AISB 2004 Convention:

Motion, Emotion and Cognition

AISB The Society for the Study of Artificial Intelligence and the Simulation of Behaviour

Proceedings of the AISB 2004

Symposium on Emotion, Cognition, and Affective Computing

29 March – 1 April, 2004

ICSRiM, University of Leeds, Leeds LS2 9JT, UK

www.leeds.ac.uk/aisb www.icsrim.org.uk

AISB 2004 Convention

29 March – 1 April, 2004
ICSRiM, University of Leeds, Leeds LS2 9JT, UK
www.leeds.ac.uk/aisb   www.icsrim.org.uk

Proceedings of the AISB 2004 Symposium on Emotion, Cognition, and Affective Computing

Published by

The Society for the Study of Artificial Intelligence and the Simulation of Behaviour

http://www.aisb.org.uk

ISBN 1 902956 36 6

Contents

The AISB 2004 Convention

On behalf of the local organising committee and all the AISB 2004 programme committees, I am delighted to welcome you to the AISB 2004 Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour (SSAISB), at the University of Leeds, Leeds, UK.

The SSAISB is the oldest AI society in Europe and it has a long track record of supporting the UK AI research community. This year, the underlying convention theme for AISB 2004 is "*Motion, Emotion and Cognition*", reflecting the current interest in such topics as: motion tracking, gesture interface, behaviours modelling, cognition, expression and emotion simulation and many others exciting AI related research topics. The Convention consists of a set of symposia and workshop running concurrently to present a wide range of novel ideas and cutting edge developments, together with the contribution of invited speakers:

- Prof Anthony Cohn
  Cognitive Vision: integrating symbolic qualitative representations with computer vision;
- Prof Antonio Camurri
  Expressive Gesture and Multimodal Interactive Systems;
- Dr David Randell
  Reasoning about Perception, Space and Motion: a Cognitive Robotics Perspective; and
- Dr Ian Cross
  The Social Mind and the Emergence of Musicality,

not to mention the many speakers invited to the individual symposia and workshop, who will made the Convention an exciting and fruitful event.

The AISB 2004 Convention consists of symposia on:

- Adaptive Agents and Multi-Agent Systems;
- Emotion, Cognition, and Affective Computing;
- Gesture Interfaces for Multimedia Systems;
- Immune System and Cognition;
- Language, Speech and Gesture for Expressive Characters; and the
- Workshop on Automated Reasoning.

The coverage is intended to be wide and inclusive all areas of Artificial Intelligence and Cognitive Science, including interdisciplinary domains such as VR simulation, expressive gesture, cognition, robotics, agents, autonomous, perception and sensory systems.

The organising committee is grateful to many people without whom this Convention would not be possible. Thanks to old and new friends, collaborators, institutions and organisations, who have supported the events. Thanks the Interdisciplinary Centre of Scientific Research in Music (ICSRiM), School of Computing and School of Music, University of Leeds, for their support in the event. Thanks to the symposium chairs and committees, and all members of the AISB Committee, particularly Geraint Wiggins and Simon Colton, for their hard work, support and cooperation. Thanks to all the authors of the contributed papers, including those which were regretfully not eventually accepted. Last but not least, thanks to all participants of AISB 2004.
We look forward to seeing you soon.

Kia Ng
AISB 2004 Convention Chair
ICSRiM, University of Leeds,
School of Computing & School of Music,
Leeds LS2 9JT, UK
kia@kcng.org   www.kcng.org

Proceedings of the AISB 2004 Symposium on Emotion, Cognition, and Affective Computing

Symposium Preface

Welcome to the 2004 AISB Symposium on Emotion, Cognition and Affective Computing. This symposium contains a number of papers on these topics, ranging from philosophical investigations of theories of mind to applications of affective computing.

We would like to thank all the people who submitted or reviewed papers, and the AISB for handling the organizational side of the conference. We hope that you will enjoy the conference.

The Organising Committee

Chair: Colin Johnson, University of Kent at Canterbury, England

Programme Committee:

Roman Belavkin, Middlesex University, England

Bridget Cooper, University of Leeds, England

Luis Custodio, Instituto Superior Technico, Portugal

Darryl Davis, University of Hull, England

Gareth Jones, Dublin City University, Ireland.

David Moffat, Glasgow Caledonian University, Scotland

Juan Jesus Romero Cardalda, University of Coruña, Spain

On Relation between Emotion and Entropy

Roman V. Belavkin
*School of Computing Science, Middlesex University
London NW4 4BT, UK
R.Belavkin@mdx.ac.uk

Abstract

The ways of modelling some of the most profound effects of emotion and arousal on cognition are discussed. Entropy reduction is used to measure quantitatively the learning speed in a cognitive model under different parameters' conditions. It is noticed that some settings facilitate the learning in particular stages of problem solving more than others. The entropy feedback is used to control these parameters and strategy, which in turn improves greatly the learning in the model as well as the model match with the data. This result may explain the reasons behind some of the neurobiological changes, associated with emotion and its control of the decision making strategy and behaviour.

1 Introduction

It is popular to believe now that emotion is an important (if not essential) component of intelligence (Salovey and Mayer, 1990). This is, however, hard to prove unless some quantitative methods are introduced that will allow us to evaluate such claims in an experiment. An example of such an experiment could be a competition between several agents, with architectures incorporating various theories of emotion and cognition. In practice, however, the results of such an experiment would very hard to interpret because of the great number of components (e.g. perception, memory, planning, action, etc) involved in the agents' architectures.

The research described in this paper pursues a different approach by studying the effects of emotion on decision making and learning. Using entropy reduction as a quantitative measure of learning allows for a better analysis and comparison of the results from different experiments.

The ability to learn is one of the most important features of intelligent systems. While leaving to philosophers the question of what is the purpose of learning, let us assume that this process is beneficial to intelligent systems, and the faster and more effectively it occurs the better. From information theory point of view learning is equivalent to reducing the uncertainty (entropy) about the environment and the system itself within this environment. Many areas of artificial intelligence have already successfully employed the mathematical apparatus of information theory, which advanced greatly the neural networks learning algorithms, search methods and case–based reasoning systems. Recently, the notions of information and entropy have been applied to analyse and control cognitive models (Belavkin and Ritter, 2003). In particular, it became possible for the models implemented in hybrid cognitive architectures, such as ACT–R (Anderson and Lebiere, 1998), which mixes the high level symbolic processing with the low level subsymbolic computations accounting for fuzzy or probabilistic properties of cognition.

The comparison of model results with data (e.g. from human subjects or animals) is one of the most important aspects of the cognitive modelling research. A cognitive model of a classical animal learning experiment will be used in this study to evaluate theoretical predictions.

In the next Section, the most general effects of basic emotions and arousal on behaviour will be discussed and grounded in the relevant literature. The ambiguity of the term emotion will be avoided by replacing it with the principle components of emotions.

The notion of entropy and its application to cognitive models will be discussed in Section 3. This section will repeat some of the previous work (Belavkin and Ritter, 2003). Section 4 will highlight how speed of learning in the model varies as a function of some parameters in the architecture. These parameters (namely the noise variance and goal value used in decision making mechanism) have been used before to simulate different levels of motivation and arousal (Lovett and Anderson, 1996; Anderson and Lebiere, 1998; Belavkin, 2001). The entropy reduction will be used to measure the speed of learning in the model.

Section 5 will discuss the idea of using the entropy of success as a feedback parameter to control the decision making mechanism of the architecture. It will be shown how entropy evaluating model's own performance moderates the choice strategy and controls the behaviour making it more adaptable. In addition, the model match with the data improves, which supports the idea that a similar strategy control takes place in subjects. Some more speculative ideas about the role of emotion in evaluating the entropy and controlling the behaviour will be discussed in the end of the paper.

2 The Principle Components of Emotions

The important role of emotion in cognition has been extensively discussed in the literature, particularly over the last two decades (Salovey and Mayer, 1990; Damasio, 1994; LeDoux, 1996). Despite the great interest in the subject of emotion across several disciplines of science, there is still lack of understanding and clear definition of what emotion actually is. Psychologists and philosophers still cannot agree on some of the fundamental points in the subject, such as what comes first: Feelings or thought? (Schachter and Singer, 1962; Zajonc, 1980).

This ambiguity is multiplied when one attempts to integrate emotion into a unified theory of cognition, and into its computational implementations, such as ACT–R (Anderson and Lebiere, 1998) or SOAR (Newell, 1990). The need to include emotion into cognitive models, however, is rarely disputed (Simon, 1967). With the existence of many computational models of affect (see Hudlicka and Fellous (1996) for a review) and even a greater number of different emotions (Lambie and Marcel, 2002), the problem seems to be intractable. However, the dimensionality can be reduced if we concentrate our research on measurable and the more consistent features of the phenomena, or what we shall call the *principle components of emotions*.

Probably the most common measure of various emotional experiences is *valence* indicating whether an emotion is positive or negative. Cannon (1929) argued that all emotions can be classified into 'fight or flight', which is probably not far from the truth. Another important measure is *arousal*, or the intensity of emotional experience. Arousal is a broad term covering a variety of phenomena, but generally it is associated with different levels of activation of the autonomic nervous system (ANS), and it can be influenced by external or internal stimulation including emotion (Humphreys and Revelle, 1984). As has been shown by Russell (1983, 1989), valence and arousal are the two most common dimensions in classifications of emotions, and they are included in many other classifications (Plutchik, 1994).

Both valence and arousal are measurable and even predictable. Indeed, negative emotions occur when we experience a failure in achieving a particular goal. On the contrary, a success is accompanied by positive emotions. Arousal can be either measured directly in subjects (e.g. using galvanic skin response), or predicted based on the strength of the stimuli (e.g. reward or penalty). Therefore, in this paper, when discussing the role of emotion in cognition, we shall concentrate on the effects of arousal and valence, and we shall not consider other aspects of the phenomenon, such as particular emotions or their role in social interaction and so on.

On individual level, emotion is known to play a role in different aspects of cognition, such as perception, memory, action and learning (LeDoux, 1996). There is quite a lot of experimental evidence suggesting the relation between arousal and cognitive performance. For example, the studies of the inverted–U effect showed the relation between arousal and the speed of learning (Yerkes and Dodson, 1908; Mandler and Sarason, 1952; Matthews, 1985). Another series of experiments showed how the expectation of positive or negative outcomes may change the decision making strategy (Tversky and Kahneman, 1981; Johnson and Tversky, 1983). Below is the summary of some effects of valence and arousal that can be useful in designing a cognitive model:

- Positive valence is associated with success, choice involving gains, risk aversive behaviour. Negative valence is associated with failure, choice involving losses, the behaviour is usually more risk taking (Tversky and Kahneman, 1981; Johnson and Tversky, 1983).

- Low arousal is associated with low level of stimulation or motivation, actions requiring less efforts are more likely. High arousal is associated with high level of stimulation or motivation, actions involving more efforts are more probable (Humphreys and Revelle, 1984).

It has been suggested before (and will be discussed in Section 4 of this paper) how to achieve the above types of behaviour in cognitive models using parameters manipulation (Belavkin, 2001). The speed of learning in the model under these parameters settings will be measured by means of entropy reduction. In the next section, we discuss some definitions of entropy and an example of calculating it a cognitive model.

3 Information and Learning

Learning is one of the most important characteristics of intelligence. It allows a subject or a system to improve the performance in certain tasks or class of problems. The most obvious measure of such an improvement is an increase of success rate, or equivalently a reduction of failures (errors). Ultimately, learning reduces the uncertainty of the outcome with the success being more probable one. Thus, entropy reduction could be a convenient measure of learning. However, in practice it is impossible to measure directly in subjects the parameters necessary for entropy computations (e.g. synaptic weights), and traditionally learning is judged based on external observations (i.e. the reduction of errors such as shown on Figure 1).

Unlike the brains of subjects, however, cognitive architectures allow for a relatively easy access to all the internal variables. This opened a possibility to measure the learning in cognitive models directly by calculating the entropy change or information (Belavkin and Ritter, 2003). The advantage of using the entropy is that it provides a compact display of the internal changes in a model as a result of learning, which may not always have external manifestations. In this section, the use of entropy to describe learning in intelligent systems will be described and shown on example of a cognitive model.

3.1 Entropy and surprise

In the most general case, entropy H is a monotonous function describing the complexity (or uncertainty) of a system, such as $H = \ln M$, where M is the number of states a system can be in. This canonical definition assumes no information about the probabilities of individual states. If, however, we know the probabilities $P(\xi)$ of different (random) states ξ, then the entropy can be calculated as:

$$H(\xi) = -E\{\ln P(\xi)\} = -\sum_{\xi} P(\xi) \ln P(\xi)\,, \qquad (1)$$

where $E\{\cdot\}$ denotes the expected value operator. If all states ξ are equally probable, then entropy (1) equals $\ln M$, and it corresponds to the maximum value of H for given M. Thus, the uncertainty can be reduced if by means of Bayesian estimation we find out which states have greater likelihood. Shannon (1948) defined information as the difference between entropy before and after an observation of some event y:

$$I(x, y) = H(x) - H(x \mid y)$$

Here, x denotes some variable, the information about which is received indirectly through observation of y.

Interestingly, information and entropy have been used before to explain one basic emotion — surprise. Indeed, the lower is the probability P of event ξ, the greater is the amount of information $-\ln P(\xi)$ received when this event happens (i.e. the greater is the surprise). This early observation points to the possibility that our nervous system and body reacts to the amount of information received, and the feedback seems to be proportional to this amount. Note, however, that surprise can be positive as well as negative, and the reaction can be different in each case. In this paper, we shall look more carefully into the nature of such a feedback, and investigate using a cognitive model whether this feedback is beneficial for an intelligent system (i.e. helps in learning and adaptation).

3.2 Uncertainty of success

It is quite difficult to estimate the entropy of a large system with many states (e.g. a cognitive model). However, for an intelligent system it is possible to look at the problem from a different perspective: The uncertainty of whether it achieves the goal or not (Belavkin and Ritter, 2003). The *entropy of success* has been defined as

$$H_{\mathrm{SF}} = -\left[P(\mathrm{F}) \ln P(\mathrm{F}) + P(\mathrm{S}) \ln P(\mathrm{S})\right]\,, \qquad (2)$$

where $P(\mathrm{S})$ is the probability of success in achieving the goal, and $P(\mathrm{F})$ is the probability of failure. Note that $P(\mathrm{F}) = 1 - P(\mathrm{S})$. If a system (e.g. a cognitive model) has to choose from a set of n alternative decisions to achieve the goal, then the probability of success is:

$$P(\mathrm{S}) = \sum_{i=1}^{n} P(\mathrm{S}, i) = \sum_{i=1}^{n} P(\mathrm{S} \mid i) P(i)\,, \qquad (3)$$

where $P(\mathrm{S}, i)$ is the joint probability of successful outcome and ith decision, $P(\mathrm{S} \mid i)$ is the conditional probability of success given that ith decision has been made, and $P(i)$ is the probability of ith decision. Thus, to calculate the entropy of success H_{SF}, one should estimate probabilities $P(\mathrm{S} \mid i)$ and $P(i)$, which depend on specific architectural implementation (i.e. SOAR, ACT–R, neural networks, etc).

Conditional probabilities $P(\mathrm{S} \mid i)$ represent the prior knowledge about the likelihood to achieve a success, if certain decisions (and associated actions) are taken. Note that a problem solver may not be aware of or not considering some decisions initially. However, the number of decisions n to choose from may increase with time as the result of learning. Probability $P(i)$ depends on the way the decision making (e.g. rule selection algorithm) is implemented. Thus, $P(i)$ is more related to the architecture rather than the knowledge of a system. As an example, let us consider the ACT–R cognitive architecture (Anderson and Lebiere, 1998).

3.3 Computation of entropy in ACT–R

ACT–R (Anderson and Lebiere, 1998) is a general purpose hybrid cognitive architecture for developing cognitive models that can vary from simple reaction tasks to simulations of pilots navigating airplanes and operators of airtraffic control systems. ACT–R follows the approach of *unified theories of cognition* (Newell, 1990), in which several theories about different aspects of cognition are used in a single simulation system. Today, ACT–R has emerged as the architecture of choice for many cognitive modelling problems.

In ACT–R, decisions are encoded in a form of production rules, and during the model run the number of successes and failures of each rule is recorded by the architecture. This information is used to estimate empirically the probabilities $P(\mathrm{S} \mid i)$ of success for ith rule:

$$P(\mathrm{S} \mid i) \approx P_i = \frac{\mathrm{Successes}_i}{\mathrm{Successes}_i + \mathrm{Failures}_i}\,. \qquad (4)$$

Here P_i is statistics of ith rule. In addition, ACT–R records the efforts (i.e. time) spent after executing the rule and actually achieving the goal (or failing). This information is used to calculate the average cost C_i of ith rule. Parameters P_i and C_i represent subsymbolic information about the decisions, and can be learned statistically. On symbolic level, a model can learn new rules as well as new facts used by these rules.

When several alternative rules are available that match the current working memory state (i.e. the current goal, perception, retrieved facts), then one rule has to be selected using the conflict resolution mechanism. In ACT–R, this is done by maximising the expected utility of rules in the conflict set: $i = \arg\max U_i$, where

$$U_i = P_i G - C_i + \xi(\sigma^2)\,. \tag{5}$$

The above equation has allowed ACT–R to model successfully some important properties of human (and animals) decision making: Probability matching (use of P_i in utility); The effect of a payoff value (G represents the goal value); Stochasticity (the utility is corrupted by zero–mean noise of variance σ^2) (Anderson and Lebiere, 1998).

Although there are other mechanisms in ACT–R, such as chunks (facts) retrieval, that may affect rules' selection, the probability $P(i)$ that ith rule will be chosen can be approximated by Boltzmann equation as:

$$P(i) \approx \frac{e^{\bar{U}_i/\tau}}{\sum_{j=1}^{n} e^{\bar{U}_j/\tau}}, \tag{6}$$

where $\bar{U}_i$ is the utility not corrupted by the noise, and $\tau = \sqrt{6}\sigma/\pi$ is called the *noise temperature*. Now, using approximations (6) and (4), one can calculate the success probability (3) and entropy of success (2).

3.4 A model example

The reduction of entropy of success has been used to analyse the learning in an ACT–R model of the Yerkes and Dodson (1908) experiment (Belavkin, 2003). In this classical experiment, mice were trained over several days to escape discrimination chamber (a box with two doors) from one particular door, and the number of errors was measured for every day. Figure 1 shows an example of the learning curve representing the number of errors produced by the model in this task during 10 tests per each simulated day. The learning curve, however, does not provide a very detailed picture of what and when is learned.

The performance of the model improves because it learns new production rules, and then by trying these rules the model updates their statistics (P_i and C_i) and uses the most efficient and effective ones. Figure 2 shows the traces of probabilities P_i of production rules relevant to the problem goal in the same experiment. One can see that as new rules and statistics are learned after Day 1, the number of errors decreases (see Figure 1). However, the model produces more errors during Days 5, 6 and 7, which means that the model did not have sufficient knowledge, and the errors forced the model to learn more rules. The model learned new rule during Day 5, but the trace of its statistics indicates that the rule was not very helpful (probability of success quickly decreased to $P_i \approx .5$). The new rules learned on Day 7 turned out to be more successful, and the model did not produce any errors after simulated Day 8. One can see that probability trace reveals much more about the learning in the model than the number of errors.

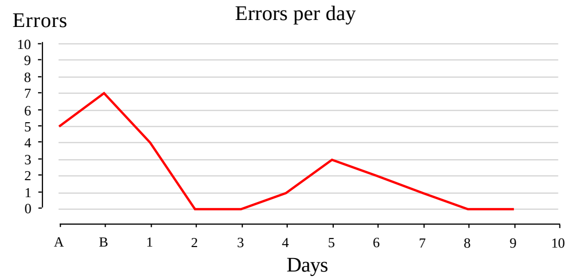

Figure 1: Error curve produced by the model in one experiment.

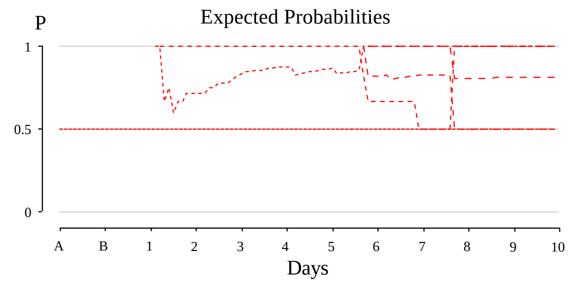

Figure 2: Dynamics of probabilities of rules matching the problem goal. The number of curves increases as new rules are being learned.

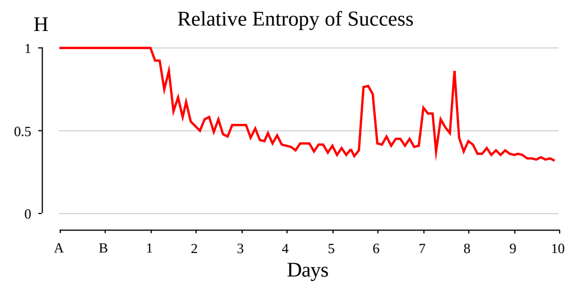

Figure 3: Relative entropy of success of the choice rules. Entropy increases on errors (see Figure 1) and when new rules are learned.

Figure 3 shows the dynamics of relative entropy of success (relative to the maximum entropy $\ln 2$), calculated using equations (4) and (6) over the probabilities of rules shown on Figure 2. The entropy clearly decays over time indicating the amount of information gained by the model. Also, the entropy increases when the model produces errors, which confirms the idea that entropy of success predicts how certain is the outcome. However, one may notice that the entropy increases most dramatically when new rules are learned (i.e. Days 5 and 7). This can be explained as follows. When new rules are created, the number n of decisions increases, thus making the system more complex (recall that entropy is a function of the number of states). Moreover, the probabilities $P(\mathrm{S} \mid i)$ of the new rules initially have default prior estimates (e.g. .5), and they can only be updated statistically after their application. If the new rules improve the performance, then the entropy of success reduces again (see Day 8, Figure 3).

This example illustrates how entropy change or information can be used as a quantitative measure of learning in a cognitive model. In the next section, the entropy will help analyse how the speed of learning in the model varies as a function of parameters settings in the ACT–R architecture.

4 Variable speed of learning

In ACT–R, the choice of decisions does not depend only on the statistical information about the rules (i.e. estimates of probabilities). Indeed, choice probability (6) depends also on two global parameters in the architecture: The amount of noise (noise variance σ^2 parameter) and the goal value G used in the utility equation (5). Asymptotic analysis of choice probability as a function of σ^2 and G has suggested how different levels of arousal and valences can be simulated in an ACT–R model (Belavkin, 2001):

- At a low noise variance σ^2, the choice is more rational and driven by utility maximisation. Thus, it can be well suited for simulation of the risk aversive behaviour typical for choice with positive expectations (Tversky and Kahneman, 1981; Johnson and Tversky, 1983).
- On the contrary, high noise variance leads to a risk taking, irrational choice, which is less defined by utility maximisation. According to Tversky and Kahneman (1981), this is characteristic of choice with high expectation of a negative outcome.
- At a low goal value G, the costs C_i make more significant contribution to the utility (5). Thus, decisions with higher costs are less likely to be chosen. This is suitable for simulating a low arousal state.
- On the contrary, high goal value G is better for simulating a high arousal level, because under these conditions the model is more likely to take costly decisions.

