year-old children. *Journal of Child Psychology and Psychiatry, 43*(3), 289-305.

Chow, B. (2003). *PacMan*. http://www.bennychow.com/pacman_redirect.shtml.

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

Coddington, A., & Luck, M. (2003). Towards motivation-based plan evaluation. In: I. Russell and S. Haller (eds.), *Proc. of the Sixteenth International FLAIRS Conference* (pp. 298-302).

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. In: *Proceedings of the Workshop on Modeling Natural Action Selection* (pp. 225-228). AISB Press.

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.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper & Row.

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*. Penguin Putnam.

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

Dayan, P. (2001). Motivated reinforcement learning. In: *NIPS 14*, 11-18. MIT Press.

Dayan, P., & Balleine, B. W. (2002) Reward, motivation, and reinforcement learning. *Neuron 36*(2), 285-298.

Dehaene, S., Sergent, C., & Changeux, J-P. (2003). A neuronal network model linking subjective reports and objective physiological data during conscious perception. *PNAS, 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.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization.* MIT Press.

Doya, K. (2000). Metalearning, neuromodulation, and emotion. In: G. Hatano, N. Okada and H. Tanabe (eds.), *Affective Minds* (pp. 101-104). Elsevier Science.

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.

Dunn, B. D., Dalgleish, T., & Lawrence, A. D. (2006). The somatic marker hypothesis: A critical evaluation, *Neuroscience & Biobehavioral Reviews, 30*(2): 239-271.

Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robots and Autonomous Systems, 42*, 143-166.

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

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

Gadanho, S. C. (1999). *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. PhD Thesis, University of Edinburgh.

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

Gasper, K., & Clore, L. G. (2002). Attending to the big picture: Mood and global versus local processing of visual information. *Psychological Science, 13*(1), 34-40.

Gmytrasiewicz P. & Lisetti, C. (2002). Emotions and personality in agent design and modeling. In: J.-J.Ch. Meyer and M. Tambe (eds.), *Proc. of Intelligent Agents VIII, LNAI 2333* (pp 21-31). Springer.

Gratch, J. & Marsella, S. (2004). A domain independent framework for modeling emotion. *Journal of Cognitive Systems Research, 5*(4), 269-306.

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

Hebb, D. O. (1949). *The Organization of Behavior*. John Wiley.

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

Henninger, A.E., Jones R.M., & Chown, E. (2003). Behaviors that emerge from emotion and cognition: Implementation and evaluation of a symbolic-connectionistarchitecture. *AAMAS 2003* (pp. 321-328).

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

Heylen, D. Nijholt, A., Akker op den, R., & Vissers, M. (2003). Socially intelligent tutor agents. In: T. Rist, R. Aylett, D. Ballin, and J. Rickel (eds.), *Proc. of the 4th International Workshop on Intelligent Virtual Agents (IVA 2003)* (pp. 341-347).

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). 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.

Isbell, C. L. Jr., Shelton, C. R., Kearns, M., Singh, S., & Stone, P. (2001). A social reinforcement learning agent. *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 377-384). ACM Press.

Jennings, N., Sycara, K. & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems, 1*(1), 7-38.

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

Krödel M., Kuhnert K. (2002). Reinforcement learning to drive a car by pattern matching. In: *24th DAGM Symposium, LNCS 2449/2002* (pp. 322-329). Springer.

Laird, J. (2001). It knows what you're going to do: Adding anticipation to a quakebot. *In: Proc. of the fifth international conference on Autonomous agents* (pp. 385-392). ACM Press.

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.

Lazarus, R. S. (2001) Relational meaning and discrete emotions. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 37-67), Oxford University Press.

Levine, L. J., Prohaska, V., Burgess, S. L., Rice, J. A., & Laulhere, T. M. (2001). Remembering past emotions: The role of current appraisals. *Cognition and Emotion, 15*(4), 393-417.

Lewis, M. D. (2001). Personal pathways in the development of appraisal. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 205-220). Oxford University Press.

Lin, L. J. (1993). *Reinforcement Learning for Robots Using Neural Networks*. Doctoral dissertation. Carnegie Mellon University, Pittsburgh.

Mallery, J. C. (1988). *Thinking about Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers*. Master's Thesis, MIT Political Science Department.