Let us measure how the speed of learning in the model changes under different conditions. We shall use the entropy reduction as a measuring tool. However, because one of the parameters to be changed is noise variance, it is necessary to make the calculation of entropy independent of these changes. This means substituting the choice probability (6), which depends on τ (noise temperature), by a different probability. For example, we can assume that the choice of a rule is completely random: $P(i) = \frac{1}{n}$, where n is the number of rules (decisions). In this case, probability of a success $P(\mathrm{S})$ can be calculated as

$$P(\mathrm{S}) = \frac{1}{n}\sum_{i=1}^{n} P_i \,. \qquad (7)$$

The entropy associated with this probability (calculated similarly by eq. 2) can be used to estimate the knowledge accumulated in the system in the form of empirical probabilities P_i, because it is independent of the way the decisions are made. We refer to this entropy as the *entropy of knowledge* H_k.

The experiments showed that H_k decays differently under different noise variance settings. It turns out that although noise hinders the performance of the model, at the same time it may help learn faster. Figure 4 illustrates the probability learning in the model for two noise settings: Left plot shows traces of probabilities with low noise ($\tau = 1\%$ of goal value G), and right plot for high noise settings ($\tau = 20\%$).[1] One can see that at a higher noise settings (top right), probabilities of rules were updated much more often than at a lower noise (top left). Therefore, the model on the right has better estimates of probabilities. Also, the new and probably more successful rules have been learned earlier in the case of high noise.

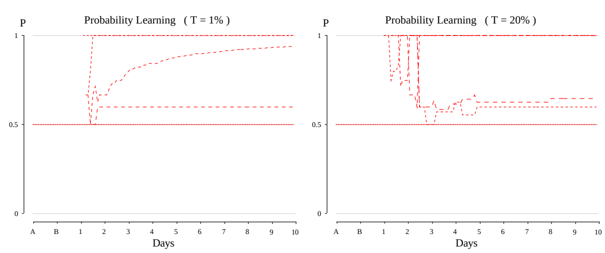

Figure 4: Probability learning under a low noise (left) and a high noise conditions (right).

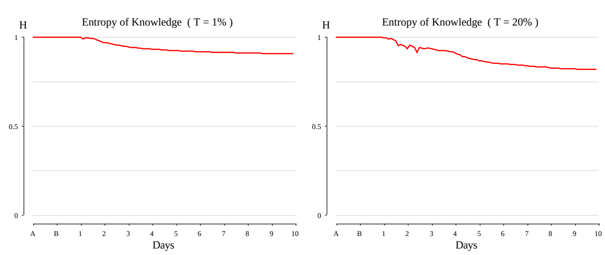

Figure 5: Dynamics of entropy under a low noise (left) and a high noise condition (right).

The corresponding traces of entropies H_k are shown on Figure 5. One can see that by day 10 the entropy on the right plot decayed significantly more than on the left plot. Thus, by day 10 the model with a greater noise gained more information than the model with less noise. These

[1] Here noise temperature is calculated as a proportion of the goal value: $\frac{1}{G}\tau \cdot 100\%$.

results confirm the idea that exploratory behaviour, triggered by an noise increase in ACT–R, facilitates learning in the model.

In the next section, the question of adaptation of behaviour and dynamic control over the parameters in the architecture will be discussed.

5 Entropy feedback and adaptation

The analysis of H_{k} reduction for different noise settings suggested that an intelligent system could benefit from dynamic control over the noise variance. Indeed,

1. At the beginning of solving a problem, exploratory behaviour (high noise) would help gaining the information about the task or the environment more quickly.

2. After the important knowledge has been acquired, the choice should concentrate on more successful decisions, which is achieved by the reduction of noise. This should improve the performance.

3. If the environment changes and the number of errors suddenly increases, then a noise increase can speed–up the learning and adaptation of behaviour.

Note that the dynamics of the noise variance, described above, corresponds to the dynamics of entropy in the model (e.g. Figure 3). A simple way to control the noise variance by the entropy parameter has been proposed recently (Belavkin, 2003). More specifically, noise temperature τ was modified in time as:

$$\tau(t) = \tau_0 H_{\mathrm{SF}}(t)\,, \tag{8}$$

where t is time, and $\tau_0 = \tau(0)$ is the initial value of the noise. One can view the noise here as a compensation for the 'missing information', and the otherwise rational, utility–based choice behaviour is corrupted proportionally to the uncertainty.

As predicted, the model with dynamic noise converges faster to a successful behaviour (no errors), and adapts better to changes. What is even more interesting, is that the model fit to the data has improved as well: In one experiment, R^2 increased from $.77$ to $.86$ and the root mean square (RMS) error reduced from 13.2% to 8.8%. Figure 6 shows the learning curves from the static noise model (top) and dynamic noise model (bottom) compared against the data from Yerkes and Dodson (1908). A similar improvement has been consistent across several data sets.

The dynamics of noise variance, controlled by the entropy feedback, implements one well–studied heuristics. Indeed, by looking at the Boltzmann equation (6), one can notice that the decrease of noise temperature τ is similar to the optimisation by simulated annealing (Kirkpatrick et al., 1983).

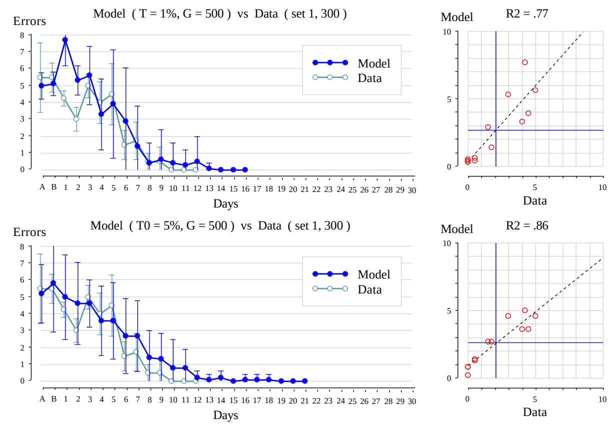

Figure 6: Static noise model (top) and dynamic noise model (bottom) compared with the data (Yerkes and Dodson, 1908). The dynamic model achieves the better match.

Furthermore, noise variance is not the only parameter in the ACT–R conflict resolution that can optimise the learning process. It was shown that goal value G controls the type of the search (Belavkin, 2001): Low G implements the breadth–first search, while high G corresponds to the depth–first search strategy. A search method combining these two strategies is known as the best–first search (from breadth to depth). Thus, gradual increase of G during problem solving can implement the best–first search method.

One can see that the suggested dynamical control of the decision making parameters in the architecture implements some well–known optimisation heuristics, and, therefore, should improve the overall problem solving performance.

6 Discussion

It has been shown in the previous section how dynamic control over two parameters in the ACT–R cognitive architecture improves the learning and adaptive capabilities of the model. In particular, entropy of success has been used as a feedback parameter to control the choice strategy. In addition, this control has improved the match between the model and data. On the other hand, the same parameters have been used to simulate the effects of the principle components of emotions (valence and arousal). Therefore, the dynamic changes of the parameters during problem solving may represent the changes in the behaviour due to experiencing emotions of positive or negative valence and the resulting changes of the arousal level. This idea is supported by a number of works in neuroscience and artificial neural networks.

Indeed, in neural networks, the effect of noise can be simulated by changing the bias (or activation threshold) of neurons (Hinton and Sejnowski, 1986). Some neurotransmitters in the brain have a similar effect, and there are areas of the brain (e.g. amygdala) that have connections with the areas of neocortex believed to be responsible for decision–making (LeDoux, 1996). The role of such interactions have been discussed in the reinforcement learning literature (Sutton and Barto, 1981; Barto, 1985). However, one of the unknown variables there is the amount of reinforcement (e.g. the noise temperature). It has been shown how the entropy of success may help optimise this parameter. Interestingly, entropy and noise temperature have been used for control in the work on analogy by Hofstadter and Marshall (1993).

Today, the idea that emotion plays an important role in controlling and regulating the decision making and actions aspects of cognition is shared by many researchers (Bartl and Dörner, 1998; Sloman, 2001). The results, discussed in this paper, illustrated how the learning in an intelligent system can be improved by using the entropy of success of the system to moderate and control its own behaviour. These observations suggest that appreciation of the system's own performance (entropy of success) and regulating the decision making strategy may indeed be one of the main functions of emotional system in the brain. Including such an information theoretic feedback mechanism into the design of cognitive models, agent architectures or robots will not only improve their performance, but also will extend our knowledge about the mind and emotion within its context.

References

J. R. Anderson and C. Lebiere. *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.

Christina Bartl and Dietrich Dörner. PSI: A theory of the integration of cognition, emotion and motivation. In Frank E. Ritter and Richard M. Young, editors, *Proceedings of the Second European Conference on Cognitive Modeling*, pages 66–73, Nottingham, UK, April 1998. Nottingham University Press.

A. G. Barto. Learning by statistical cooperation of self–interested neuron–like computing elements. *Human Neurology*, 4:229–256, 1985.

Roman V. Belavkin. The role of emotion in problem solving. In Colin Johnson, editor, *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, pages 49–57, Heslington, York, England, March 2001. AISB.

Roman V. Belavkin. *On Emotion, Learning and Uncertainty: A Cognitive Modelling Approach*. PhD Thesis, The University of Nottingham, Nottingham, NG8 1BB, United Kingdom, March 2003.

Roman V. Belavkin and Frank E. Ritter. The use of entropy for analysis and control of cognitive models. In Frank Detje, Dietrich Dörner, and Harald Schaub, editors, *Proceedings of the Fifth International Conference on Cognitive Modelling*, pages 21–26, Bamberg, Germany, April 2003. Universitäts–Verlag Bamberg.

W. B. Cannon. *Bodily changes in pain, hunger, fear and rage*. Appleton, New York, 1929.

A. R. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain*. Gosset/Putnam Press, New York, NY, 1994.

G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. In *Parallel Distributed Processing*, pages 282–317. MIT Press, Cambridge, Mass., 1986.

D. R. Hofstadter and J. B. D. Marshall. A self–watching cognitive architecture of high–level perception and analogy–making. Technical Report 100, Indiana University Center for Research on Concepts and Cognition, 1993.

Eva Hudlicka and Jean-Marc Fellous. Review of computational models of emotion. Technical Report 9612, Psychometrix, Arlington, MA, April 1996.

Michael S. Humphreys and William Revelle. Personality, motivation, and performance: A theory of the relationship between individual differences and information processing. *Psychological Review*, 91(2):153–184, April 1984.

E. Johnson and A. Tversky. Affect, generalization, and the perception of risk. *Journal of Personality and Social Psychology*, 45:20–31, 1983.

S. Kirkpatrick, C. D. Gelatt, and Jr. M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598): 671–680, May 1983.

John A. Lambie and Anthony J. Marcel. Consciousness and the varieties of emotion experience: A theoretical framework. *Psychological Review*, 109(2):219–259, 2002.

Joseph E. LeDoux. *The emotional brain*. Simon & Schuster, New York, 1996.

Marsha C. Lovett and John A. Anderson. History of success and current context in problem solving: Combined influences on operator selection. *Cognitive Psychology*, 31:168–217, 1996.

G. Mandler and S. B. Sarason. A study of anxiety and learning. *Journal of Abnormal and Social Psychology*, 47:166–173, 1952.

G. Matthews. The effects of extraversion and arousal on intelligence test performance. *British Journal of Psychology*, 76:479–493, 1985.

Allen Newell. *Unified theories of cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.

R. Plutchik. *The psychology and biology of emotion*. HarperCollins College Publishers, New York, 1st edition, 1994.

J. A. Russell. Two pan–cultural dimensions of emotion words. *Journal of Personality and Social Psychology*, 45:1281–1288, 1983.

J. A. Russell. Measures of emotion. In *The Measurement of Emotions*, volume 4. Academic Press, New York, 1989.

P. Salovey and J. D. Mayer. Emotional intelligence. *Cognition and Personality*, 9(3):185–211, 1990.

S. Schachter and J. E. Singer. Cognitive, social, and psychological determinants of emotional state. *Psychological Review*, 69:379–399, 1962.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

Herbert A. Simon. Motivational and emotional controls of cognition. *Psychological Review*, 74:29–39, 1967.

Aaron Sloman. Varieties of affect and the CogAff architecture schema. In Colin Johnson, editor, *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, pages 39–48, Heslington, York, England, March 2001. AISB.

Richard S. Sutton and Andrew G. Barto. Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88(2):135–170, 1981.

Amos Tversky and Daniel Kahneman. The framing of decisions and the psychology of choice. *Science*, 211: 453–458, 1981.

R. M. Yerkes and J. D. Dodson. The relation of strength of stimulus to rapidity of habit formation. *Journal of Comparative Neurology and Psychology*, 18:459–482, 1908.

R. Zajonc. Feeling and thinking: Preferences need no inferences. *American Psychologist*, 35:151–175, 1980.

Integration of Psychological Models in the Design of Artificial Creatures

Catherine M. Bryant[⋆‡]
[⋆]Department of Computer Science
University of Exeter
Exeter EX4 4QF, U.K.
catherinebryant@care2.com

Gareth J. F. Jones[⋆†]
[†]School of Computing
Dublin City Universtiy
Glasnevin, Dublin 9, Ireland
gjones@computing.dcu.ie

A. J. Wills[‡]
[‡]School of Psychology
University of Exeter
Exeter EX4 4QG, U.K.
A.J.Wills@exeter.ac.uk

Abstract

Artificial creatures form an increasingly important component of interactive computer games. Examples of such creatures exist which can interact with each other and the game player and learn from their experiences. However, we argue, the design of the underlying architecture and algorithms has to a large extent overlooked knowledge from psychology and cognitive sciences. We explore the integration of observations from studies of motivational systems and emotional behaviour into the design of artificial creatures. An initial implementation of our ideas using the "sim_agent" toolkit illustrates that physiological models can be used as the basis for creatures with animal like behaviour attributes. The current aim of this research is to increase the "realism" of artificial creatures in interactive game-play, but it may have wider implications for the development of AI.

1 Introduction

Over the last few decades Artificial Intelligence (AI) has become more than a philosophical consideration or science fiction plot device. With hardware advances it has become possible to incorporate more powerful AI into games as well as increasingly complex graphics and environments. A recent poll of developers showed a sevenfold increase in CPU time used for AI in the average game since 1997 (Johnson, 2002). A large proportion of this interest in AI is in improving the behaviour of NPCs (non-player characters), making them more believable and engaging. It is important to stress the difference between this 'character-based' AI and that in strategic or turn-based games. Isla and Blumberg (2002) elucidate this in a recent paper:

"These latter categories might be considered attempts to codify and emulate high-level logical human thinking. Character-based AI, on the other hand, is an exercise in creating complete brains. Strategic and logical thinking in this type of work usually takes a back seat to issues of low-level perception, reactive behaviour and motor control....work is often rendered with an eye towards recreating life-like behaviour, and emotion modelling and robustness are often also central issues." (2002, p.1)

Essentially 'character-based' AI is a move away from programming an artificial opponent capable of playing against the human mind in intellectual or strategic games such as chess. Rather than refining specific high-level logical thinking, the aim is to capture life-like behaviour and move towards modelling a complete mind. Thus it aims to populate the game environment with agents who act in a realistic and capable manner. Enemy 'bots' in games such as "Quake" or "Half-life" do not need to understand chess or engage in complex reasoning, but they do need to navigate their environment and know when to attack the player. These virtual 'creatures' should be able to perceive and learn about the environment on their own, make decisions, and in some instances interact with other 'creatures' in a limited way.

The applications for this type of AI are becoming increasingly popular in commercial games, and fairly sophisticated designs are emerging. For example Peter Molyneux's game 'Black and White' included creatures with impressive learning and the potential to develop interesting 'personalities' depending on how the player interacted with them. 'Bots' in games such as the "Quake" series need to navigate a 3D environment realistically as well as try to kill the player without being shot in the process. In later incarnations of similar games, for example "Return to Castle Wolfenstein", the bots also interact with each other and can develop limited team-based plans. However at present knowledge from psychology and cognitive sciences about the processes of the mind appears to a large extent to be under used or overlooked in the design of game AI.

This is clearly an interesting area not just in terms of making better games, but in the development of new AI techniques and algorithms. Laird (2002) argues that computer games provide challenging environments and offer many isolated research problems. As the worlds become more realistic, so too must the behaviour from their characters become more complex. Psychologists, in particular those who have worked on animal cognition, have been studying and detailing the behaviours of autonomous creatures in complex environments far longer than AI researchers have been attempting to model them. Yet many designers of 'virtual creatures' seem unaware of recent developments in psychology and how these might be applied. Emotion provides a good example of one such area of research.

Laird mentions that "emotion may be critical to creating the illusion of human behaviour", but seems at a loss how to go about incorporating this - "Unfortunately, there are no comprehensive computational models of how emotions impact with behaviour. What are the triggers for anger? How does anger impact other behaviours?" (Laird (2002), p.4).

Isla and Blumberg (2002) also discuss the modelling of emotions in character-based AI. They point out that much of the work done so far uses emotion as a "diagnostic channel"; a convenient indicator which can be routed from an internal "emotion" value straight to a facial-expression or visual animation. This value is usually derived from a series of expressions to calculate how 'happy', 'sad' or 'angry' the character is feeling. Isla & Blumberg assert that "emotions clearly play a far larger role in our behaviour ... (they) influence the way that we make decisions, the way we think about and plan for the future and even the way we perceive the world" (2002, p. 4). The general approach of Blumberg and other members of the MIT 'synthetic character research group' is that Game AI should be inspired by work from animal learning and psychology. For example they discuss how the Pavlovian conditioning paradigm can be used, and the importance of the character being able to form predictions about the world. With regard to emotions, they discuss their possible application in "action-selection functions", and making exploratory decisions through a "curiosity emotion". However, they make no reference in this case to work done in psychology.

Emotion is certainly very subjective and personal, and at first seems quite inaccessible to the manipulations and measurements of science. However psychologists have been theorising about emotion for over a century. Since William James first tried to define emotion in his 1884 thesis, research has been done to investigate what emotion is, and more importantly if and how it interacts with the rest of our cognitive system. James himself contended that emotions were nothing more than the feelings which accompany bodily responses to a stimuli. Recent work in cognitive neuroscience provides evidence to the contrary: emotions are linked to brain function, to the point that neural systems of emotion and other mental behaviour are interdependent (Gazzaniga, Ivry and Mangun, 2002). The implications of these results are now finding interest in current work in AI. In this work it is important to focus away from the subjective, conscious 'feelings' of emotion and study the underlying systems which give rise to them and their impact on behaviour. Generally, it seems that these systems are heavily involved in reactive mechanisms and learning, and possibly also decision making and attention.

This paper describes our work towards the development of a basic agent architecture which incorporates motivational and emotional elements derived using ideas and findings from psychology to inform the design. In particular this aims to incorporate some emotional mechanisms that have a deep effect on the decision making process.

The remainder of this paper is organised as follows: Section 2 reviews literature on the psychology of animal motivation, Section 3 outlines work from current developments in artificial intelligence, Section 4 describes our working environment, Section 5 introduces the architecture of our artificial creature agents, Section 6 gives some initial results and finally Section 7 draws conclusions from our current study and considers how the work might be extended.

2 Animal Motivation Theories

In this section we explore some key observations from animal motivation theories and their implications for the design of our model for an artificial creature.

2.1 Miller's equilibrium model and the approach-avoid conflict

Generally speaking, animals react to signals they receive from environmental stimuli. Depending on the nature of the stimulus itself and knowledge of past experience with this type of object, the animal will either approach or avoid it. An approach-avoidance conflict occurs when these signals impel an animal towards these two incompatible forms of action.

Gray (1987) notes that conflict of this kind is extremely common. For animals, it is particularly apparent in their behaviour towards a novel object. Novelty is an important stimulus for both eliciting fear (avoidance) and encouraging exploration (approach). In general, animals appear to avoid extremely novel stimuli, but be attracted to ones which are mildly novel.

Experimental psychologist Neal Miller performed a series of studies on the approach-avoid behaviour of rats. The resulting findings allowed him to develop a model which incorporates the various factors involved.

In Miller's basic experimental situation, a rat is trained to run down an alley to get a food reward. However, every time it reaches the goal, it receives a shock. This sets up a conflict situation. Miller observed that the rat ended up oscillating round an equilibrium 'stopping point' a certain distance from the goalbox. The distance of this point from the goal is defined by the strength of the tendencies to approach and avoid the food. The diagram below shows the factors that affect these tendencies and the resulting decision. Miller's model is represented in Figure 1.

Note that the factors include both internal states of the rat as well as external information from the environment and previous experience. Increasing the hunger or de-

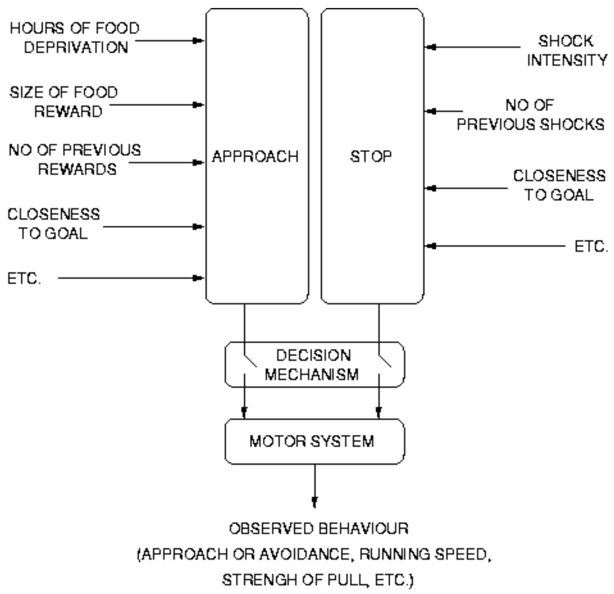

Figure 1: Miller's equilibrium model. (Adapted from Gray (1987), p.142.)

creasing the shock intensity will in turn affect the approach/stop tendencies, and move the equilibrium point closer to the goal. If the approach tendency is much larger than the stop one, you would expect the rat to actually reach the food.

Another point is that 'distance to the goal' is a critical factor in both 'approach' and 'stop' tendencies. However distance cannot affect them in an identical manner: if this was the case then whichever was stronger at the start point would be stronger at the end, resulting in a behaviour where the animal either stops as far as possible from the goal, or completely approaches it.

Work in Miller's (1951, 1959 as cited in Gray 1987) laboratory demonstrated that the strength of the avoidance tendency increases more rapidly with nearness to the goal than that of approach.

Miller noted that there are two main forces behind the tendencies: those that are internal to the animal (such as hunger or other 'drives'), and those relating to the environment and the stimulus itself. They pointed out that there are no internal sources of motivation for the avoidance tendency, and hence it is more purely dependent on environmental factors than the approach tendency. This helps explain why distance has a greater effect on the avoid tendency, especially when near to the goal.

It is clear then that the action towards a certain object is not clear-cut. It is not a simple case of approaching food and avoiding negative objects. Where an animal has learnt to associate pain with an otherwise positive stimulus it may avoid it; conversely if it is hungry enough it will still approach food even if this means receiving a shock.

In terms of programming design, this means that it is wrong to divide the world up into 'good' and 'bad' objects. Instead, every object has the potential to be an overall positive-approach stimulus or a negative-avoid one. It depends not just on the properties of the object, but also what it is associated with and the current internal condition of the animal. This notion of approach-avoid conflicts forms the core of our system design.

2.2 Motivation systems

It is difficult to find one all-inclusive definition of motivation, instead there are various different features which are important to consider.

Firstly, a motivated action differs from a reflex because it is not simply a reaction to an external stimulus. It is also in someway 'driven' by internal states. Teitelbaum (1977, as cited in Toates 1986) argues that "To infer motivation we must break the fixed reflex connection between stimulus and response." Teitelbaum feels that motivation is always directed towards obtaining a certain goal.

Epstein (1982, as cited in Toates 1986) also argues that motivations are complex properties that arise from both external and internal factors. He also considers a third factor: what the animal remembers from past encounters with an incentive object, and the consequence of this encounter.

There are a variety of different models of motivation, of which the simplest is a homeostatic model. Essentially, a homeostatic model is about maintaining essential parameters (e.g. energy level, fluid level) at a near constant 'normal' level. If there is a disturbance then corrective action is taken. Homeostatic mechanisms are driven by 'negative feedback', which can 'switch off' motivation once the deficit has been recovered. The homeostatic model is represented in Figure 2.

According to Grossman (1967, as cited in Toates 1986), there are two types of motivation systems: one which is homeostatic and includes hunger, thirst and other internal factors, while the other is only driven by external factors and includes sex, exploration and aggression.

This dichotomy, however, is too simple, and models developed later do not separate out motivations into these two different types. Homeostatic mechanisms may play a part in explaining the negative-feedback aspects of hunger and thirst, but by themselves are not sufficient as a model. There are other factors to take account of, such as the availability or 'cost' of food - when access to food is made difficult and more energetically costly, animals eat less Toates (1986).

Homeostatic models which look at correcting an energy depletion also do not explain why animals (or indeed people) will overeat if provided with sweet or tasty foods. A final problem is that they do not adequately explain how having a water deficit can then steer an animal towards a water-related goal: in other words they miss the link between the internal state of the animal, and acting towards the external incentives available.

In Bindra's theory (1976, 1978, as cited in Toates 1986), the emphasis is on the role of 'incentive stimuli'

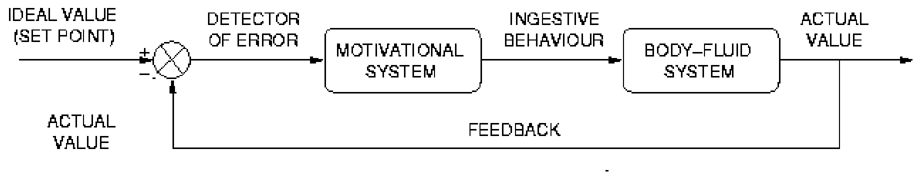

Figure 2: Homeostatic model of motivation. (Adapted from Toates (1986), p.37.)

as well as internal states in the motivation of behaviour. An incentive stimulus is an object or event judged as 'hedonically potent' - one which is affectively positive or negative. This is similar to Miller's approach/avoid tendencies; an animal will react in an appetitive way to hedonically positive incentives, and in an aversive way to negative ones.

Whether a stimulus is seen as hedonically potent depends on various factors, including previous experience with that stimulus as well as physiological states. An animal may assimilate information about a stimulus which it sees as 'neutral'; later on, if the physiological state of the animal changes, that same object could become a positive incentive. For example, an item of food may appear as neutral while the animal is satiated, but once it becomes hungrier that same piece of food becomes a positive incentive which elicits an appetitive reaction.

Bindra develops these ideas into a concept of a 'central motivational state' (c.m.s), which he defines as "a hypothetical set of neural processes that promotes goal-directed actions in relation of particular classes of incentive stimuli" (Bindra, 1974 as cited in Toates 1986).

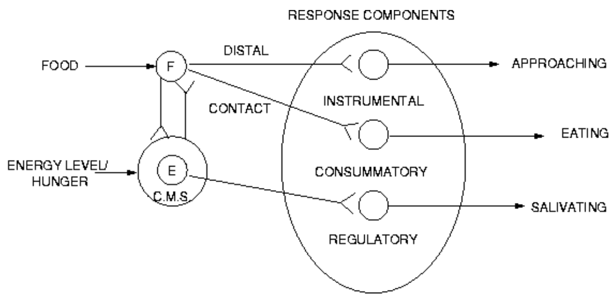

Figure 3: Bindra's model of motivation. The food acts as an incentive stimuli in the feeding motivation system. (Adapted from Toates (1986), p. 43.)

A c.m.s arises from an interaction of 'organismic states' (e.g energy level, testosterone) and the presence of incentive stimuli, see Figure 3. If there are no relevant stimuli present, for example no food when the animal is hungry, then a depletion of energy will not cause systematic goal-directed behaviour. Instead, an increase in general activity may be observed. Also, Toates (1986) notes that novel hedonically neutral stimuli may still arouse some exploration.

In contrast to the homeostatic model, where the internal state drives behaviour, the existence of an incentive stimulus is key. In feeding c.m.s, energy depletion only serves to accentuate the food representation. This explains why tasty and palatable food is sufficient to motivate consumatory behaviour without any kind of energy deprivation.

Thus we can conclude that a homeostatic model is too simplistic for understanding how animals are motivated. All the theories outlined here emphasise a complex interplay between the internal states of the animals with the properties of objects in their external environment. In Bindra's model, an animal cannot just feel motivated to eat because its energy level is depleted - it is only motivated to act in the presence of hedonically potent stimuli. These ideas counter the notion than an animal, once at a certain 'level' of hunger, then sticks rigidly to an explicit goal of 'find food' until its hunger is reduced.

Thus our system needs to include a motivation system which is more flexible than is perhaps usual in existing artificial creatures. The motivation system is a key aspect in that it affects the decision of how the creature should act at each turn in a game.

2.2.1 Toates System theory model of motivation

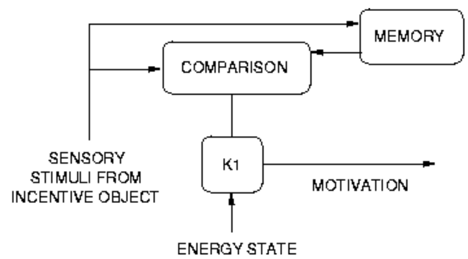

Figure 4: Toates' system theory model. $K1$ represents the energy 'gain' of the system, which determines the level and type of motivation. (Adapted from Toates (1986), p.49.)

Toates (1986) describes his own 'systems theory' model which draws together ideas on motivation similar to Bindra's work. Toates' model is shown in Figure 4. This type of model makes a good bridge from psychological models to computation ones. Toates' model takes account of the three important factors:

- the need for a sensory stimulus to arouse a motivated response.
- the role of the energy level or internal states of the animal in adjusting the 'sensitivity' of the system.

- information from past experiences.

$K1$ represents the 'gain' or sensitivity of the nervous system, and subsequent motivation. If the sensory stimulus 'revives' negative memories of a past experience with this object, it will reduce the value of $K1$. If $K1$ drops to negative numbers this will result in an active avoidance response at the motivation level.

The $K1$ parameter in Toates' model provides a convenient mechanism to encapsulate all the factors involved in motivation in a single number, making the programming of subsequent processes neater. However, it seems likely that there is more to animal motivation systems than described by Toates. Specifically, there is probably a role for emotions, such as fear or pleasure, in motivation and related decision making.

2.3 Emotions

In game AI where emotions have appeared at all, it is generally at a cosmetic level - giving the character the appearance of showing a certain emotion. Here we are concerned not with the subjective feeling or visual appearance of emotions, but rather the underlying mechanisms which give rise to these states.

In this section we review three examples from neuroscience and animal behaviour providing emotional mechanisms that could play a part in the motivation system of our artificial creatures.

2.3.1 Neuroscience and Fear Conditioning

Joseph LeDoux (LeDoux, 1999) identifies two neural routes - one cortical and one subcortical - involved in emotional learning (such as that involved in fear conditioning). The amygdala is a major part of the subcortical route, and removing it prevents fear conditioning from occurring at all. LeDoux suggests that the role of this subcortical route is as a quick-and-dirty reaction mechanism; emotional responses such as fear begin in the amygdala before we even recognise completely what it is we are reacting to.

LeDoux maintains that "Emotion is not just unconscious memory: it exerts a powerful influence on declarative memory and other thought processes." According to Antonio Damasio, one such thought process is that of decision making. He argues that the idea of a totally rational decision maker is not appropriate when quick decisions must be made, and affective memories are invaluable in these cases (Damasio, 1994).

Damasio proposes a "somatic marker hypothesis" which suggests that certain structures in the prefrontal cortex create associations between somatic responses triggered by the amygdala and complex stimuli processed in the cortex. The idea is that both positive and negative associations can be created. Somatic markers help limit the number of possibilities to sort through when making a decision by directing the person away from those associated with negative feelings.

These ideas suggest that not only do affective associations play a part in decision-making, but that there is a physically different route in the brain which processes basic emotional information. In terms of the design of an artificial creature, it would seem sensible to have a similar route, whereby fearful reactions can override more complex processing and steer the animal away from danger.

How do these findings relate to the design of synthetic characters? Firstly, as asserted by LeDoux, whilst consciousness is needed for the subjective feeling of emotion, the basic function of emotional processing and response can be found even in a fruit fly. Thus it seems a possible and useful task to incorporate emotional learning into an AI agent in some way. Since fear conditioning has been extensively studied, it would seem to make a good choice as a place to start. Damasio's hypothesis of 'somatic markers' suggests ways that emotion is important in decision making as well as aspects of learning. It would be interesting to see if basing algorithms around his hypothesis could make for a more 'emotional agent'; one that makes more than completely rational, logical decisions as is generally the case in current game AI. Could this make for a more believable character?

2.3.2 Learning

Toates (1986) notes that when it comes to motivation systems, animals respond to 'primary incentives' (such as food) and 'cues predictive of primary incentives'. In fear-conditioning, animals learn to associate a particular stimulus (e.g. the sound of a bell) with an aversive stimulus such as shock. Once this has occurred, the initial stimulus alone is enough to rouse the animal into a state of fear.

In this way, fear plays a role in animal learning. If a stimulus puts the animal in a state of fear, then its aversive reaction to a subsequent powerful or noisy stimulus is enhanced Toates (1986).

Combined with Damasio's theory, this means that any stimuli occurring while the animal is in a state of fear will be associated more strongly with a negative somatic marker. To replicate this idea, the design of an AI architecture could include a process whereby being in a state of fear affects the strength and type of associations formed by the program.

An advantage of reacting fearfully to cues which predict pain is that the animal will take an appropriate avoidance response before the pain actually occurs.

Gray (1987) explains that rats respond differently in two conditions - receiving a shock, and being exposed to a stimulus that they have learnt predicts a shock occurring. In the first condition, there is a great increase in activity, frantic scampering, or attacking some feature of the environment. In contrast, encountering a stimuli which predicts shock results in the rat freezing. Gray suggests this is an adaptive response that occurs when a rat spots a predator - it freezes in an attempt to avoid detection. He also adds that the response is affected by distance - if the stimulus (or predator) gets too close, the rat shows a strong aversive reaction.

By incorporating fear appropriately into learning and decision mechanisms, an approach to AI could be developed that responds pre-emptively rather than just reactively to pain. Also, the priming effect of fear on forming associations may result in a program which learns to avoid painful situations more efficiently than one with no fear.

2.3.3 The Role of Pleasure

Emotions can also impact animal behaviour to support positive behaviour. For example, there is the concept of a 'positive feedback' priming mechanism that helps to sustain certain activities. Evidence for this was found by McFarland and McFarland (1968, in (Toates, 1986)). They noticed that interrupting doves while they were drinking caused them to 'lose momentum'. This implies that there was something about drinking itself that increased the motivational state of the dove. Toates (1986, p. 116) explains that an animal needs such a positive feedback effect, particularly in situations where simultaneous feeding and drinking tendencies exist of almost identical strength. If it decides to eat and only negative feedback exists, then after the first couple of mouthfuls the feeding motivation will drop, in turn making the drinking tendency stronger. The animal would end up oscillating between food and water, which is costly in terms of time and energy. It would be more advantageous to stick with one activity for a longer period of time before switching.

It would seem vital to have some kind of positive feedback mechanism to reduce the chance of the AI oscillating, and hence to look more believable as well as being more efficient. While the animal motivation literature does not discuss pleasure as such, this concept makes at least a good metaphor for the 'positive feedback' concept. It would make sense that the animal would feel something good when it starts eating or drinking. Essentially, pleasure can be thought of as a reward from an internal, rather than external, origin. Finally, in the same way that the fear emotion might enhance learning about dangerous objects, it would seem a good idea to have a similar 'emotion' which affects the learning about really positive objects or encounters.

3 Artificial Intelligence

In order to make use of the ideas from the previous section, we need to consider what sort of design and framework would be conducive to the incorporation of emotional processes. Despite the lack of sophisticated emotional agents in modern computer games, emotions in general are not a new topic for AI. For example, Simon (1967) had already explored the need to account for 'alarm mechanisms' in artificial systems.

Since the 1980s, many different programs have been specified and sometimes implemented. One of the most notable examples in this area is the work of Sloman (Sloman, 1999)(Sloman, 2000) (Sloman, 2001). He argues for more sophisticated theories of affect and emotion, and has suggested an architecture-based approach to the design of affective agents. This means starting with specifications of architectures for complete agents, and then finding out what sorts of states and processes are supported by those architectures. Sloman himself specifies a multi-level 'CogAff Architecture Schema' (Sloman, 2001) in which 'affective' states and processes "can be defined in terms of the various types of information processing and control states supported by different variants of the architecture, in which different subsets of the architecture are present."

It interesting to note that Sloman has severe objections to Damasio's hypothesis and does not believe that "emotions somehow contribute to intelligence: rather they are a side-effect of mechanisms that are required for other reasons." Despite the debate over emotions and intelligence, Sloman's work is still consistent with that of LeDoux and neuroscience in general. For example, the 'reactive layer' in his architecture which monitors automatic responses is similar to the direct activation of the amygdala from the sensory thalamus e.g in fear conditioning. His 'deliberative', reasoning layer is equivalent to the slower reasoning performed in the cortex. The 'meta management' layer, for monitoring internal states and processes is a little more tricky to pinpoint, however LeDoux (1999) identifies neural systems which may support the awareness of the activity of bodily responses.

Work done by Moffat (2001) 'on the positive value of affect' also draws on psychology to improve AI performance, and provides more inspiration for the relevance of emotion. Moffat feels that cognitive psychologists tend to focus on the function of negative emotions (such as fear), but positive emotions are also important, particularly in learning. On the other hand, machine 'learning classifier systems' (LCSs) model reward and not punishment. 'EMMA', the model resulting from attempts to combine positive and negative affect, was found to learn certain behaviours better than the LCSs. More importantly, Moffat found that the 'emotions' provided a way of signifying importance to EMMA:

"LCSs do not distinguish between stimuli of varying priorities.... EMMA devotes her attention and all her resources to the most important aspect of her current situation. In this respect, emotion is a kind of biological optimiser that could be put to good use in artificial agents too; especially learning ones" (Moffat, 2001), p.61.

Moffat's work suggests the importance of incorporating negative and positive affect. Our work adopts an archictecture-based model as advocated by Sloman. This means rather than trying to code specific behaviours and abilities as they are needed, the starting point is to specify an architecture for a complete agent, and investigate which processes are supported by that architecture.

4 Agent and Game Design

In this section we outline the "sim_agent" toolkit used to implement our prototype agent system, and the design of a simple game framework to explore agent behaviour.

4.1 Programming Environment

The "sim_agent" toolkit developed by the 'Cognition and Affect project' at University of Birmingham, is designed with the specific intention of enabling the building of agent architectures[1]. It runs using the Pop-11 language within the POPLOG environment, on both Linux and Windows systems. Sim_agent was chosen for our work since it allows a wide range of programming techniques, and for the possibility of hybrid systems, for example incorporating neural networks.

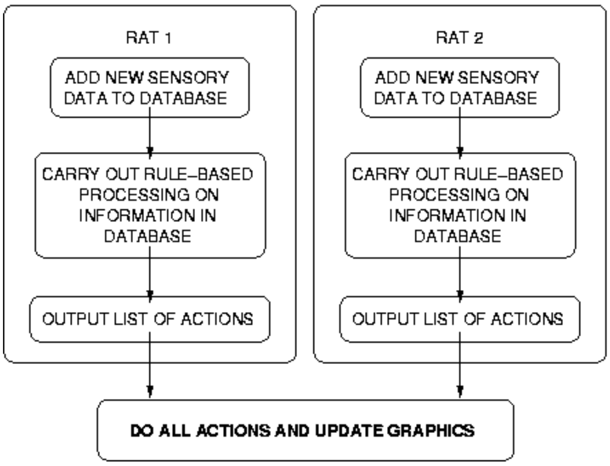

Figure 5: For each 'time-slice', the sim_agent Scheduler runs through processes for each agent. After this is complete, the Scheduler executes any actions, such as moving the agents to a new location, and updates the graphics accordingly.

Figure 5 shows the operation of the sim_agent toolkit. Time is simulated in discrete 'time-slices', which effectively act as a counter. This means that time is not truly continuous, and that the agents all act in a synchronous way. During each time slice, the agent does the following:

- New sensory data is added to the agent's personal database.

- Next, its rulesystem runs, acting on the information available in the agent's database. Unless the agent is going to do nothing during this time-slice, the rulesystem will output one or more 'do X' items into the agent's database.

- The scheduler moves on to any other agents or objects that exist in the environment, and repeats the procedure. When this is finished, it goes back and 'picks up' all the 'do' actions, and executes them.

[1]Details available from: `http://www.cs.bham.ac.uk/~axs/cog\_affect/sim\_agent.html`

4.2 Game Design

A simple game was designed to explore our approach to programming artificial creatures for computer games. This incorporates a set of 'Rat' agents, two sets of 'Rat' agents were designed, one with 'emotional mechanisms' involving fear and pleasure, and the other without. The aim then is to ask participants to play two different versions of the game, taking objective measures of the Rat's performance and a subjective measure of which version the participant thought was more believable.

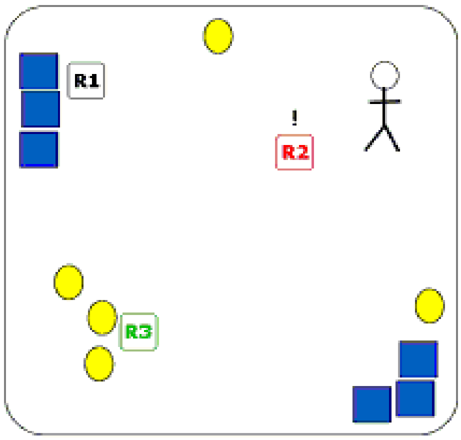

Figure 6: Concept diagram showing typical graphics for the game.

Rats will be implemented in sim_agent, and consist of a 'hunger level', 'thirst level', 'speed', and a 'heading' (direction). The Rat also has a value expressing its current emotional state (fear, pleasure or neutral), and a flag for being in pain or not. 'Food', 'water' and 'person' are all created as objects, of which the game player can move only the food and person.

The idea of the game from the player's point of view is to score points by shocking Rat agents. It uses a turn-based system, whereby the Rats all make an action choice and move, then the player takes a turn.

The aim of the Rats is to basically stay alive, by keeping their hunger and thirst levels relatively low. They have a simple learning system whereby they can form associations between objects which occur together in space, and events that occur together in time. They start off knowing nothing about the player. In other words they have no 'instinctive fear'. Also, the Rats do not immediately understand that a received shock is related to the person - this is something they should learn to associate over time.

Shocking a Rat puts it into a state of pain. In Rat agents with emotions, it also puts them into a state of fear. Both these affect the processing of the Rat during its subsequent turn.

Each turn, the player can move the person within a certain distance, then has the option to shock up to one Rat, if that Rat is 'in range' of the shocking device which the person carries. The Rat cannot discern the direction that the shock came from; instead it decides which object is the most likely 'cause' of the pain, based upon the associations stored in its memory. Note that the range of the shocking device is greater than the visual range of the Rat. This means it is possible to shock the Rat without it seeing the person at all. If the Rat cannot decide where the shock came from it will react differently; perhaps running in a random direction as opposed to freezing or actively avoiding the object it links with causing pain. We hope that this feature will make the Rat appear more believable.

The player also has the option of moving one piece of food around, within a certain distance. This ensures not only a more dynamic environment, but opens up a few more strategies to the player, such as piling all the food together in one place and standing the person next to it.

Rats that feel fear should learn more quickly that the person is associated with pain. This is because being in a state of fear enhances the memory updating and associations involved with pain and objects that might be causing it. Secondly, it is possible for Rats to feel fear at certain objects before they are actually in pain. This should help them pre-empt the shock and hopefully avoid the feared object before it causes pain.

The role of the pleasure emotion is slightly more subtle. It occurs when the Rat starts eating or drinking; to a greater extent the more hungry or thirsty it is. It provides a positive feedback mechanism, which will encourage the Rat to continue consuming until its hunger/thirst level drops quite low. This aim here is to prevent the Rat from 'oscillating' between food and water objects if its hunger and thirst levels are at similar values.

Both emotions are continuous, occurring at certain levels rather than being simply on or off. This allows for some more complex possibilities, such as a situation where the Rat feels a little bit fearful but very hungry; so it approaches the food despite being slightly afraid of it.

While we have a complete design of the architecture for the game, its implementation is incomplete. The system currently does not incorporate interaction with a user, and the memory and emotion systems are not yet functional.

5 Architecture Overview

5.1 Basic Framework

Figure 7 shows the architecture of the Rat agent. The currently implemented basic design is shaded grey. This includes the core decision-making aspect, and the motivation systems. Running from top to the bottom is roughly equivalent to the order of the sim_agent rulesystem run by each agent during the cycle.

Perceptual system This identifies what the object is, along with other properties such as how far away it is, how much there is, and in the case of food/drink a 'hedonic' value representing how 'tasty' or desirable it is. Any information about objects recognised as food will be passed on to the feeding motivation system, and the details of drink objects filtered to the drinking motivation system. At this stage any other objects, such as Rats or perhaps the human player are not processed further.

Motivation systems Here a value for each object is calculated. The value represents an overall 'weight' of importance. It takes account of the properties of the individual item, and how far away it is, along with specific information on the internal condition of the Rat. The Feeding motivation system uses the Rat's hunger value, while the drinking systems uses the thirst value. (Hunger does not affect the drinking motivation system.) An equation for this is as follows:

$$\text{Weight} = \frac{a \times \text{Hunger} + b \times \text{Amount}}{c \times \text{Distance}} + d \times \text{Hedonic Value}$$

A weight value is computed for each object, along with an appropriate action. If the weight value is positive, then the action will be to approach the object; if it is negative then the suggestion will be to move away from it (particularly unpleasant food i.e. with a large negative hedonic value, might be aversive). If the Rat is currently consuming the object, the weight will represent how important it is to carry on doing so.

Finally, if there are no food objects going into the feeding system, it will output an 'explore' action, with a weight evaluated using the Rat's current hunger level as the main variable.