Marsella, S., & Gratch, J. (2001). Modeling the interplay of emotions and plans in multi-agent simulations. *Proc. of the 23rd Annual Conference of the Cognitive Science Society* (pp. 294-299).

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.

Mehrabian, A.. (1980). *Basic Dimensions for a General Psychological Theory*. OG&H Publishers.

Meyer, J.-J.Ch. (2004). Reasoning about emotional agents. In: R. López de Mántaras and L. Saitta (eds.), *Proc. of the 16th European Conference on Artificial Intelligence* (pp. 129-133).

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

Morgado, L. & Gaspar, G. (2005). Emotion based adaptive reasoning for resource bounded agents. *Proc. of the AAMAS'05* (pp. 921-928). ACM Press.

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

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

Oatley, K. (1999). Emotions. In: R. A. Wilson, and F. Kiel (eds.), *The MIT Encyclopedia of the Cognitive Sciences*, MIT Press.

Ortony, A., Clore G. L., & Collins A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press.

Pantic, M., & Rothkranz, L.J.M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22* (12), 1424-1445.

Partala T., & Surakka, V. (2004). The effects of affective interventions in human-computer interaction. *Interacting with Computers, 16*, 295-309.

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

Picard, R. W. (1997). *Affective Computing*. MIT Press.

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.

Reisenzein, R. (2000). Wundt's three-dimensional theory of emotion. In: W. Balzer, J. D. Sneed and C. U. Moulines (eds.), *Poznan Studies in the Philosophy of the Sciences and the Humanities, 22*, 219-250.

Reisenzein, R. (2001). Appraisal processes conceptualized from a schema-theoretic perspective: Contributions to a process analysis of emotions. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 187-204). Oxford University Press.

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

Ropella, G. E. P., Railsback, S. F., & Jackson, S. K. (2002). Software engineering considerations for individual-based models. *Natural Resource Modeling, 15*(1), 5-22.

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.

Roseman, I. J., & Smith, C. A. (2001). Appraisal theories: Overview, assumptions, varieties, controversies. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 3-19). Oxford University Press.

Rozenberg, G., & Spaink, H. (2002). Preface. *Natural Computing, 1*: 1-2.

Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. Tech. rep. CUED/F-INFENG/TR166, Cambridge University.

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

Russell S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (second edition). Prentice Hall.

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. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 92-120). Oxford University Press.

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

Singh, S., Barto A. G., & Chentanez N. (2004). Intrinsically motivated reinforcement learning. *Proc. NIPS'04*. MIT Press.

Smith, C. A., & Kirby, L. D. (2000). Consequences require antecedents: Toward a process model of emotion elicitation. In: J. P. Forgas (ed.), *Feeling and Thinking: The role of Affect in Social Cognition* (pp. 83-108). Cambridge University Press.

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).

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In: *Advances in Neural Information Processing Systems 8* (pp. 1038–1045). MIT Press.

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

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

Tesauro, G. (1994). TD-Gammon, a self-teaching backgammon program, achieves master level play. *Neural Computation, 6* (2), 215-219.

Theocharous, G., Rohanimanesh, K., & Mahadevan, S. (2001). Learning hierarchical partially observable Markov decision processes for robot navigation. *In: Proc. of the IEEE Conference on Robotics and Automation* (pp. 511-516).

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind, 59*, 433-460.

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. 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 Belgian-Dutch Conference on Artificial Intelligence* (pp. 187-194).

van Reekum, C. (2000). *Levels of Processing in Appraisal: Evidence from Computer Game Generated Emotions*. PhD Thesis nr. 289, Université de Geneve, Section de Psychology.

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

Wehrle, T., & Scherer, K. R. (2001). Towards computational modeling of appraisal theories. In: K. R. Scherer, A. Schorr and T. Johnstone (eds.), *Appraisal Processes in Emotion: Theory, Methods, Research* (pp. 350-368). Oxford University Press.

Westmeyer, H. (1989). Psychological theories from a structuralist point of view: A first introduction. In: H. Westmeyer (ed.), *Psychological Theories from a Structuralist Point of View* (pp. 1-12). Springer.

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

Zajong, R. B. (2000). Feeling and thinking: Closing the debate over the independence of affect. In: J. P. Forgas (ed.), *Feeling and Thinking: The role of Affect in Social Cognition* (pp. 31-58). Cambridge University Press.