Decision The decision mechanism simply chooses whichever action has the highest 'weight' associated with it. However, it could be more complex than this - taking account of what other objects lie in the same direction. So a good decision might be to go towards a mediocre item of food if there also happens to be some water nearby. Conversely, if a great item of food is very close to a dangerous object it might be better to avoid that direction.

Motor system The processes here figure out how far the Rat can move in the chosen direction, and evaluates the new co-ordinates to be put out as a movement action.

5.2 Full Version

The Full Architecture design shown in Figure 7 includes two important additions to the basic version: memory and emotion systems.

Memory This stores locations of objects which the Rat encounters, and includes a simple learning mechanism

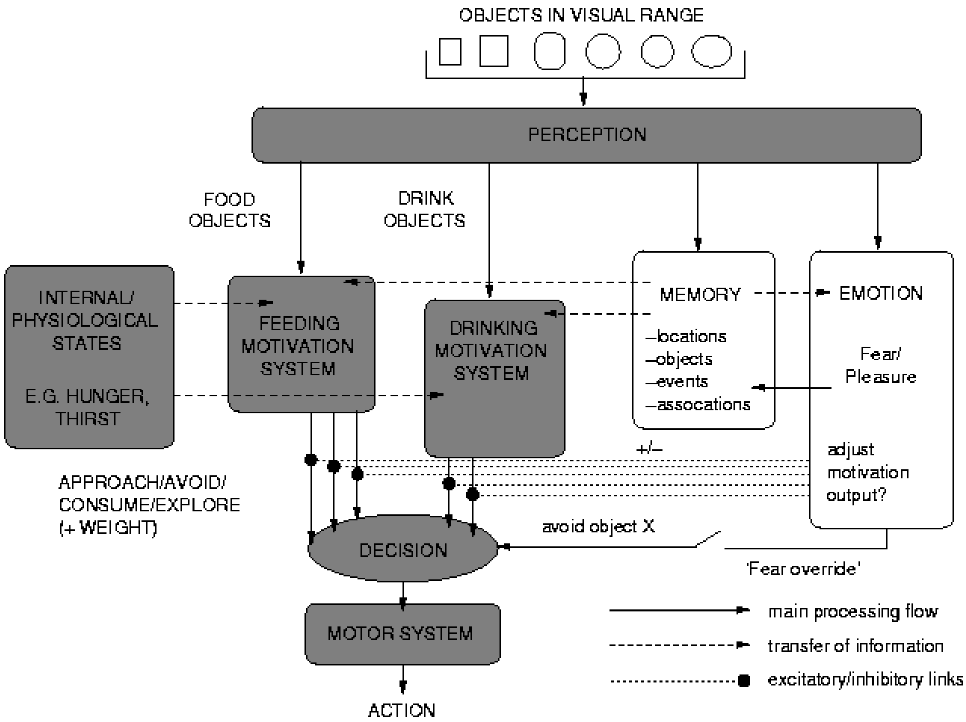

Figure 7: Rat Agent architecture design. The implemented base design is shaded grey.

which can develop conditional associations between objects and events which occur together in space or time, in additional to unconditional ones arising from the unconditional stimulus of the object. It provides extra detail to the motivation systems, so their evaluation equation can also take account of any past experience with the object.

The object memory does not remember food items as 'specific' e.g 'food item one', but instead stores food by location e.g 'food at (x,y)'.

Emotion This does several things, but all the actions essentially involve fear and pleasure. Firstly, it cross references incoming visual information with details in the memory to see if any objects should elicit a state of fear, and if so then what level of fear. The level relates directly to the strength of the association between that object and being in pain.

In terms of pleasure, at the moment it only produces this state if the Rat is actually consuming, however this could be extended to an anticipative pleasure. The level of pleasure is determined by how hungry the Rat is. So if it is really hungry before it starts eating, the level of pleasure will be high. In a sense 'pleasure' here can also be thought of as 'relief'.

The emotion system can adjust the weight values produced by the motivation systems to enhance or reduce particular signals. As an example, if one of the food objects is associated with something nasty the feeding motivation system may output a negative 'avoid' signal for that object. If it is particularly nasty - enough to cause some degree of fear - the emotion system will enhance the signal, making it particularly aversive, while decreasing the strength of all the other signals.

It is important to note that in this situation the emotion system does not necessarily get the last word - if the rat is especially thirsty, one of the 'approach water object' signals might still be greater than the avoidance one. However, if recognising an object pushes fear above a certain threshold, an override happens; the rat will run from that object despite how hungry or thirsty it might be. This route is approximately similar to the 'quick and dirty' fear reaction mechanism discussed by neuroscientists.

If the Rat is feeling pleasure at consuming an object, the emotion system will also adjust the weights, increasing the consume signal while decreasing the others. The amount that the signals are altered will relate directly to the level of emotion - a higher level resulting in a greater signal adjustment.

Feeling either emotion to any level will also feed back into the memory system, enhancing specific associations formed or reinforced during that cycle. In particular if the Rat was in a state of fear because it could see the player, and then subsequently experienced a painful shock, the association between the player and pain would be strengthened to a greater level than if the Rat was in a neutral emotional condition.

5.3 System Implementation Details

While it is often comparatively easy to specify the desired features and behaviour of a system, actually encoding these into a working agent is often much more difficult. In this section we discuss our current implementation of the systems within the Rat and how these might be extended.

5.3.1 Hunger/Thirst Systems

After some consideration the following relationship was used to calculate the hunger and thirst values in each cycle.

$$Y = \frac{2x-1}{(2x-1)^2+1} + 0.5$$

where Y is the new Hunger or Thirst value and x represents a counter which increments each cycle. It is a fairly arbitary choice, and could be replaced with an equation (or indeed series of equations) which more accurately reflect how hunger changes in a real animal.

This function was chosen since it increases slowly, indicating that the Rat's hunger/thirst level rises slowly at first, but then increases rapidly to a point where it is 'very hungry', with the limiting value $Y = 1.0$ leading to death of the Rat from starvation. This function is not taken from any particular animal psychology literature, but is based on intuition of the relationship between hunger/thirst and time. During each run of the Rat agent the hunger and thirst levels are updated.

It would be good if food of a higher 'quality' actually reduced their hunger by more - in other words there would be some real benefit in going for these type of objects. This is one of the many ideas which could relatively easily be added into the program in the future.

5.3.2 Motivation Systems

The feeding and drinking systems are identical, and we describe only the feeding system here.

The purpose of the motivation system is to process the relevant visual information and output a database entry for each object determining the most appropriate action.

'Food Weight' FW is calculated using the following equation,

$$\text{Food Weight} \propto \frac{H^2}{D} + FQ$$

where H is the hunger, D is the distance to the food, and FQ the food quality. The hunger value is squared so that the resulting weight is exponentially greater at high levels of hunger. FW is proportional to $1/D$, resulting in lower weights with greater distances between the Rat the the food. Food quality is added to the end to provide a final adjustment. If it is negative, it may push the resulting weight to negative values and a subsequent 'avoid' action. The constants in the equation were derived from trial-and-error testing until the Rats behaved in a reasonably balanced way.

At the end of the day, the motivation equation is key to the decisions made by the Rat, and behaviour may be further improved by use of alternative functions. Another option would be to use a genetic algorithm approach to try to 'evolve' an optimal equation that produces the most 'fit' Rats. Fitness could be simply a survival rate, or relate to how well the Rat maintains a balanced level of hunger and thirst.

5.3.3 Explore System

If there is no visual data on food objects available, the system outputs an explore action.

The 'Explore Weight' EW is calculated using the following method,

$$EW \propto \text{need}^2$$

where need is the current hunger level of the Rat.

Again, this is another equation that could benefit from being 'evolved' by genetic algorithms. At the moment it is roughly balanced so as to become more urgent to find food the hungrier the Rat becomes, but at lower levels of hunger it's still better to carry on drinking if drink is available.

Again the exploratory mechanism is not based on psychological literature, but in its current intuitive form merely ensures that the Rat moves to locate sources of food and drink. There is considerable existing work on animal foraging patterns that could be applied here.

The following exploration method was developed using trial and error experimentation. The explore action has the potential to span up to 6 turns, during which the Rat does the following:

Turn	Action	Count
1	Choose random direction X, move that way.	1
2	Continue to move in X direction	2
3	Continue to move in X direction	3
4	Reverse direction X, move that way	4
5	Continue to move in (reversed) X direction	5
6	Continue to move in (reversed) X direction	6
7	Back in starting position, choose direction Y	1

This means that the Rat spends 3 turns moving in one direction, at which point it turns round and goes back to the starting position. If in any of these turns it encounters food/drink then it reacts to those objects: in other words it is not 'committed' to completing the exploration sequence.

When it comes to step 6, a new angle for exploration is chosen. Essentially the new angle cannot be anywhere within the range of the old one, plus or minus 45 degrees. This makes sure that after an unsuccessful exploration in one direction, the Rat chooses a significantly different direction to explore in next.

6 Results

Figure 8 shows a series of images showing the progress of Rat agents. The frame number refers to the time-slice at which the snapshot was taken. The rats are the square

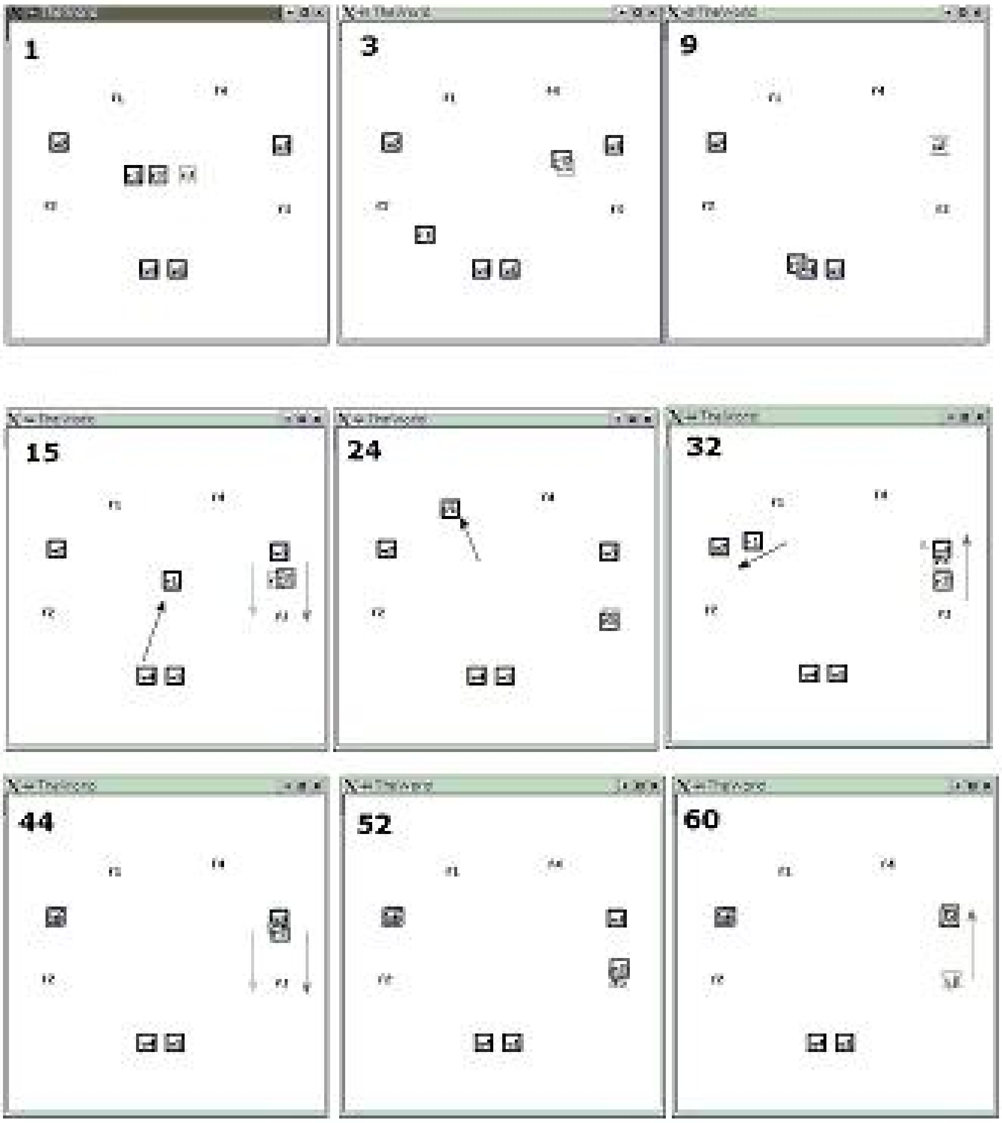

Figure 8: Images showing an example of Rat agent progress.

boxes in the centre of frame (1). They all start off with hunger and thirst at the same low level in all of the results discussed. This is probably why r2 and r3 head towards the same water object (the square boxes) at the beginning.

At cycle 3 R1 can be seen to be fairly close to both a food item (the circles) and a water item. The water item it heads towards has the highest 'quality' value of the objects in the environment, so this move makes sense. After drinking for a bit, the food motivation system pushes him to explore (about cycle 12/13). He finds food and consumes this for about 5 cycles then makes his way to the nearby water object. At this point though he hits a bug whereby no matter how much he drinks the thirst does not go down. By cycle 52 he is dead from hunger.

Rats R2 and R3 essentially oscillate between the food and water objects on the far right.

While the motivation equations could do with some adjusting, it is still good to see that the agents make some attempt to keep their hunger and thirst levels low. Also it is good to see the inefficient oscillating behaviour occurring as predicted. Including the 'pleasure emotion' may really help to reduce this.

7 Discussion

Although the implementation of the Rat agent architecture is not complete, some conclusions can be drawn at this stage. The work completed so far is very promising, and we are confident that developing it further would result in some very interesting results.

Firstly, there is a wealth of psychology literature which makes for good source material and inspiration. The work described here focuses on motivation and emotion systems. However, there is much more information and theory available than has been incorporated in this design.

Many of the ideas described in this literature are not ones that are typically explored when considering problems purely from an AI point of view.

The biggest advantage of considering animal motivation studies is that the researchers spent a lot of time observing and testing the animals, and really getting to grips with the basic systems that drive and affect behaviour. Regardless of whether their findings accurately explain how the animal mind really works, their descriptions still relate strongly to real observable actions. The resulting models and diagrams make it fairly simple to port the ideas over to a computing environment.

It is encouraging to see that even the basic version resulted in agents that made appropriate decisions to reduce their hunger and thirst levels. Their behaviour was always slightly unpredictable (and hence, perhaps more believable?) since they never followed a 'set path' or 'set procedure'. It was not a case of 'when hunger is X, find food'.

The architecture-based approach lends itself well to the approach taken in this work. Once the basic architecture design was in place, other aspects could integrated in quite a natural way. For example, once the base motivation system was in place, it was fairly straightforward to see how the fear emotion could be incorporated and affect the decision making.

In the games industry, it is becoming more common to use pre-designed 'engines' to cover whole aspects of the coding. These engines tend to be specialised, for example it is possible to get physics engines that deal specifically with car crashes. Considering how complex just designing the motivation, or learning, or perceptual system can be it would seem a good idea to put them together as an AI creature 'engine'. This could be the basic all-purpose agent which could then be tweaked and adapted by the specific game designers to suit their needs. Using a system like sim_agent would be perfect for this, since it is easy to adjust old rulesets, add in new ones, or simply change the base variables for the agent instance. The architecture and design ideas presented here could form a component of such an engine.

To really achieve this effectively, it may be necessary to bring together a hybrid of AI techniques. In this investigation we saw how difficult it is to know what functions to use to provide the most efficient and realistic behaviour. This is exactly the type of problem that genetic algorithms could help with. Neural networks, or at least a connectionist approach, seem like the best strategy for implementing learning systems. However, without being implemented in a way that makes them useable inside the symbolic environment of game code they are not too practical. Both these areas would provide good grounds for further study.

Overall, we are encouraged by our results. They demonstrate that psychology literature is a very fruitful resource. If a complete AI engine which has been inspired by psychology in is developed, we feel that it would indeed create more believable agents and much more immersive game play.

References

G. Colombetti. The Somato-Cognitive System. Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing, University of York, pp21-28, 2001.

A. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain*, Putnam, 1994.

M. Gazzaniga, R. Ivry and G. Mangun *Cognitive Neuroscience*, W.W. Norton & Company, Inc., 2002.

J. Gray. *The Prychology of Fear and Stress*, (2nd edition) Cambridge, 1987.

D. Isla and B. Blumberg. New Challenges for Character-Based AI for Games. Proceedings of the AAAI Spring Symposium on AI and Interactive Entertainment, Stanford, CA, 2002.

S. Johnson. Wild Things. *Wired* (issue 10.03), 2002.

J.Laird. Design Goals for Autonomous Synthetic Characters. Retrieved from `http://ai.eecs.umich.edu/people/laird/papers/AAAI-SS00.pdf`.

J. LeDoux. *The Emotional Brain*, Phoenix, 1999.

D. Moffat. On the Positive Value of Affect. Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing, University of York, pp58-62, 2001.

H. A. Simon. Motivational and Emotional Controls of Cognition. In *Models of Thought*, pp29-38. Yale University Press, New Haven, 1967.

A. Sloman. What Sort of Architecture is Required for a Human-Like Agent? In M. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, pp35-52, Kluwer Academic, 1999.

A. Sloman. Architectural Requirments for Human-Like Agents Both Natural and Artificial (what sorts of machines can love?). In K. Dautenhahn, editor, *Human Cognition And Social Agent Technology, Advances in Consciousnes Research*, pp163-195, John Benjamins, 2000.

A. Sloman. Varieties of Affect and the CogAff Architecture Schema. Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing, University of York, pp39-38, 2001.

F. Toates. *Motivational Systems*, Cambridge University Press, 1986.

Why do anything?

Emotion, affect and the fitness function underlying behaviour and thought.

Darryl N. Davis

Computer Science, University of Hull

Kingston-upon-Hull, HU6 7RX, UK

D.N.Davis@dcs.hull.ac.uk

Abstract

What is mind? A straightforward answer is that which decides what to do next. How does a mind decide what to do next? A straightforward answer is by processing, acting and sensing. What does a mind sense? Everything? What is processed? Everything? How is everything processed? In every possible way? What actions are selected? Every action? Ten simple questions and two straightforward if rather mischievous answers. In this article differences in the nature and requirements of biological and synthetic minds is investigated in terms of control: control over what is sensed; control over how that is perceived; control over how those perceptions are processed; and control over how this epistemic flow leads to control over actions. No straightforward answers to any of the questions posed are presented. Rather, different perspectives on how investigations into these questions are used to present the thesis that some means of valencing the mind is necessary. In short this article considers how the economics of thought and action reside in the currency of affect.

1 Introduction

Control of behaviour is vital to animate biological systems. Mistakes in such systems lead to at best ineffective use of possibly scarce resources; at worst such mistakes lead to injury and death. Consider the scope of biological systems from solitary insects, insect communities through to vertebrates, mammals and primates. Many insects simply act out genetically determined behaviours, with the species surviving due to sheer number of individuals. The more sophisticated the biological system becomes, the more scope there is for adaptation, learning and error. The more sophisticated the biological system becomes the greater the range and diversity of type of drives that need to be fulfilled. Yet in every case the control mechanism within the biological system, whether cricket, ant, lizard, anteater, leopard or chimpanzee, needs, in some sense, to make a decision about what to do next. With increasing sophistication of biological system comes an increasing degree of autonomy. With the increasing degree of autonomy comes flexibility, the possibility of behaviour adaptation and learning. With the increased behavioural flexibility comes greater choice and a greater range of potential error. Without the increased behavioural flexibility, the range of behaviours triggered by any situation is more constrained, limited and sharply defined in terms of their effectiveness. The symbiotic nature of organism-niche evolution has determined (and continues to determine) the environmental scope of any given organism. The effectiveness of the evolved control mechanism(s) is self-evident in the diversity of biological organisms across individual and the many different environments. An important question for the designer of synthetic systems is whether there are levels of abstraction across these biological control mechanisms useful in the design of artificial systems. A further question is what types of commonality are there across the control mechanisms in these different biological systems? Salient answers to these and related questions will do more than simply provide useful insight into the design of artificial systems. It is within such a framework that the recent growth in research giving artificial systems emotional capabilities or qualities is questioned (Davis and Lewis 2004). This framework may provide the means by which we can advance our understanding of the phenomena that is affect (Sloman et al 2004).

This article makes a case for developing this framework in terms of affect, motivation and other control states, plus an analysis of niche and design space in terms of complexity of information processing structures. It then places recent investigations into affect and affordance within ongoing research into the development of architectures for synthetic intelligence. In these developing computational systems, activity and behaviour at one level is represented and controlled at other layers. The primary conjecture is that the design and implementation of such architectures can proceed using a systematic control language that obliviates the need for ad hoc heuristics to direct the processing within an intelligent system. This control language is grounded in affect. The aim is to try and develop a control language that is consistent across different domains, tasks and levels of processing. If and where this attempt to achieve this objective fails, the result will be a deeper understanding of the nature of the control systems necessary for synthetic (and natural) mind. The computational work is being developed with no explicit requirement for emotion but rather a reliance on affect (a valencing of and within internal processes), affordance and motivational constructs that together can be used to guide both internal and external acts.

2 Emotion, Affect and Theories of Mind

The philosophical foundations of cognitive science rest on a number of assumptions. One very important one is that cognition is a natural kind (Fodor 1983, Pylyshyn 1984). It has been suggested that emotion too is natural kind (Charland 1995). In effect to understand how human (and similar) minds work, to develop theories about mind and to build computational systems capable of simulating (human) mind they should include both cognitive and affective mechanisms. Counter arguments to this latter claim do exist (Griffiths 2002). The argumentation for the counter claim bears similarities to that to be found in Sloman's research (2001, 2004a, 2004b).

There is a growing consensus among theorists and designers of complete intelligent systems (Minsky 1987, Sloman 2001, Franklin 2001) that synthetic minds, to be complete and believable, require a computational equivalent to emotion to complement their behavioural and cognitive capabilities. This need not be a deep model as the thesis behind the work on the OZ project (broad and shallow) demonstrates (Bates et al 1991, Reilly and Bates 1993). This requirement has been highlighted by earlier prominent researchers (Simon 1967, Norman 1980) in their discussions on the nature of cognition in biological systems (typically humans).

Over the history of psychology, emotion has attracted attention. Hebb (1946) for example could not provide an adequate explanation for observed primate behaviour without the incorporation of emotion. There is no theory of emotion that is consistent across the many competing theory types. Most pointedly with regard to the arguments presented here, it is not clear what level of neural sophistication is required to experience emotive qualities. So, while the need for emotion in theories of human (primate) mind is not disputed, what emotion actually is and the processes and mechanisms that give rise and support its function are still very much open to debate.

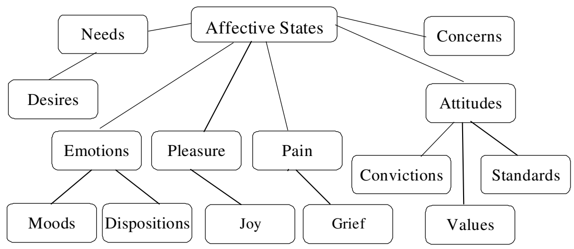

Figure 1. An incomplete taxonomy of affective states.

The emotions are but one type of affect among the various classes of sometimes fuzzily differentiated control states associated with mind (Simon 1967, Sloman et al 2004). Previous research has attempted to organize motivational control states in an initial taxonomy as a starting point for future research (Davis 2001b). A similar (fuzzy and very incomplete) taxonomy for affective states is shown in figure 1. The ones shown in figure 1 have been addressed, albeit in a relatively shallow manner, at a theoretical, design or computational level in earlier research (Davis 1996, 2001a, 2001b, 2002, Davis and Lewis 2003, 2004). This taxonomy and the type of categorisations made through the rest of this article are wider in scope to the conceptual analysis of emotion made in for example (Ortony et al 1992), albeit at a relatively shallow level. Section 5 of this article provides further analysis of the affective categories associated with needs and desires in terms of motivational control states.

Theories of emotion can be typified as belonging in one of several types, for example physiological (James 1884; Plutchik 1994), evolutionary (Darwin 1892), expression (Ekman 1994), appraisal (Scherer 2001) or goal based (Oatley 1992). This is partially due to different programmatic objectives within, for example, neurophysiology, psychology, philosophy and cognitive science. If a software engineer were to use many of these theories of emotion as the starting point for a specification of emotion in a synthetic computational system, a number of very obvious comments would be expected. One there is no consistency across these theories. Two, some of the earlier but still prominent theories are internally inconsistent. Third, most of the theories are so loosely defined that they do not provide for a suitable specification for a computational mind. As Sloman

(Sloman et al 2004) points out, this is to be expected with any developing scientific theory.

Duffy (1962) considers the use of the fuzzy, ambiguous and misleading term "emotion" as fundamentally flawed. Such terms should be abandoned as confusing and new or clearly delineated terms used only where such concepts are clearly and unmistakably identified. There is such a volume of research in this area that a significant academic revolution would be required to pursue such a path with any success. While this may be true of disciplines that study human intelligence, the same does not hold for artificial systems. However there are many types of artificial system and there are quite legitimate and necessary reasons why a model of emotion (albeit shallow) may be required within these systems (see Sloman et al 2004). The research paradigms of artificial intelligence, cognitive science, computer science and psychology overlap and any purported boundaries are somewhat arbitrary. The question addressed here is not to dispute the importance of emotion for human mind, nor its study in psychology and cognitive science, but to dispute its necessity in the design (and implementation) of intelligent synthetic systems.

Numerous prominent researchers into intelligent systems have suggested that affect-like mechanisms are necessary for intelligence (Simon 1967; Norman 1980; Minsky 1987) or will arise out of the interaction of the processes necessary for intelligent behaviour (Sloman and Croucher 1987). More recently, Sloman (Sloman 2001) has suggested that while emotion is associated with intelligent behaviour, it may not be a prerequisite. If that is the case and that emotion is a side-effect of mechanisms in sophisticated and complex biological architectures, intelligence is now tightly bound to the control of these side-effects through evolution. The development of control mechanisms to harness and cope with the affective associations of the mechanisms necessary for intelligence, over the diachronic intervals associated with evolution, is such that in effect emotion and affect are now central to intelligence in biological systems.

3 A Requirement for Affect?

Norman's pivotal paper (Norman 1980) suggested emotion-like processes are necessary for artificially intelligent systems. This section builds an argument that denies the need for emotion in many synthetic systems, while accepting that notable systems have been built based on models of emotion using a diverse range of computational mechanisms (Adamatzky 2003; Elliot 1992, Frijda and Swagerman 1987, Ortony et al. 1988; Riley and Bates 1991, Scherer 1993, Velasquez 1996, Wehrle, 1994).

Griffiths (2002) suggest that there are different kinds of emotion or emotional process. This is different to the claim that there are basic emotions, for example (Ekman 1994), and more sophisticated emotions that combine the basic emotions with higher level (neocortical) processes. Broadening the scope to include other affective states highlights the diverse nature of these phenomena. There are many potential types (and labels) for the range of affective states. For example my office thesaurus lists twenty-seven synonyms for pleasure (and two antonyms). A trace through the thesaurus following up all antonyms and synonyms will quickly produce an extensive list of affective terms. It would take the remainder of this paper just to provide linguistic definitions. Highlighting the full extent of the possible relations between them (as in for example a plausible dimension of affect that includes pain, distress, sorrow, torment, grief etc.) is not possible here. These states differ broadly in their situational context, their duration and their possible effects. A complete theory of affect should be able to provide a coherent structure across these issues. It should also provide an account for these in terms of precursors, initiating events, supporting processes, individual and situational differences etc.

There is also the question of what level of control structure sophistication is required for any of these states. It does not make (much or any) sense to discuss how an insect, for example an ant, can grieve over the loss of fellow ants. Why therefore should it make more sense to discuss how a synthetic intelligence, possibly of similar information processing complexity as an ant, can experience affective states qualitatively similar, in type, to grief? It is as yet unclear where it is even sensible to associate the concept of pain with such an organism. The folk psychology of affect is less strict in the application of such terms; for example, a mother may chide her son for "*tormenting*" the ant's nest. Progress in understanding affect in terms of the information processing complexity of the behavioral control systems of the organism is required if any effort at modeling affective states in synthetic systems is to be something more than silicon folk psychology.

There are many questions that research into the emotions and affect needs to address. Are all of the possible affective states appropriate to computational modeling? If not, which are plausible and why? For example how can a machine experience joy? Wright and colleagues (1996) used the CogAff architecture as the basis for an account of grief, but they do not imply that their computational designs would be capable of suffering so. Are there categories of affect that are needed if the theory of affect (and hence emotion) is to progress? For example, is joy is akin to pleasure, in the same way that grief is akin to pain? Cognitive systems that attempt to model human functioning and cognate theories need to explain how these are alike and the different levels of abstraction over the affective states. Such mind models are qualitatively different to the (insect or at best perhaps pigeon level) systems currently being developed by AI practitioners. Do the decision and arbitration functions and processes required in these latter systems really require the conflict resolution processes to validate their choices in terms of a shallow and sometimes arbitrary use of affect. Do emotive recognisers in sophisticated interfaces require any more than the coarsest granularity in their discrimination of the possible affective state of the user?

4 Niches, Designs and Affect

Using the running definition as mind as a control system that decides what to do next, we now visit some alternative designs. The framework used, even if at a relatively shallow level of analysis, is the idea of niche and design space (Sloman 1995, 2001; Sloman et al 2004). Figure 2 provides a simple exemplar of alternative environmental niches, defined in terms of altitude and aquaticity, and designs for life that inhabit them.

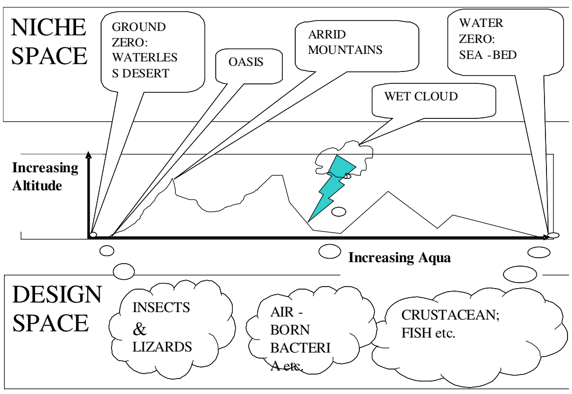

Figure 2. Environmental niche space and associated designs.

A different type of niche can be specified in terms of the resource and task requirements for any organism. The suggestion is that different categories of affect are associated with different levels of complexity in the structures and processes that support different classes of mind. Animal psychology and comparative ethology can help here in identifying the broad categories of mind (Davey 1989, Gibson and Ingold 1993, McFarland 1993, Toates 1998). Rolls (1999) provides four broad categories of brain complexity: mechanisms for taxes (for example reward and punishment); mechanisms capable of stimulus response learning via taxes; mechanisms capable of stimulus reinforcement association learning and two-factor learning; and finally explicit systems that guide behaviour through syntactic operations on semantically grounded symbols. A similar continuum, in niche space, for conceptualising the increasing sophistication of mind is presented here. Along this continuum thresholds can be placed for stating the “mind” has: mechanisms of adaptation; mechanism capable of learning via the equivalent of classical conditioning; mechanism capable of learning via operant conditioning; mechanisms allowing tool use; mechanisms for tool manufacture; map formation; and the use symbols. Figure 3 shows these niche spaces and in the associated design space, examples of architectures from the animal kingdom.

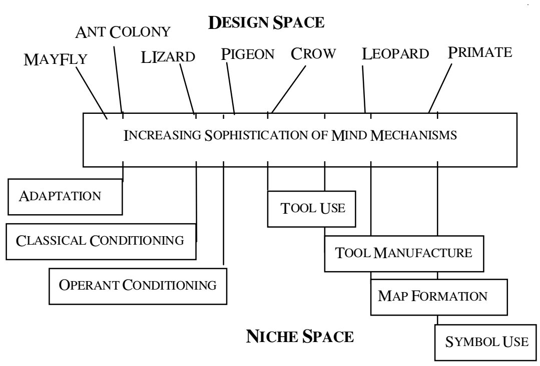

Figure 3. A tentative niche space of increasing sophistication with associated design examples.

From an evolutionary and anatomical perspective, there is some commonality in the mechanisms running across the dimension of figure 3. For example the chemical (hormone) and amygdala routes to behaviour in the description of the dual routes to behaviour (Rolls 1999). However while the organisms to the right of figure 3 may share the use of hard-wired finite state automata-like mechanisms, for example compare the fly-tongue reflex of the frog with the knee-jerk reflex of humans, the capabilities of the organisms to the right of figure 3 far surpass those to the left (mayflies and grasshoppers do not perform behavior rehearsal for example!).

Even a relatively trivial analysis of the opportunities offered by this perspective, shows how difficult this task is. The continuum of increasing sophistication of behaviour (and presumably mind mechanism) is neither discrete nor linear; perhaps dimension will be a better term than continuum. Consider the case of two very different organisms such as the crow and the leopard. At first it would be tempting to unequivocally suggest that the information processing requirements associated with the adaptive hunting strategies and rearing capabilities of the leopard far outstrip the information processing requirements of the crow. Yet as recent experimental evidence suggests (Weir et al 2002), the crow is capable of innovative tool creation in feeding itself, yet the leopard uses no recognisable tool. Does this place to the crow to the right of the leopard in the design space of figure 3? No! - At least not wholly to the right. The crow's behaviour while interesting is an adaptation of its natural tool making activity to support food foraging. The leopard however does use tools, for example sound, in the modifications that it can make to its hunting tactics. Typically, while stalking at night, a hunting leopard, close to a herd of prey, will typically move with retracted claws and with sometimes very slow and deliberated movement (for example fifteen metres over two hours). However it can modify this almost silent hunt, and deliberately create sound, with a pounding paw, to agitate and disorientate gazelle herds. In raising their offspring, the crow will not dwell over the loss of a brood. The leopard on the other does appear to dwell over the loss of her cubs. In short, in moving across the range of warm-blooded animals from for instance pigeons there is an information processing complexity change in moving to mammals. At that point up to the more advanced primates (for example the orang-utan) there are genera and species level partial advantages, related to fulfilling or taking advantage of specific niches and environments. The theory of affect would benefit if a similar conceptualisation as produced by ethologists were produced for affect.

5 Needs, Desires and Motivations

The previous section provided a tentative look at taxonomy of control mechanisms, the degree of task complexity and diversity of task repertoire. Here we look behind the behaviours to see the motivational mechanisms responsible for the behaviours. This builds on earlier work (Sloman 1990, Beaudoin 1994, Davis 2001b) on motivators. This differentiation between emotional (affective) and motivational control states is not new (Simon 1967). Here, however, previous analyses are revisited in terms of furthering the aims of the tentative analysis of affective states given in section 2.

At a very coarse grain we can differentiate between primal needs required to maintain the life-force of an individual organisms, the requirements of the species and the requirements arising from social interaction. For example, Aubé (2004) in his analysis of needs in nurturing species, differentiates between primal needs, that are related to the resource requirements of an individual organism, and second order resource requirements, that are related to requirements arising and made available through activities such as social bonding and collaborative behaviours. Aubé suggests that the affective states associated with these requirements differ too; he terms these commitments.

An alternative (and perhaps complementary) approach is to look to develop the taxonomy of primary reinforcers that Rolls (1998:table10.1) provides. That taxonomy is differentiated primarily in terms of five sensory modalities, reproduction and a collection of diverse reinforcers related to social and environmental interactions. The relevance is that these reinforcers, either positive or negative, are mapped onto drives and affective states. In the somatosensory modality for example pain is a negative reinforcer, while touch positive. Control over action is a positive reinforcer

In accordance with earlier research (Davis 2003) needs are manifested in processing terms as drives. Drives are low-level, ecological, physiological and typically pre- or non-conscious. They provide the basis for an agent's behaviour in the world, are periodic but short-lived and are defined in terms of resources essential for an agent. Such activities for information agents include the need to gather resources and propagate information to associates in their society. In biological agents such drives include thirst, hunger, and reproduction. Nurturing sublimates some of these drives in the service of others. Thresholds for the onset and satiation of such drives are variable and dependent upon processes internal to an agent and external factors arising through the agent's interaction with its environment. We can model such drives relatively easily in computational agents using intrinsic control structures. Prior work (Davis 2003) used fuzzy logic models to do just that.

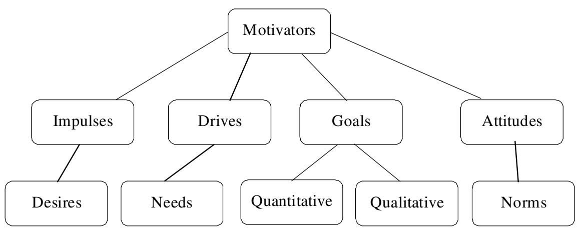

Figure 4. Taxonomy of Motivational States

Having established a primal motivational category, we can now look further at the types of taxonomy produced for motivational control states (Davis 2001b). Figure 4 provides four major types with, in each case, subcategories. In keeping with the theme of tentative dimensions for control states that is being used throughout this article, there is an implied ordering from left to right across figure 4. The processing requirements and representational qualities associated with these four broad categories become more sophisticated towards the right of the figure.

Impulses are related to spontaneous behavior, for example suddenly leaving the cinema during the screening of a film or making a rash purchase. They are associated with the instantaneous formation of an idea, perhaps unrelated to current cognitive context, and can cause a temporary or more persistent re-focus of mind. Here *Desires* are only partly analogous to their use in BDI agent architectures (Georgeff and Lansky 1987), for example *desires(agent, stateY)*.. Desires can underpin goals and other purposeful behavior. Desires and impulses are akin in that impulses may arise out of desires, and that neither need be realistic, achievable or rational. Drives and needs, as described above, do not require deliberative mechanisms and architectures capable of supporting adaptive state automata suffice to model these. Quantitative goals can encompass needs and drives but are differentiated to allow for more flexible representations and process models. These are the types of goals discussed in engineering control theory (Sontag 1998) and reinforcement learning, for example (Maes 1989, Toates 1998). Qualitative goals are the types of motivators discussed in most planning literature (Nilsson 1998). The remaining category identified here, attitudes, are pre-dispositions to respond to specific sensory or cognitive cues in specific ways. For example, an agent could generate pro-active goals to investigate a hapless agent based on an altruistic standard (an attitude) and a set of beliefs about the capabilities of that agent. The work on norms (Staller and Petta 2001) is relevant to this category. The following sections describe computational work in bringing together these analyses in terms of working models. For conceptual (and historical) reasons motivational control states are dealt with before the computational model of affect.

6 Motivated Architectures

Current work on architectures for motivated agents is based on experiments in the theory, design and implementation of affect and emotion based architectures (Davis 1996, 2001a, 2001b). It builds on the ecological perspectives offered by Gibson (1979), and on the work of Simon's control state theory. Preliminary work (Davis 1996) centered on motivators and goals, how they come into being and how they are managed. This led to work on agents and control states (Davis 2001b), again focused on goal processing. It addressed how goals need to be valenced in a number of different ways, for example intensity, urgency, insistence (see table 1). Motivators in these architectures were representational structures generated at a reactive level. The generic representational schema made use of fifteen components that reflected the nature of the most expansive of motivational control states. In many instances, for example behaviours related to drives, many of these components were unused and the stack of motivators could be manipulated by mechanisms analogous to the reactive planners of Kaelbling (1989). Where required more extensive (and computationally expensive) deliberative processes are used. An instance of this is the motivator merging, given in (Davis 2003a), which made use of mechanisms analogous to those used in teleological planning (Nilsson 1994).

Valence	Process and Dimension Category
Belief Indicator	Function over Truth values for Semantic Content and Motivator Attitude
Commitment Status	Fuzzy Model (ignored to first priority)
Dynamic State	Fuzzy Model (instantiated to complete)
Importance	Fuzzy Model (low to high)
Insistence	Fuzzy Model (low to high)
Intensity	Fuzzy Model (low to high)
Urgency	Fuzzy Model (low to high) or time cost function
Decay	Fuzzy Model (low to high) or time cost function

Table 1. Valences within a motivational construct.

The architectures developed in this work, and related research into a multi-level representational framework for emotion (Davis 2002), made use of variations of the three column, three level architecture developed with the Cognition and Affect project (Beaudoin 1994, Davis 1996, Sloman 1990, 1995, Sloman et al 2004, Wright et al 1996).

We continue to use variations of a three-column, three layer architecture but are not unequivocally committed to such architectures, if the research requires other frameworks. Figure 5, for example, shows a four tier, five column instance. Some experimentation (Davis and Lewis 2003, 2004) makes use of an architecture based on cognitive models of reasoning in children (Bartsch and Wellman 1989, Wahl and Spada 2000). The approach taken is one merges the principles of architectural parsimony (Hayes-Roth 1993) and the conceptual ease through architectural expansion of Singh and Minsky (2003).

In the three-layer model, there exist reflexes and reactive behaviours that allow a direct response to sensory events. These can provoke processes or being modified at a more abstract level. Other automatic processes *necessitate* the generation of deliberative control states to achieve their goals. The deliberative layer represents those (control state) processes typically studied in thinking, human problem solving etc., plus other processes related to the management of low level actions. The reflective processes serve to monitor cognitive behaviour or control it in some other way. The more extreme affective states (symptomatically categorised as a loss of control or perturbance) are effectively clamped by means of self-regulatory processes within the architecture. This model is quite general. The effect of altering the relative size and importance of the layers is an open issue. High level and low level processes coexist and interact in a holistic manner through the use of motivation and affect. In effect, goal processing, planning, decision making and other cognitive processes are not purely abstract but exist in relation to other automatic, affective and motivational processes. They are, in effect, embodied within the context of their interactions with their underlying processes and the agent's relationship(s) with its environment.

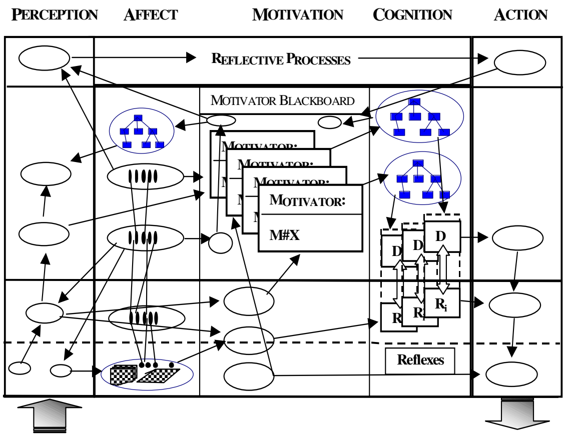

Figure 5. The Four Tier-Five Column Architecture

The most recent design work sketched in figure 5 shows a four tier five column architecture. The four tiers represent reflex, reactive, deliberative and reflective processes. The five columns separate perception, affect and affordance processes, the motivational blackboard, generic cognitive processes (planning, behaviours etc) and output to actions. This framework extends earlier work with the architectural global blackboard for the motivational constructs. Earlier research (Davis 2001b) did not separate these processes from generic cognitive functions. This architecture makes use of the extended motivational constructs as blackboards that provide the context for ongoing (and most other dynamics of) processing. The representational structure that is the architecture can use one or more motivational constructs concurrently. Both architecturally generic and motivational construct specific processes can access the blackboards and in turn be influenced by their content and processes. The emotion engine of earlier research (Davis 2001a) is now superceded by the affect processes column. The work on multi-level representations of emotions that run over semi-autonomous cellular automata models is being revisited in the light of current thoughts on the nature of affect (as outlined in this article) and the work of Adamatzky (2003) on computational chemistry models of affect. The latter in hand with the blackboard scheme for motivation provide a sophisticated interaction of very low level, reactive and deliberative processes in a multiply valenced framework.

The affective valencing of processes and representational structures can be given or the agent can adapt or learn appropriate affordances according to its role and current environment. It forms the basis for perceptual valences that support the external environment affordances appropriate to the agent. As an agent monitors its interactions within itself and relates these to tasks in its external environment, the impetus for change within itself (i.e. a need to learn) is manifested as a motivational state. Such a control state can lead to the generation of internal processes requiring the agent to modify its behaviour, representations or processes in some way. The modification can be described in terms of a mapping between its internal and external environments. This influences the different categories of cognitive and animated behaviour. To paraphrase Mearleu-Ponty (1942), an agent is driven to learn, adapt and act in its environment by disequilibria between the self and the world. The valences used in the current motivational structure (table 1) provide the means to characterise the disequilibria. The multi-dimensional measures associated with the motivational construct, in effect, provide the fitness function for easing any such disequilibria. The problem remains how to generate these values and decide across the current stack of motivators in a manner that does not rely on ad hoc control heuristics.

6.1 Affect, Affordance and Motivation

Previous research (Davis 2001a) has used emotional models that include basic emotions. The current stance is that basic emotions are unnecessary in a theory of emotion. A number of emotion theories use the concept of basic emotions; Scherer (1994) instead allows for modal forms of emotive processing. Of the many modes that an emotion system can take, some are near identical or relatively similar to the states described as basic emotions. However the range of states in a modal model is far more diverse. A salient feature of many theories of emotion is that they are described in terms of goals and roles. Emotion in the goal-based theories, for example (Oatley 1992), can be described as "*a state usually caused by an event of importance to the subject*". This involves mental states directed towards an external entity (attitudes, motivations, expectations etc.), physiological change (increased heart beat, hormone release etc), facial gestures and some form of expectation. Scherer (1994) defines emotion as *"a sequence of interrelated, synchronised changes in the states of all organismic subsystems (information processing, cognition, support, ANS, execution, motivation, action, SNS, monitoring, subjective feeling) in response to the evaluation of an external or internal stimulus event that is relevant to central concerns of the organism"*. These emotional processes involve five functionally defined systems involving information processing over perception, regulation of internal states, decision making over competing motives, the control of external behaviour and a feedback system across these four. This differentiation of processes can be easily mapped onto the architectural model of figure 5. While still accepting the validity of developing a computational theory of emotion, there is a very important adjunct. Emotions are considered unnecessary for most synthetic systems, and that the case for including emotion in a synthetic system should be based on an analysis of the demand for emotions in the developed system. Given the motivational framework outlined in the previous sections, the requirement is that some model of affect is required. This may not necessarily involve the emotions, and may be simpler in its requirements than the mechanism necessary for a fully developed implementation of the emotions.

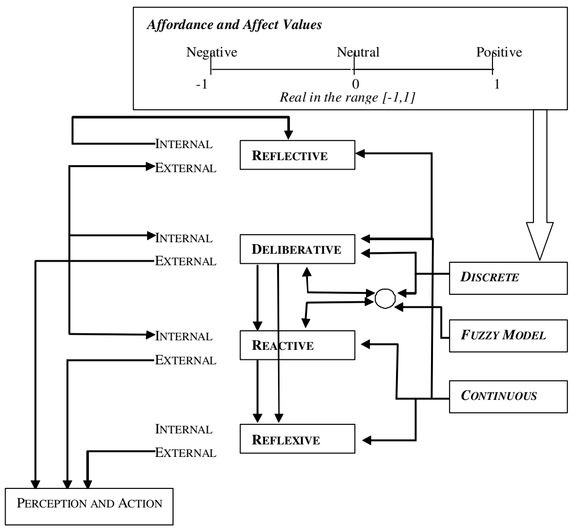

Figure 6. Affect Model (in second column of figure 5)

We are developing a theory of affect that draws on themes such as control states and motivators (Simon 1967; Sloman 1990; Davis 2001b) and affordances (Gibson 1979; Davis 2001a). The overlap between the goal-based and modal response theories provides for a coherent hybridization and defines the bare bones of the affect model used here. We define affect in terms of reinforcers over processes and representational structures. It is qualitatively defined over negative, neutral or positive values, as in the work of Rolls (1999), or numerically over the interval (-1.0,1.0). Future work will look to develop the fine details of a fuzzy (and/or neural) valued processing model that maps across these measures at different levels of the architecture (Figure 6). This will build on the research on the emotion engine and also relate to the eight valences for the currently developed motivational construct (Table 1). Hence, affect forms a basis for a control language for agent architectures. It allows external events and objects to take valenced affordances, and allows the results of internal mechanisms to be prioritised and compared via valenced processes. At the deliberative level, affective values can be associated with processes and control signals to instantiate and modify aspects of motivators and their associated representations. Furthermore, if an agent is to recognize and manage emergent behaviours, and particularly extreme and disruptive control states, this multi-layer model of affect provides the means for reflective processes to do this. This model of affect addresses the need to integrate reflective, deliberative, reactive and reflexive level agencies in a synergistic fashion.

7 Discussion

This paper has confronted the now widely held requirement for emotion in intelligent systems on a number of grounds. The starting thesis is that overall the theory of emotion is currently too disorganised to be of much use in the design of synthetic intelligence. More pointedly, emotion is not really a requirement for many forms of synthetic intelligence, and that more straightforward affective means, based on something as straightforward as the concept of affective taxes or reinforcers, can be used to enable effective decision-making. Elsewhere, it has been suggested (Davis and Lewis 2004) that a direction given by the less semantically overloaded term affect is more appropriate for synthetic intelligence. The problem is however that the phenomena covered by affect are even more diverse and currently less well specified than emotions! Future research will determine how complex are the states arising from the adoption of the simple model outlined here.

If our research agenda is slightly different and pursues the theory, design and building of artificial systems that sometimes work analogously to human mind does this requirement for emotions still hold? In negating the use of emotion some alternative is required, not just to simply mirror the fact that natural minds use emotion but because some form of motivational control language is required to do anything associated with mind. Consider activities such as sensory attention, behaviour selection, goal maintenance and the learning of new skills. There needs to be some valence or fitness function associated with these, whether explicit or implicit. Some means of conflict resolution is required. For example given two contrasting percepts, both of which are equally viable for an agent to act on, but which require mutually exclusive processing, how does the agent determine which to attend? Without the appropriate criteria to choose between two equally plausible activities, the agent in effect will have to choose at random. Many artificial systems in the past have used ad hoc control heuristics to solve prioritization of activity or heuristically defined domain parameters (see for example Englemore and Morgan 1988). Here we suggest that at a theoretical, design, architectural and implementation level a consistent valencing and control language based may offer more to the pursuit of synthetic intelligent systems. That this language at times bears similarities to the language used to describe emotion and affect should not be surprising.

Consider a highly modular architecture for a synthetic mind. Within this framework exist many vertical and horizontal modules, some highly specialized and responsible for specific activities and processing, some generic, some very localised and others more global. There should exist some global mechanisms that at least provide for context and integration of modules. It matters not for the time being whether the global mechanisms for context are based on ideas such as computational chemistry (Adamatzky 2003), or global workspaces (Baars 1997, Franklin 2001) or blackboards (Hayes-Roth 1993) or some combination or neither. Should and how can the control architecture make consistent decisions across these different modules and mechanisms? We suggest the use of multiple-level representation based on the idea of affective taxes. This will bear some similarity to aspects of a number of theories of emotion where they serve useful satisfaction for system requirements. For example in integrating behaviours (whether innate, adapted or acquired) into a skill sequence for a particular context, affective dissonance provides a fitness function to be minimized. At the individual module level, we require a fitness function mapping input to output (for example as an affordance and accordance over the requisite sensori-motor mapping). At a more abstract level, we are using a representational schema (Davis 2001b) as local blackboards for reasoning about motivations, goals and other forms of control states. Again we look to provide a consistent valencing mechanism across control states, behaviours and architecture levels.

The theory of synthetic intelligent systems can therefore progress without the need for emotion per se but with a requirement for affective control states that can draw on theories of emotion and cognition in biological intelligent systems. This would mean for example that a synthetic system need not model or recognise the emotive state termed fear but recognise highly valenced negative internal states and environmental affordances that (potentially) jeopardise its role and tasks in its current environment. Put simply, theories of emotion from the cognate disciplines such as neurophysiology, philosophy and psychology can afford functional models of affect for synthetic systems without the need for the theorist or designer of synthetic systems to be concerned with the semantic overloading associated with specific emotions. Furthermore most theories of emotion involve body attitude or facial expression changes that are typically inappropriate for machines. As yet, there are no machines that rely on body posture or facial expression for inter-communication other those affective systems that attempt model the emotive state of their user (Picard 1997). Even there the interactive system needs only to model the emotive or affective state of its user, and not function in terms of emotion.

8 Conclusion

Recent experimental work (Nunes 2001, Bourgne 2003) has revisited the representational structure and processes associated with motivators (Beaudoin 1994, Davis 1996), but made use of affect and affordances to valence the motivational constructs. Associated with motivational structures are attitudes to classes of events and entities relevant to that motivator. These are valenced in the same way that affordances and affect are valenced. The association of perception, behaviour and abstract representations about plans of actions and the relevance of actions and entities in the environment with agent internal worlds can now be defined and compared in terms of a common criteria. Affect and affordance become the means by which an agent architecture can weigh and control the economics of its processing. It provides a means whereby attention can be directed to the most relevant and/or pressing aspects of the interactions of the agent with the environment, its needs and its goals. Related work (Davis and Lewis 2003, 2004) suggests that adding a simple model affect to cognitive desire and intention models such as CRIBB (Bartsch and Wellman 1989), result in more effective processing and task management in resource competitive environments.

Returning to the theme of the introductory paragraph, as the designer of artificial intelligent systems one could ask what is the biological analogue to the information processing complexity of the system being designed and developed? If it is insect, does it need affect or emotion and would not some other criteria be more appropriate? In short is there a need for emotion in the system. The developer of multi-media interfaces (Kort et al 2002) may require some form of affective processing in generating hypotheses about the emotive state of the user sat outside the video-camera within the interface. But does the designer of the computational equivalent to a grasshopper? Albeit a grasshopper that can manipulate text?

The reason these questions is raised is the ongoing efforts of cognitive scientists across many disciplines, philosophers, psychologists, computer scientists, neurophysiologists etc., to move the theory of affect (and emotions) forward. The roots of current theories reside in folk psychology and historical theories of affect. Are the efforts of the practitioners in giving their artificial systems emotion helping this progress? It is suggested here that in order to make more substantial progress, efforts are required to provide the means by which we can categorise the types of information processing systems in existence and being developed, whether natural or synthetic. A means of providing the discriminatory criteria necessary to perform such conceptual analysis, built from the work of Sloman and others (2004) has been given here.

References

Adamatzky, A., Affectons: automata models of emotional interactions. Applied Mathematical Computation, 146(2), 579-594, 2003.

Aubé, M., Beyond needs: Emotions and the commitments requirement. In: Davis, D.N. editor, Visions of Mind: Architectures for Cognition and Affect, Idea Group Inc., Hershey, PA, USA, 2004.

Baars, B. J., In the Theater of Consciousness. Oxford: Oxford University Press, 1997.

Bartsch, K. and Wellman, H. Young children's attribution of action to beliefs and desires. Child Development, 60, 946-964, 1989.

Bates, J., Loyall, A.B. and Reilly, W.S., Broad agents, SIGART BULLETIN, Vol. 2, No. 4, 1991

Beaudoin, L.P., Goal Processing in Autonomous Agents, Ph.D. Thesis, School of Computer Science, University of Birmingham, 1994.

Bourgne, G., Affect-based Multi-Agent Architecture (for a 5-aside football simulation), Thesis, Computer Science, University of Hull, 2003.

Charland, L.C. Emotion as a natural kind: Towards a computational foundation for emotion theory, Philosophical Psychology, Vol. 8, No. 1, 59-85, 1995.

Damasio, A. Descartes' error: Emotion, reason and the human brain. New York: Grosset/Putman, 1994.

Darwin, C. The Expression of Emotions in Man and Animals. London. Murray, 1872.

Davey, G., Ecological Learning Theory, Routledge, 1989.

Davis, D.N., Reactive and motivational agents. In: Intelligent Agents III, J.P. Muller, M.J. Wooldridge and N.R. Jennings (Eds.), Springer-Verlag, 1996.

Davis, D.N., Multiple Level Representations of Emotion in Computational Agents, Emotion, Cognition and Affective Computing, AISB2001: Agents and Cognition, University of York, 2001

Davis, D.N., Control States and Complete Agent Architectures., Computational Intelligence, 17(4):621-650, 2001

Davis, D.N. Computational Architectures for Intelligence and Motivation, IEEE Systems and Intelligent Control, Vancouver, Canada, 2002.

Davis, D.N. Architectures for Cognitive and Artificial-Life Agents In: Intelligent Agent Software Engineering, V. Plekhanova (eds.), IDEA Group Publishing, Copyright 2003

Davis, D.N. and Lewis, S.C., Computational Models of Emotion for Autonomy and Reasoning. Informatica (Special Edition on Perception and Emotion Based Reasoning), 27(2):159-165, 2003.

Davis, D.N. and Lewis, S.C. Affect and Affordance: Architectures without Emotion. AAAI Spring symposium 2004, Stanford University, California, USA

Duffy, E., Activation and Behaviour, Wiley and Sons, 1962.

Ekman, P. The Nature of Emotion: Fundamental Questions. Oxford University Press. New York, 1994.

Elliott, C., The Affective Reasoner: A Process Model of Emotions in a Multi-Agent System. PhD Thesis. Northwestern University, USA, 1992.

Englemore, R. and Morgan, T., Blackboard Systems, Addison-Wesley, 1988.

Fodor, J., The Modularity of Mind (Cambridge, MA, A Bradford Book), 1983.

Franklin, S., A consciousness based architecture for a functioning mind, Proceedings of the AISB'00 Symposium on How to Design a Functioning Mind, April 2000.

Franklin, S., Conscious Software: A Computational View of Mind. In Soft Computing Agents: New Trends for Designing Autonomous Systems, ed. V. Loia, and S. Sessa. Berlin: Springerm 2001.

Frijda, N and Swagerman, J.. Can computers feel? theory and design of an emotional system. Cognition and Emotion, 1(3):235--257, 1987.

Georgeff, M.P. and Lansky, A.L., Reactive reasoning and planning. Proceedings of the Sixth National Conference on Artificial Intelligence, 2, pp. 677-682, Seattle, WA: AAAI, 1987.

Gibson, E. and Ingold, T (Editors), Tools, Language and Cognition in Human Evolution, Cambridge University Press, 1993.

Gibson, J.J., The Ecological Approach to Visual Perception, Boston: Houghton Mifflin, 1979.

Griffiths, P. E., Is emotion a natural kind? In Solomon, R., editor, Philosophers on Emotion. Oxford University Press, 2002.

Hayes-Roth, B., Intelligent control. Artificial Intelligence, 59:213—220, 1993.

Hebb, D.O. Emotion in man and animal: an analysis of the intuitive process of recognition, Psychological Review, 53, pp. 88-106, 1946.

James, W. What is an Emotion? Mind. 9, 188-205, 1884.

Kaelbling, L.P., An Architecture for Intelligent Reactive Systems, Readings in Planning, Morgan Kaufmann, 1989.

Kort, B., Reilly, R. and Picard, R., Analytical Models of Emotions, Learning, and Relationships: Towards an Affective-sensitive Cognitive Machine, Proceedings of the Intelligent Tutoring Systems Conference (ITS2002), pp.955-962, France, 2002

Maes, P., How to do the right thing, Connection Science, Vol. 1, 1989.

McFarland, D., Animal Behaviour, Addison Wesley Longman, 1993.

Merleau-Ponty, M., The Structure of Behaviour, *Methuan:London, (ET 1965), 1942.*

Minsky, M.L., The Society of Mind, William Heinemann Ltd., 1987.

Nilsson, N.J, Teleo-reactive programs for agent control, Journal of Artificial Intelligence Research 1, 1994.

Nilsson, N.J., Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998.

Norman, D.A., Twelve issues for cognitive science, Cognitive Science, 4:1-33, 1980.

Nunes, H.A., Investigation of Motivation in Agents Using Five-Aside Football, M.Sc. Thesis, Department of Computer Science, University of Hull, 2001.

Oatley, K. Best Laid Schemes. Cambridge. Cambridge University Press, 1992.

Ortony, A., Clore, G.L. and Collins, A., The Cognitive Structure of Emotions. Cambridge University Press, 1988

Picard, R., Affective Computing, MIT Press, 1997.

Plutchik, R. The Psychology and Biology of Emotion. Harper Collins. New York, 1994.

Pylyshyn, Z. Computation and Cognition: Toward a Foundation for Cognitive Science, Cambridge, MA, A Bradford Book, 1984.

Reilly, W S. Bates, J. Emotion as Part of a Broad Agent Architecture. In WAUME 93: Workshop on Architectures Underlying Motivation and Emotion. University of Birmingham, England, 1993.

Scherer, K. Studying the emotion-antecedent appraisal process: An expert system approach, Cognition and Emotion, 7, 325-355, 1994.

Scherer, K. Toward a Concept of 'Modal Emotions'. In P. Ekman and R. Davidson (Eds): Nature of Emotion, Oxford University Press. New York, 1994.

Scherer, K. Appraisal Considered as a Process of Multilevel Sequential Checking. In K. Scherer, A. Schorr, T. Johnstone (Eds). Appraisal Processes in Emotion. Oxford University Press. New York, 2001.

Simon, H.A., Motivational and emotional controls of cognition, Originally 1967, Reprinted in Models of Thought, Yale University Press, 29-38, 1979.

Singh, P. and Minsky, M., An Architecture for Combining Ways to Think, Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems, 2003.

Sloman, A. and M. Croucher, Why robots will have emotions. Proceedings of IJCAI87, 197-202, 1987.

Sloman, A., Motives mechanisms and emotions. In: M.A. Boden (Editor), The Philosophy of Artificial Intelligence, Oxford University Press, 1990.

Sloman, A. Exploring design space and niche space, Proceedings of the 5th Scandinavian Conference on AI, Trondheim, Norway, IOS Press, 1995.

Sloman, A., Beyond shallow models of emotion, Cognitive Processing, 2(1):177-198, 2001.

Sloman, A. What are emotion theories about? Multidisciplinary workshop on Architectures for Modeling Emotion at the AAAI Spring Symposium at Stanford University in March 2004.

Sloman, A., Chrisley, R. and Scheutz, M.The Architectural Basis of Affective States and Processes In: 'Who needs emotions?', edited by Jean-Marc Fellous and Michael Arbib, 2004.

Sontag, E.D., Mathematical Control Theory: Deterministic Finite Dimensional Systems. Second Edition, Springer, New York, 1998

Staller A. and Petta P.: Introducing Emotions into the Computational Study of Social Norms - A First Evaluation. In: Edmonds B. & Dautenhahn K. (Eds.), Starting from Society: the Application of Social Analogies to Computational Systems, *Special Issue of the* Journal of Artificial Societies and Social Simulation*, 4(1), 2001*

Toates, F., (Editor), Control of Behaviour, Springer Verlag, 1998.

Velasquez, J. D. Cathexis – A Computational Model for the Generation of Emotions and their Influence in the Behaviour of Autonomous Agents. PhD Thesis. MIT, 1996.

Wahl, S. and Spada, H., Children's Reasoning about intentions, beliefs and behavior, Cognitive Science Quarterly, 1, 5-34, 2000.

Wehrle, T, New Fungus Eater Experiments, in: P. Gaussier and J.-D. Nicoud (Editors) From perception to action, pp 400-403, IEEE Computer Society Press, 1994.

Weir, A.A.S., Chappell, J., and Kacelnik, A., Shaping of hooks in New Caledonian crows. Science, 297:981, 9th August, 2002

Wright, I.P, Sloman, A, and Beaudoin, L., Towards a Design-Based Analysis of Emotional Episodes, Philosophy Psychiatry and Psychology, 3(2): 101—126, 1996.

Do somatic markers need to be somatic? Analogies from evolution and from hardware interlocks

Colin G. Johnson
*Computing Laboratory
University of Kent
Canterbury, Kent
CT2 7NF, England
C.G.Johnson@kent.ac.uk

Abstract

This paper considers Damasio's concept of the *somatic marker* from two new perspectives. The first of these considers them from the point of view of Dawkins's concept of the *extended phenotype*. This is used to develop the idea of the *extended somatic marker*, *viz.* a marker which uses some non-somatic feature of the external world in a similar fashion to the somatic marker. Secondly an analogy is developed with the concept of *hardware interlocks* in safety-critical systems. This is used to suggest why it is important that somatic markers are bodily states and not just mental markers.

1 Somatic Markers

Damasio Damasio (1994) has introduced the notion of the *somatic marker*—a bodily state which plays a role in cognition, in particular the direction of attention. More specifically, a somatic marker is some bodily state which is generated as the consequence of some mental process. This state is then *reperceived* by the mind, and as a consequence the mental state changed. An example of such a marker is the rapid onset of nausea upon witnessing an act of violence. This bodily state does not have any immediate relevance to the mental state which has generated it, in contrast, say, to a feeling of nausea generated by viewing a plate of rotting food. Some such states might be explained away as side-effects. For example a rapid change of hormone levels upon witnessing violence in preparation for running from the danger might also trigger nausea.

However the somatic marker hypothesis suggests that such reactions are not mere side-effects. Instead they are a way of generating a rapid shift of attention, using the body state in an arbitrary fashion to draw mental attention to the current situation. The presence of the marker in the body draws the mind's attention towards it, and as a consequence the mind if focused on the meaning of that marker. It is plausible that such phenomena are exaptations Gould and Lewontin (1979) from unwanted physical reactions to change in body state as discussed above.

This can be seen as an aspect of mind which is realised away from the usual mental substrate. The somatic response is being used as a way of carrying out a process (bringing the attention of many mental processes together to focus on a single danger point) which cannot be carried out within the computational model implemented on the substrate.

The aim of this paper is to consider why the markers in question need to be *somatic* as such. Two aspects of this question are considered. Firstly, would it be possible for markers to extend beyond the body? This is explored with reference to Dawkins's concept of the *extended phenotype*. Secondly, why is it important that such markers be in the body, instead of being more simply realised by mental markers? This is explored with regard to the idea of *hardware interlocks* in engineering design.

2 Could "somatic" markers extend beyond the body?

Why do markers need to be *internal* body states. Is there anything which is special to the body which means that the markers could not instead be realised elsewhere in the world, external to the body? Might some of our actions in the world act as triggers to affect, perceived directly through the usual perceptive system rather than by bodily self-awareness?

One approach to this draws on ideas from Dawkins's book *The Extended Phenotype* Dawkins (1982). In biology, the *phenotype* is the expression of a gene or set of genes in the world. This encompasses both the aspects concerned with the physical structure of the creature and through the ways in which genes have influences on behaviour. For example we can talk about the "blue-eyed" phenotype versus the "brown-eyed phenotype" of some animal. This is distinguished from the "genotype", i.e. the set of genes of interest. Sometimes more than one genotype can give rise to the same phenotype (e.g. where there are regressive traits).

The difficulty starts when we want to say where the boundary of the phenotype lies. Clearly certain things are in the phenotype for certain. A clear example of this is the sequence of proteins associated with a particular expression of a particular gene. A standard definition would extend this to the whole body; genes influence the growth, development, and activity of the body (alongside other influences).

Dawkins's argument is that it is naive to simply say that everything inside the body should be considered to be phenotype, whereas everything outside should not. For example consider an imaginary species of bird in which the male has a gene which predisposes itself to mate with females which have blue feathers. It could be argued that this gene is also a gene for blue feathers in the female, as as a result of the presence of the gene blue feathers will spread through the female population. To abstract this, the genotype in the male bird is having a phenotypic effect in the female bird. Why should we regard the gene's effect on the feathers of the female bird in any different way to another gene which causes the male bird to have red eyes?

A similar kind of argument can be made about the somatic marker hypothesis. Damasio argues for a body-minded brain in which we create emotions via "somatic markers". These work by parts of the brain recognizing an emotionally charged stimulus, and then rather than creating a direct link to an action on that stimulus, the "marker" consisting of a bodily reaction is created. This is then re-perceived by the brain as is the basis for action or for rapid alteration of emotional state.

Why do these markers have to be physically internal to the body? It would seem that the same reasoning could be applied to markers which I leave in the external world when I have an emotion. For example if I am anxious then I might scribble on the pad of paper in front of me, without attending to this scribbling. This could then become a marker, in this case perceived via the eyes rather than through internal perception of bodily state. Why should it matter whether I use a bodily state or an external state as the substrate for the marker?

It may be that there are reasons why somatic markers need be somatic. One could be that the speed of reaction required is just too quick to be capable of being carried out by the external perceptive system. Another more convincing explanation is that the reason we use somatic markers is to communicate with multiple brain regions in a simultaneous and co-ordinated way, and therefore we need something which can be perceived in a direct way by different parts of the brain.

This might be a continuum effect. An example of a thing which might be seen as either an external or somatic marker is biting nails when anxious. This is in many ways an external physical process, nonetheless we can perceive the nail state internally via soreness of fingers. There must be other similar examples. Perhaps nail-chewing is "causing" the anxiety (in the sense of being part of the causal chain between subconscious perception of an anxiety-producing stimulus and the affective response) rather than being an epiphenomenon of the emotional state.

3 Why do markers need to be confined to the body?

So far we have considered why it is that the somatic marker need be constrained to the body, and is it important to make a body/non-body distinction. Now we address the opposite question: why is it not sufficient for the marker to be a mental marker? Why not just make a "mental note"? Whilst there are circumstances where a truly somatic marker can get transformed into a mental process in the limbic system Damasio (1994), this is not always the case; markers are not always transferred in this fashion. It is interesting to consider whether there might be reasons why the evolution of the mind might have led to the markers being body-centred rather than mind-centered.

One reason may be for safety. In the design of complex systems involving computer-controlled mechanical and electrical devices it is common for there to be conservative safety devices included in the system known as *hardware interlocks* Leveson and Turner (1993); Leveson (1995). A hardware interlock is a device which is independent of the main control system, and which is designed to monitor just on small aspect of the system, typically by using its own sensor system. So for example in a radiotherapy device, an interlock might exist which monitors the output of radiation, and if more than a certain amount is let out in one minute, the interlock shuts down the device completely.

Hardware interlocks are designed to be parts of the overall system which do not depend on the abstraction offered by the overall control system. For example they do not take information from the main system sensors, nor do they use the main control system e.g. for timing, and they do not sit upon the operating system abstraction used by the controlling structure. To do this would compromise their role as a safety-critical component; they provide a reassurance of safety because they are separate, they are independent from the main abstraction. If the main sensors go wrong, or the builder of the controller has misunderstood the relationship between the abstraction offered by the operating system and the real hardware and software, it does not matter.

One important role in the body-mind system is to react quickly and reliably to dangerous phenomena. There would seem to be a *prima facie* case for thinking that if engineers consider the use of such hardware interlocks as an important way of responding to danger in computer-controlled systems, evolution may have created such interlock systems for dangers to animals.

It may be that our body-grounded response to danger is a response of this kind. Instead of making a mind-centered judgement about the danger of a situation, we instead make a rapid decision based on a few simple cues. One characteristic of hardware interlocks is that they typically work on a small number of basic sensors which facilitate a conservative approximation to safety. The same may be true of interlocks in the mind-body system: our sensory system perceives a small number of simple "danger signals" (such as a rapid movement) and triggers an action within the body immediately. This "massive synchronization" acts as a counterpart to the more commonly-discussed "massive parallelism" of the neural-network-based mind.

Typically the fact that the brain is a unified system with all aspects connected and mutually-accessible is seen to be to its advantage. Similarly the unity found in a complex software system is often seen as being to its advantage; instead of having to connect individual components together as needed (as might be the case in an electronic system) all information is passed to a central repository and accessed as needed. However in some situations it is necessary both with computers and with minds for the complete attention of the system to be directed towards one thing. Hardware interlocks provide a way for such responses to "leap out" of the complexity of the control software for certain emergency situations. This non-decomposability, and the consequent need for a powerful way of leaping out of the complex interactions, would seem to be particularly strong for neural-network-based systems where the system is highly non-decomposable.

Acknowlegdements

Many thanks to the reviewers for interesting suggestions about the ideas contained in this paper.

References

A.R. Damasio. *Descartes' Error : Emotion, Reason and the Human Brain*. Gosset/Putnam Press, 1994.

Richard Dawkins. *The Extended Phenotype*. Oxford University Press, 1982.

S.J. Gould and R.C. Lewontin. The spandrels of San Marco and the Panglossian paradigm: A critique of the adaptationist program. *Proceedings of the Royal Society of London, Series B*, 205(1161):581–598, 1979.

N. Leveson and C. Turner. An investigation of the Therac-25 accidents. *IEEE Computer*, 26(7):18–41, 1993.

Nancy Leveson. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

Emotional-based Planning

Luís Macedo[1,2]
[1]Department of Informatics and Systems Engineering, Engineering Institute, Coimbra Polytechnic Institute
R. Pedro Nunes, Quinta da Nora, 3030-199 Coimbra - Portugal
lmacedo@isec.pt

Amílcar Cardoso[2]
[2]Centre for Informatics and Systems of the University of Coimbra
Pinhal de Marrocos, 3030 Coimbra - Portugal
{macedo,amilcar}@dei.uc.pt

Abstract

This paper describes an emotional-based planner that combines the technique of decision-theoretic planning with the methology of HTN planning in order to deal with uncertain, dynamic large-scale real-world domains. We explain how plans are represented, generated and executed. Unlike in regular HTN planning, this planner can generate plans in domains where there is no complete domain theory by using cases instead of methods for task decomposition. The planner generates a variant of a HTN - a kind of AND/OR tree of probabilistic conditional tasks - that expresses all the possible ways to decompose an initial task network. The expected utility of alternative plans is computed beforehand at the time of building the HTN. Two approaches are proposed for this computation: based on motivational information collected from past executions of tasks (a kind of somatic-markers) or given by mathematical functions. The planner is used by agents inhabiting unknown, dynamic environments.

1 Introduction

Hierarchical Task Network (HTN) planning is a planning methology that is more expressive than STRIPS-style planning (Erol, Hendler, & Nau, 1994). Given a set of tasks that need to be performed (the planning problem), the planning process decomposes them into simpler subtasks until *primitive tasks* or actions that can be directly executed are reached. *Methods* provided by the domain theory indicate how tasks are decomposed into subtasks. However, for many real-world domains, sometimes it is hard to collect methods to completely model the generation of plans. For this reason an alternative approach that is based on cases of methods has been taken in combination with methods (Muñoz-Avila et al., 2001).

Real-world domains are usually dynamic and uncertain. In these domains actions may have several outcomes, some of which may be more valuable than others. Planning in these domains require special techniques for dealing with uncertainty. Actually, this has been one of the main concerns of the planning research in the last years, and several decision-theoretic planning approaches have been proposed and used successfully, some based on the extension of classical planning and others on Markov-Decision Processes (see (Blythe, 1999; Littman & Majercik, 1997) for a survey). In these decision-theoretic planning frameworks actions are usually probabilistic conditional actions, preferences over the outcomes of the actions is expressed in terms of an utility function, and plans are evaluated in terms of their Expected Utility (EU) (Russel & Norvig, 1995). The main goal is to find the plan or set of plans that maximizes an EU function, i.e, to find the optimal plan. However, this might be a computationally complex task.

Considered by many authors as the principal motivational system, emotion is one of the sub-systems that compose personality (Izard, 1991). Another important sub-system is the drive system (also an important kind of the motivational system). Psychological and neuroscience research over the past decades suggests that emotions play a critical role in decision-making, action and performance, by influencing a variety of cognitive processes (e.g., attention (Izard, 1991; Meyer, Reisenzein, & Schützwohl, 1997; Ortony & Partridge, 1987; Reisenzein, 2000), planning (Gratch, 1999), etc.). Actually, on the one hand, recent research in neuroscience (Damásio, 1994; LeDoux, 1996) supports the importance of emotions on reasoning and decision-making. For instance, results from recent studies of patients with lesions of the prefrontal cortex suggest an important role of emotions in decision-making. On the other hand, there are a few theories in psychology relating motivations (including drives and emotions) to action (Izard, 1991). For instance, in the specific case of emotions, as outlined by (Reisenzein, 1996), within the context of the belief-desire theories of action (the dominant class of theories in today's motivation psychology) there have been proposals such as that emotions are action goals, that emotions are or include action tendencies, that emotions are or include goal-desires, and that emotions are mental states that generate goal-desires.

In this paper we propose an emotional-based approach for decision-theoretic planning, HTN planning. In this approach, actions have several outcomes, each one eliciting different emotions, drives and other motivations (elicited by the objects perceived). This motivational information is collected from past executions of tasks (a kind of somatic-markers) or given by mathematical functions. The selection of actions is based on their EU, which is measured in terms of this motivational information, i.e., based on the intensity of the emotions, drives and other motivations it may elicit. The planner combines the technique of decision-theoretic planning with the methology of HTN planning in order to deal with uncertain, dynamic large-scale real-world domains. Unlike in regular HTN planning, we don't use methods for task decomposition, but instead cases of plans. The planner generates a variant of a HTN - a kind of AND/OR tree of probabilistic conditional tasks - that expresses all the possible ways to decompose an initial task network. The EU of tasks and consequently of the alternative plans is computed beforehand at the time of building the HTN.

The next section describes the features of the planner related with plan representation. Section 3 presents the plan generation process and section 4 the plan execution and replanning process. Finally, we present the related work, and present conclusions and future work.

2 Plan Representation

Within our approach we may distinguish two main kinds of plans: concrete plans, i.e., cases of plans (Kolodner, 1993), and abstract plans. Concrete plans and abstract plans are interrelated since concrete plans are instances of abstract plans and these are built from concrete plans. Since the concept of abstract plan subsumes the concept of concrete plan, let us first describe the representation issues related with abstract plans and then present the main differences between concrete plans and abstract plans.

We represent abstract plans as a hierarchy of tasks (a variant of HTNs (e.g., (Erol et al., 1994; Nau, Muñoz-Avila, Cao, Lotem, & Mitchell, 2001)) (see Figure 1). Formally, an abstract plan is a tuple $AP = <T, L>$, where T is the set of tasks and L is the set of links. More precisely, we represent an abstract plan by a hierarchical graph-structured representation comprising tasks (represented by the nodes) and links (represented by the edges). We adopted the adjacency matrix approach to represent these graphs (Macedo & Cardoso, 1998). The links may be of hierarchical (abstraction or decomposition), temporal, utility-ranking or adaptation kind. This structure has the form of a planning tree (Lotem & Nau, 2000), i.e., it is a kind of AND/OR tree that expresses all the possible ways to decompose an initial task network. Like in regular HTNs, this hierarchical structure of a plan comprises primitive tasks or actions (non-decomposable tasks) and non-primitive tasks (decomposable or compound tasks). Primitive tasks correspond to the leaves of the tree and are directly executed by the agent, while compound tasks denote desired changes that involve several subtasks to accomplish it. For instance, the leaf node *driveTruck* of Figure 1 is a primitive task, while *inCityDel* is a compound task. The decomposition of a compound task into a sequence of subtasks is represented by linking the compound task to each subtask by a hierarchical link of type decomposition (denoted by *dcmp*). This corresponds to an AND structure. In addition, a hierarchical plan may also include special tasks in order to express situations when a decomposable task has at least two alternative decompositions. Thus, these special tasks are tasks whose subtasks are heads of those alternative decompositions. We called abstract tasks to those special decomposable tasks because they may be instantiated by one of their alternative subtasks. Thus, they are a kind of abstractions of their alternative instances. Notice that the subtasks of an abstract task may themselves be abstract tasks. This decomposition of abstract tasks into several alternative instances is expressed by linking the abstract task to each subtask by a hierarchical link of type abstract (denoted by *abst*). This corresponds to an OR structure. As we said, in addition to hierarchical links that express AND or OR decomposition (*dcmp* and *abst*), there are also temporal, utility-ranking and adaptation links between tasks. Temporal links are just like in regular HTNs. We followed the temporal model introduced by (Allen, 1983). Thus, links such as *after*, *before*, *during*, *overlap*, etc., may be found between tasks of an abstract plan. Utility-ranking links (denoted by *more_useful*) are used between subtasks of abstract tasks in order to express a relation of order with respect to their EU, i.e., the head tasks of the alternative decompositions of a given abstract task are ranked according to the EU of their decompositions. Adaptation links (Kolodner, 1993) are useful to generate an abstract plan from several plan cases. They explain how tasks and their components are related in a plan and therefore they explain how to adapt portions of cases of plans when they are reused to construct an abstract plan.

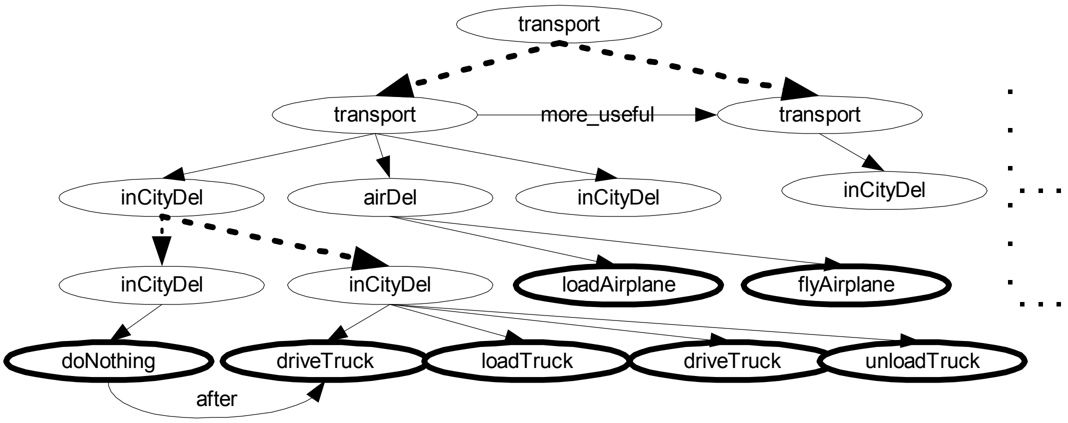

Figure 1 - Example of an abstract plan. Primitive tasks are represented by thick ellipses while non-primitive tasks are represented by thin ellipses. Dashed, thick arrows represent *abst* links, while thin arrows represent *dcmp* links.

A task T is both conditional and probabilistic (e.g.: (Blythe, 1999; Haddawy & Doan, 1994; Younes, 2003)). This means each primitive task has a set of conditions C={ c_1, c_2, ..., c_m} and for each one of these mutually exclusive and exhaustive conditions, c_i, there is a set of alternative effects $\varepsilon^i=\{<p_1^i, E_1^i>, <p_2^i, E_2^i>, ..., <p_{n_i}^i, E_{n_i}^i>\}$, where E_j^i is the j^{th} effect triggered with probability $p_j^i \in [0,1]$ by condition c_i (i.e., $P(E_j^i \mid c_i) = p_j^i$), and such that $\sum_{j=1}^{n_i} p_j^i = 1$. Figure 2 presents the structure of a task. The probabilities of conditions are represented in that structure although we assume that conditions are independent of tasks. Thus, $P(c_i|T)=P(c_i)$. The main reason for this is to emphasize that the EU of a task, in addition to the probability of effects, depends on the probability of conditions too. In addition to conditions and effects, a task has other information components. Formally, a task (primitive or not) may be defined as follows.

Definition. A **task** is a tuple <*PS*, *ID*, *TT*, *AID*, *DO*, *IO*, *ST*, *ET*, *SL*, *EL*, *PR*, *A*, *EP*, *EU*, *P*>, where: *PS* is the set of preconditions that should be satisfied so that the task can be executed; *ID* is the task's identifier, i.e., an integer that uniquely identifies the task in a plan; *TT* is the task category (e.g.: *driveTruck*, *transport*); *AID* is the identifier of the agent that is responsible for the execution of the task; *DO* is the direct object of the task, i.e., the identifier of the entity that was subjected to the task directly (e.g.: for a task of type *driveTruck*, the direct object is the object - its id - to be driven; for a task of type *transport*, the direct object is the entity that is transported – for instance, a package); *IO* is the indirect object of the task, i.e., the answer to the question "To whom?" (e.g.: for a task of type *give*, the indirect object is the entity that receives the entity (the direct object) that is given – for instance, the person who receives money); *ST* is the scheduled start time of the task; *ET* is the scheduled end time of the task; *SL* is the start location of the agent that is responsible for executing the task; *EL* is the end location of the agent that is responsible for the execution of the task; *PR* is a boolean value that is true when the task is primitive; *A* is a boolean value that is true when the task is abstract (for primitive tasks it is always false); *EP* is the set of alternative probabilistic conditional effects of the task, i.e., $EP = \{<c_i,\varepsilon^i>: 1 =< i <= m\}$; *EU* is the Expected Utility of the task; *P* is the probability of the task (this is always 1.0 for every task except the heads of alternative decompositions of an abstract task as we'll explain below).

Although non-primitive tasks are not directly executable by an agent, they are represented like primitive tasks. Therefore, some of the components are meaningful only for primitive tasks. However, others such as the set of alternative probabilistic conditional effects are essential for the ranking of the alternative decompositions of the abstract tasks in terms of the EU. That is why the set of conditional probabilistic effects and other meaningful properties are propagated upward through the hierarchy from the primitive tasks to the non-primitive tasks (this propagation will be explained in detail below).

Each effect (see Figure 2) comprises itself a few components of several kinds such as temporal, emotional, etc. These components may be of two kinds: non-procedural and procedural. The non-procedural (factual) component refers to the data collected from previous occurrences of the effect (contains the duration of the task, the emotions and respective intensities felt by the agent, the fuel consumed, etc., in previous executions of the task as stored in cases of plans). The procedural component refers to the process through which the temporal, emotional and other kinds of data may be computed (contains descriptions or rules of how to compute the components). Since the non-procedural component of an effect may differ in different occurrences of a task (the duration of the task may be different, the emotions may be different, etc.), effects of tasks belonging to abstract plans may store the probability distributions for each variable (see Figure 2).

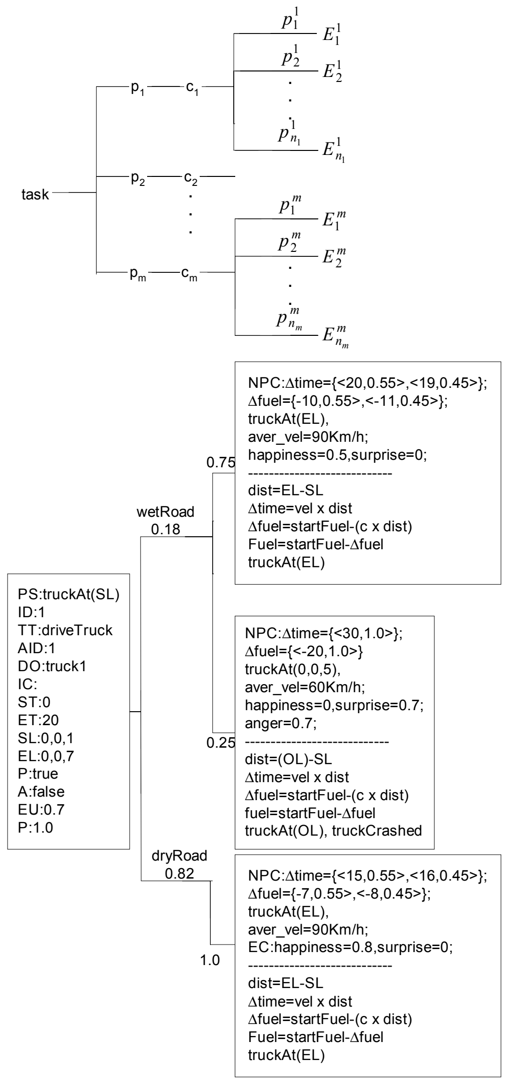

Figure 2 - Schematic representation of a task in abstract plan: general form and example.

Formally, an effect may be defined as follows.

Definition. An **effect** is a tuple *<ID*, *EC*, *EU*, *P*, *NPC*, *PC>*, where: *ID* is the identifier of the effect, i.e., an integer value that uniquely identifies the effect in the list of effects of the task; *EC* is the effect category to which it belongs (like tasks, effects are classified into categories); *EU* is the utility value (expected utility value for the case of tasks in abstract plans) of the effect; *P* is the probability value of the effect, i.e., the relative frequency of the effect (this gives us the number of times the effect occurred given that the task and the condition that triggers it occurred); *NPC* is the non-procedural component; *PC* is the procedural component.

Cases of plans share most of the features of abstract plans being also of hierarchical nature. The major differences are: unlike abstract plans, cases of plans don't have OR structures and consequently don't have abstract tasks; the primitive tasks have a probability of 1.0 (otherwise they won't belong to the case) and can only have a conditional effect since the conditions are mutually exclusive and exhaustive. Notice that, although a non-primitive task of a case of a plan may exhibit an effect, this is not relevant, since in real world only the primitive tasks are executable. However, the way a non-primitive task was decomposed is of primary importance for the generation of abstract plans as we will explain in the following section. Figure 3 shows an example of two cases of plans which are instances of the abstract plan presented in Figure 1, while Figure 4 presents an example of a primitive task which is an instance of the primitive task of an abstract plan presented in Figure 2.

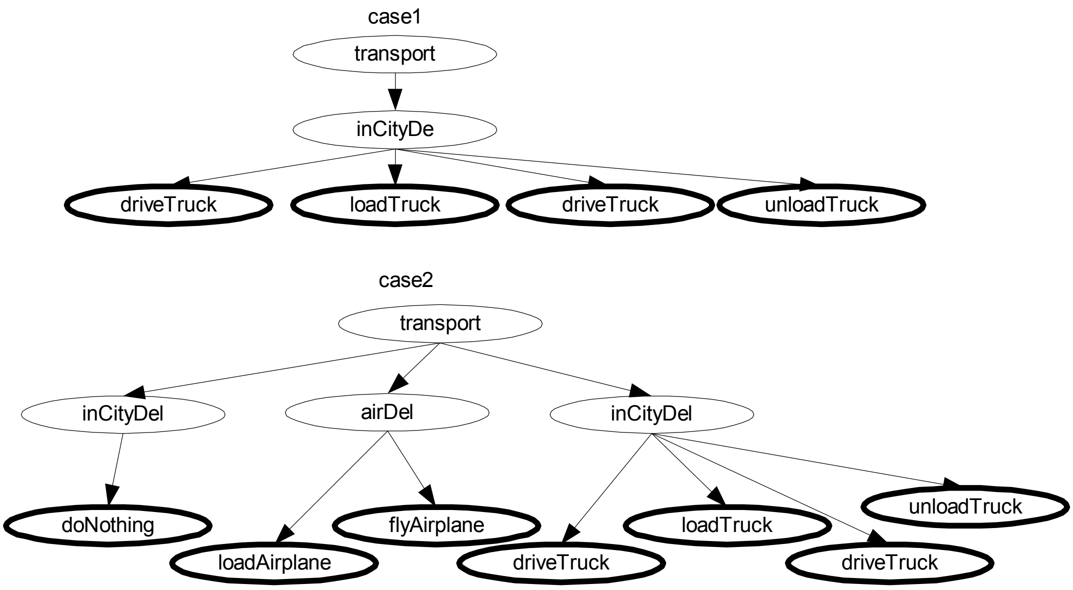

Figure 3 - Example of a case-base with two concrete plans (instances of the abstract plan of Figure 1).

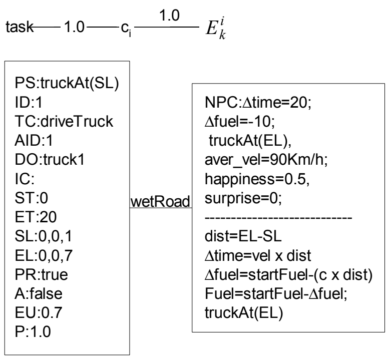

Figure 4 - Schematic representation of a task in an instance plan: general form and example.

3 Plan Generation

Since the planner is used by an agent that is part of a multi-agent environment, in order to solve a planning problem, the agent should have in memory the information of the initial state of the environment. This comprises a three-dimensional metric map of the environment (Thrun, 2002) in which inanimate and other animate agents are spatially represented. Figure 5 presents an example of a metric map that represents an initial state of world.

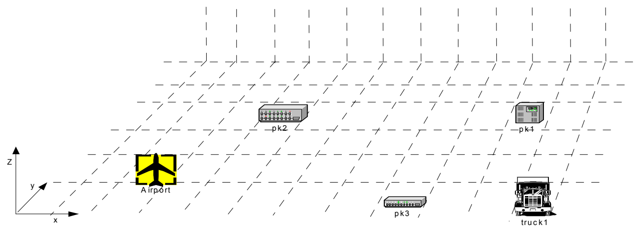

Figure 5 – Example of the metric map of an initial state of the environment in the logistics domain. It comprises: one truck (*truck1*) located at coordinates (11,0,0); three packages, *pk1*, *pk2* and *pk3*, located at, respectively, (8,0,0), (10,3,0) and (4,3,0); and, one plane located at the airport with coordinates (2,1,0).

A problem is an initial and incomplete HTN, i.e., a set of goal tasks. Planning is a process by which that initial HTN is completed resulting an abstract plan ready to be executed and incorporating alternative courses of action, i.e., it includes replanning procedures. Roughly speaking, this involves the following steps: first, the structure of the abstract plan (HTN) is built based on cases of past plans (this is closely related to the regular HTN planning procedure); then the conditional effects, probabilities are computed based on the primitive tasks of cases of past plans; the EU is computed for the primitive tasks of this abstract plan based on the procedural or non-procedural components of their effects; finally, these properties (conditional effects and respective probabilities, and EU) are propagated upward in the HTN, from the primitive tasks to the main task of the HTN. Figure 6 presents this algorithm.

Algorithm CONSTRUCT-ABSTRACT-PLAN(*abstPlan*)
abstPlan ← BUILD-STRUCTURE(*abstPlan*)
primTasks ← getPrimTasks(*abstPlan*)
primTasksAllPlanCases← getPrimTasksAllPlanCases()
COMPUT-PRIMTASKS-PROPS(*primTasks*,*primTasksAllPlanCases*)
abstPlan←PROPAGAT-PROPS-UPWARD(*primTasks*,*abstPlan*)
return *abstPlan*
end

Figure 6 - Algorithm for the construction of an abstract plan.

3.1 Building the Structure of the Abstract Plan

Much like regular HTN planning, building the abstract plan is a process by which the initial HTN is completed through the recursively decomposition of its compound tasks. Unlike regular HTN planning, within our approach the domain theory (methods and operators in regular HTN planning) is confined to a finite set of actions/operators. Thus there are no explicit methods to describe how to decompose a task into a set of subtasks. Actually, methods are implicitly present in cases of past plans (see (Muñoz-Avila et al., 2001) for a similar approach). This is particularly useful in domains where there is no theory available. Therefore, the process of decomposing a task into subtasks is case-based and is performed as follows. Given a task, the possible alternative decompositions (task and its subtasks, as well as the links between them) are retrieved from cases of past plans. Two situations might happen. If there are more than one alternative decomposition, the given task is set as abstract and the set of decompositions are added to the HTN, linking each head task to the abstract task through a hierarchical link of type *abst*. Thus these head tasks are now the subtasks of the abstract task (see Figure 7 for an illustration of this process). On the other hand, if only one decomposition is retrieved, its subtasks are added as subtasks of the given task, linked by a hierarchical link of type *dcmp* (see Figure 8 for an illustration of this process). Whether a single decomposition or multiple decompositions are retrieved, the addition of it/them comprises an adaptation process (Kolodner, 1993), i.e., the retrieved decomposition(s) is/are changed if necessary so that it/they is/are consistent with the rest of the HTN.

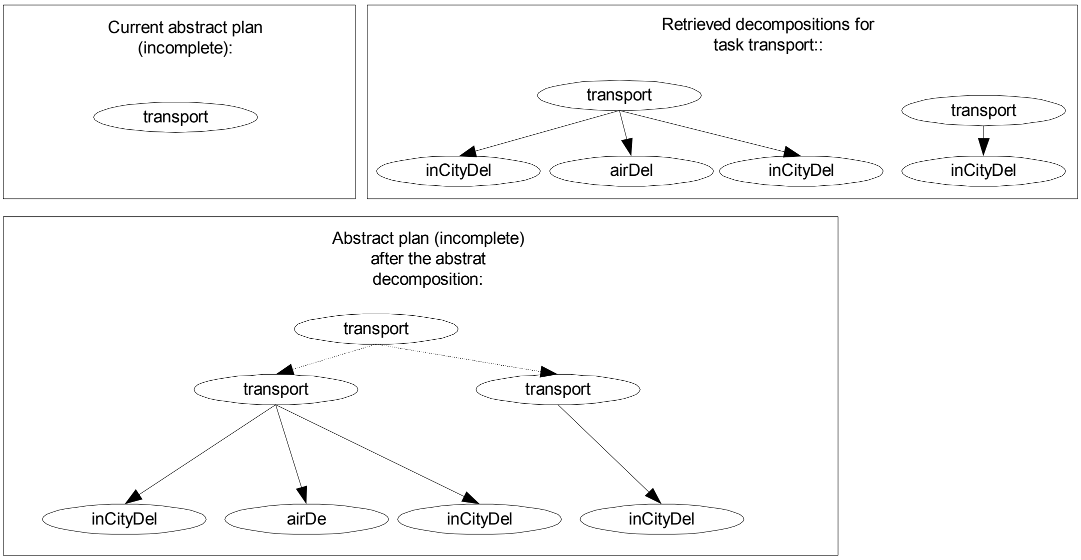

Figure 7 - Illustrative example of an OR-decomposition of an abstract task.

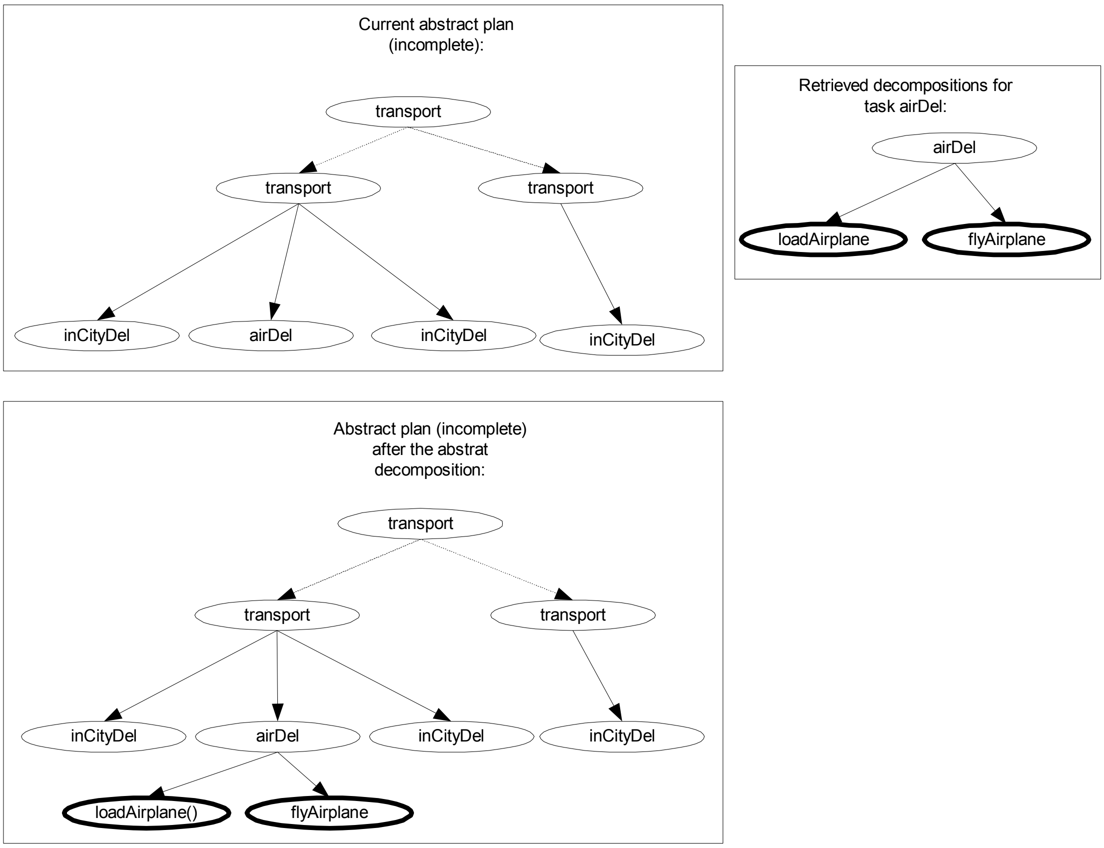

Figure 8 - Illustrative example of an AND-decomposition of a regular compound task.

The process of building the HTN ends when there is no more compound tasks to decompose, i.e., when the leaves of the tree are primitive tasks, or when there is no available decompositions in the case-base for at least one compound task.

Within our approach, a task belonging to an HTN has a probability value associated to it. This value expresses the probability of being executed given that its ancestor is executed. Thus, this probability is actually a conditional probability. Obviously, the probability of a task belonging to a case of a past plan is always 1.0 because it was executed (otherwise it won't belong to the case). The probability of the tasks belonging to an abstract plan is computed during the process of building the HTN as follows. Given the i^{th} subtask, ST_i, of a task T both belonging to an abstract plan, the probability of ST_i be executed given that T is executed is given by the conditional probability formula $P(ST_i / T) = \frac{P(ST_i \cap T)}{P(T)}$. Since within our approach there is no probabilistic model available, these probabilities have to be computed from data, i.e., from past occurrences of the tasks in cases of past plans, in the following manner. According to the frequency interpretation of probability, in r repetitions of an experiment, the value $P(X)$ is given by the number of times X occurred in the possible r times. This value is given by $S_r(X)/r$, where $S_r(X)$ denotes the absolute frequency of X (i.e., the number of times X occurred in the r repetitions of the experiment). As r increases, $S_r(X)/r$ converges to $P(X)$. In the context of HTN planning, the experiment should be understood as the decomposition of a task into subtasks. According to this frequentist approach of probability it can be shown that, $P(ST_i / T) = \frac{P(ST_i \cap T)}{P(T)} = \frac{S_r(ST_i \cap T)}{S_r(T)}$, when r is big. Thus, this expresses the number of times ST_i and T occurred together in the total amount of times T occurred, or in the context of HTN planning, this expresses the number of times ST_i was subtask of T in the total amount of times T was the task decomposed in past HTN plans. When ST_i is not a head of an alternative decomposition in the new plan (i.e., when T is not an abstract task), it means that T was always decomposed in the same way in past plans, i.e., into the same subtasks, which means ST_i occurred always when T occurred, otherwise ST_i won't be subtask of T. Thus, in this situation, the numerator and denominator of the above equation are equal and therefore $P(ST_i/T)$=1.0. However, when ST_i is a head of an alternative decomposition, it means there were more than one way to decompose T in past plans, one of them being the decomposition headed by ST_i. Thus, counting the number of times the decomposition headed by ST_i was taken to decompose T, i.e., the number of times ST_i instantiated T, $S_r(ST_i \cap T)$, in all past plans and dividing this number by the number of times T was decomposed, i.e., $S_r(T)$, yields the value for $P(ST_i/T)$ for this situation.

After the abstract HTN is built, the conditional effects (and respective probabilities) and the EU are computed for the primitive tasks.

3.2 Motivation and Emotion-based Computation of the EU

As said above, a task T is both conditional and probabilistic (e.g.: (Blythe, 1999)). Thus, the execution of a goal task under a given condition may be seen according to Utility Theory as a lottery (Russel & Norvig, 1995):

$$Lottery(T) = \left[p^1 \times p_1^1, E_1^1; p^1 \times p_2^1, E_2^1; ...; p^m \times p_{n_m}^m, E_{n_m}^m\right]$$

, where p^i is the probability of the condition c_i, p_j^i is the probability of the j^{th} effect, E_j^i, of condition c_i.

The EU of T may be then computed as follows:

$$EU(T) = \sum_{k,j} p^k \times p_j^k \times EU(E_j^k)$$

The computation of $EU(E_j^k)$ is performed predicting the motivations that could be elicited by achieving/executing the goal task (Castelfranchi, Conte, Miceli, & Poggi, 1996; Reisenzein, 1996). We confined the set of motivations to surprise, curiosity and hunger[1]. As said above, two methods may be used for predicting the intensities of those motivations: based on the non-procedural component of the effects, or based on the procedural component.

If we take into account the procedural component of the effects, the intensities of surprise, curiosity and hunger felt by the agent when the effect takes place are estimated based on the information available in the effect about the changes produced in the world.

Surprise is given by (Macedo & Cardoso, 2001a):

$$SURPRISE(Agt, Obj_k) = UNEXPECTEDNESS(Obj_k, Agt(Mem)) =$$
$$= 1 - P(Obj_k)$$

, where Obj_k is the direct object of task T when E_j^k takes place, i.e., the entity that is visited (for the case of exploratory behaviour).

Curiosity is computed as follows (Macedo & Cardoso, 2001b):

$$CURIOSITY(Agt, Obj_k) = DIFFERENCE(Obj_k, Agt(Mem))$$

The measure of difference relies heavily on error correcting code theory (Hamming, 1950): the function computes the distance between two objects represented by graphs, counting the minimal number of changes (insertions and deletions of nodes and edges) required to transform one graph into another.

The drive hunger is defined as the need of a source of energy. Given the capacity C of the storage of that source, and L the amount of energy left ($L \leq C$), the hunger elicited in an agent is computed as follows:

$$HUNGER(Agt) = C - L$$

The following function is used to compute $EU(E_j^k)$:

$$EU(E_j^k) =$$
$$= \frac{\alpha_1 \times U_{surprise}(E_j^k) + \alpha_2 \times U_{curiosity}(E_j^k) + \alpha_2 \times U_{hunger}(E_j^k)}{\sum_i \alpha_i} =$$
$$= \frac{\alpha_1 \times SURPRISE(E_j^k) + \alpha_2 \times CURIOSITY(E_j^k) + \alpha_2 \times HUNGER(E_j^k)}{\sum_i \alpha_i}$$

, where, α_2 = -1 and α_i ($i \neq 2$) may be defined as follows:

$$\alpha_i = \begin{cases} 1 \Leftarrow C - HUNGER(Agt) - D > 0 \\ 0 \Leftarrow otherwise \end{cases}$$

, where D is the amount of energy necessary to go from the end location of goal task T to the closer place where energy could be recharged, and C is the maximum amount of energy that could be stored by the agent.

If we take into account the non-procedural component of the effects, we avoid the computations of the intensities of the motivations. In fact, doing so, we are taking into account the intensities of the emotions, drives and other motivations in previous occurrences of the tasks and respective effects. This emotional/motivational information collected from previous occurrences of a task is a kind of Damásio's *somatic marker*. For this reason, tasks are called *somaticly-marked tasks*. When a task is about to occur again, the planning agent may compute its EU based on this data. In fact, this seems to be faster than the alternative approach of estimating the emotions that a task may elicit based on the values of the variables of the state of the world such as the time duration, fuel consumed, etc. Anyway, the same formula (present above) is used to compute $EU(E_j^k)$.

3.3 Propagation of the Properties Upward

After the primitive tasks have the conditional effects and respective probabilities, the probability and EU computed, these properties are propagated bottom-up (from primitive to non-primitive tasks), from the sub-

[1] The agents that make use of the planning approach described in this paper have been used to explore unknown environments, and to create things. Among motivations, surprise, curiosity and hunger have been closely related with this exploratory and creative behaviour (Berlyne, 1950; Boden, 1995; Izard, 1991).

tasks to the task of a decomposition and from the subtasks (heads of alternative decompositions) to the abstract task of an abstract decomposition). Notice however that the goal of this propagation is twofold: to complete the non-primitive tasks so that they can be ranked according to their EU when they are heads of alternative decompositions, and to know the overall EU of the abstract plan which is given by the EU of the main task of the plan.

4 Plan Execution and Replanning

Finding the optimal plan consists simply of traversing the abstract plan, selecting the most EU subtask of an abstract task. Backtracking occurs when an alternative decomposition fails execution. In this case, the next alternative decomposition that follows the previous in the EU ranking is selected for execution.

5 Related Work

Our work is closely related to HTN planning. This methology has been extensively used in planning systems such as UMCP (Erol et al., 1994), SHOP and SHOP2 (Nau et al., 2001). Unlike these planners, the planner presented in this paper don't use methods as part of the domain theory for task decomposition, but instead methods that are implicitly included in cases that describe previous planning problem solving experiences. SiN (Muñoz-Avila et al., 2001) also uses a case-based HTN planning algorithm, in which cases are instances of methods.

Among decision-theoretic planners, DRIPS (Haddawy & Doan, 1994) is probably the most closely related to the planner presented here. Actually, DRIPS shares a similar representation approach for abstract plans (an abstraction/decomposition hierarchy) and for actions. Besides, it also returns the optimal plan according to a given utility function. However, in contrast to DRIPS, in our planner the variant of a HTN that represents abstract plans is automatically built from cases and not given as input for the planning problem. Besides, it includes temporal, utility ranking and adaptation links in addition to decomposition links. Another major difference is that, in our planner, the EU of tasks and of alternative plans are computed when the abstract plan is built, while in DRIPS this occurs when the optimal plan is searched. Besides, in our planner, there is the possibility of computing the EU of tasks based on the non-procedural component of their effects, which avoids some additional computations at the cost of being less accurate. Moreover, finding the optimal plan in our planner consists simply of traversing the HTN with backtracking (or replanning) points located at the subtasks of an abstract task. In our planner the propagation of properties upward in the hierarchy is closely related with the approach taken in DRIPS for abstracting actions (Haddawy & Doan, 1994). A propagation of properties in the planning tree, bottom-up and left-to-right, is also used in GraphHTN (Lotem & Nau, 2000) in order to improve the search algorithm.

Another important work that addressed planning in agents inhabiting dynamic, uncertain environments is that of (Wilkins, Myers, & Wesley, 1994).

The relationship between emotions and plans has been considered previously by several authors (e.g.: (Bates, 1994; Gratch, 1999; Oatley & Johnson-Laird, 1987; Simon, 1967; Sloman, 1987)). Our main additional contribution to this works is considering somaticly-marked tasks (Damásio, 1994).

6 Conclusions and Future Work

We have presented an approach for decision-theoretic, HTN planning. In this approach emotions and motivations play a central role in that the EU of the tasks is based on the intensity of the emotions and other motivations they elicit. Two approaches have been proposed to compute the EU of tasks based on motivations: based on the procedural or non-procedural (factual) component of the effects of the tasks. The latter approach seems to be faster and is deeply related with Damásio's somatic-marker hypothesis. However, additional experiments are required to assess these ideas. In the future, we plan to perform such experiments.

Aknowledgements

The PhD of Luís Macedo is financially supported by PRODEP III.

References

Allen, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832-- 843.

Bates, J. (1994). The Role of Emotion in Believable Agents. *Communications of the ACM, 37*(7), 122-125.

Berlyne, D. (1950). Novelty and curiosity as determinants of exploratory behavior. *British Journal of Psychology, 41*, 68-80.

Blythe, J. (1999). Decision-Theoretic Planning. *AI Magazine, Summer 1999*.

Boden, M. (1995). Creativity and unpredictability. *SEHR, 4*(2).

Castelfranchi, C., Conte, R., Miceli, M., & Poggi, I. (1996). Emotions and goals. In B. Kokinov (Ed.), *Perspectives on Cognitive Science* (pp. 131-145). Sofia: New Bulgarian University.

Damásio, A. (1994). *Descartes'error, Emotion Reason and the Human Brain*. New York: Grosset/Putnam Books.

Erol, K., Hendler, J., & Nau, D. (1994). UMCP: A sound and complete procedure for hierarchical task-network planning, *Proceedings of the International Conference on AI Planning Systems* (pp. 249-254).

Gratch, J. (1999). Why you should buy an emotional planner, *Proceedings of the Agents'99 Workshop on Emotion-based Agent Architectures*.

Haddawy, P., & Doan, A. (1994). Abstracting probabilistic actions, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 270-277). San Mateo, CA: Morgan Kaufmann.

Hamming, R. (1950). Error Detecting and Error Correcting Codes. *The Bell System Technical Journal, 26*(2), 147-160.

Izard, C. (1991). *The Psychology of Emotions*. NY: Plenum Press.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan-Kaufmann.

LeDoux, J. (1996). *The Emotional Brain*. New York: Simon and Schuster.

Littman, M., & Majercik, S. (1997). Large-Scale Planning Under Uncertainty: A Survey, *Workshop on Planning and Scheduling for Space* (pp. 27:21--28).

Lotem, A., & Nau, D. (2000). New advances in GraphHTN: Identifying independent subproblems in large HTN domains, *dings of the International Conference on AI Planning Systems* (pp. 206-215).

Macedo, L., & Cardoso, A. (1998). Nested-Graph structured representations for cases. In B. Smyth & P. Cunningham (Eds.), *Advances in Case-Based Reasoning - Proceedings of the 4th European Workshop on Case-Based Reasoning* (Vol. 1488, pp. 1-12). Berlin: Springer-Verlag.

Macedo, L., & Cardoso, A. (2001a). Modelling Forms of Surprise in an Artificial Agent. In J. Moore & K. Stenning (Eds.), *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 588-593). Mahwah, NJ: Erlbaum.

Macedo, L., & Cardoso, A. (2001b). SC-EUNE - Surprise/Curiosity-based Exploration of Uncertain and Unknown Environments., *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing* (pp. 73-81). York, UK: University of York.

Meyer, W., Reisenzein, R., & Schützwohl, A. (1997). Towards a process analysis of emotions: The case of surprise. *Motivation and Emotion, 21*, 251-274.

Muñoz-Avila, H., Aha, D., Nau, D., Breslow, L., Weber, R., & Yamal, F. (2001). SiN: Integrating Case-based Reasoning with Task Decomposition, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*. Seattle, WA: Morgan Kaufmann.

Nau, D., Muñoz-Avila, H., Cao, Y., Lotem, A., & Mitchell, S. (2001). Total-order planning with partially ordered subtasks, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*. Seattle, WA: Morgan Kaufmann.

Oatley, K., & Johnson-Laird, P. (1987). Towards a cognitive theory of emotions, cognition and emotion. *Cognition and Emotion, 1*(1), 29-50.

Ortony, A., & Partridge, D. (1987). Surprisingness and Expectation Failure: What's the Difference?, *Proceedings of the 10th International Joint Conference on Artificial Intelligence* (pp. 106-108). Los Altos, CA: Morgan Kaufmann.

Reisenzein, R. (1996). Emotional Action Generation. In W. Battmann & S. Dutke (Eds.), *Processes of the molar regulation of behavior*. Lengerich: Pabst Science Publishers.

Reisenzein, R. (2000). The subjective experience of surprise. In H. Bless & J. Forgas (Eds.), *The message within: The role of subjective experience in social cognition and behavior*. Philadelphia, PA: Psychology Press.

Russel, S., & Norvig, P. (1995). *Artificial Intelligence - A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.

Simon, H. (1967). Motivational and emotional controls of cognition. *Psychological Review, 74*, 29-39.

Sloman, A. (1987). Motives, mechanisms and emotions. *Cognition and Emotion, 1*, 217-234.

Thrun, S. (2002). Robotic mapping: A survey. In G. Lakemeyer & B. Nebel (Eds.), *Exploring Artificial Intelligence in the New Millenium*. San Mateo, CA: Morgan Kaufmann.

Wilkins, D., Myers, K., & Wesley, L. (1994). Cypress: Planning and Reacting under Uncertainity. In M. Burstein (Ed.), *ARPA/Rome Laboratory Planning and Scheduling Initiative Workshop Proceedings* (pp. 111-120). San Mateo, CA: Morgan Kaufmann Publishers Inc.

Younes, H. (2003). Extending PDDL to model stochastic decision processes, *Proceedings of the ICAPS-02 Workshop on PDDL*.