CZECH TECHNICAL UNIVERSITY IN PRAGUE

DOCTORAL THESIS STATEMENT

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics

Ing. Karel Kohout

MULTI-LEVEL STRUCTURE OF ANTICIPATORY BEHAVIOUR IN ALIFE

Ph.D. Programme: Electrical Engineering and Information Technology

Branch of study: Artificial Intelligence and Biocybernetics

Doctoral thesis statement for obtaining the academic title of "Doctor",
abbreviated to "Ph.D."

Prague, June 2011

The doctoral thesis was produced in full-time manner

Ph.D. study at the department of Cybernetics of the Faculty of Electrical Engineering of the CTU in Prague

Candidate: ***Ing. Karel Kohout***
Department of Cybernetics (K13133)
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 16627 Praha 6

Supervisor: ***Doc. Ing. Pavel Nahodil, CSc.***
Department of Cybernetics (K13133)
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6

Supervisor-Specialist: ***Prof. Ing. Vladimír Mařík Dr.Sc.***
Department of Cybernetics (K13133)
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6

Opponents:

....

....

....

The doctoral thesis statement was distributed on: ....

The defence of the doctoral thesis will be held on .... **at** .... **a.m./p.m. before the Board for the Defence of the Doctoral Thesis in the branch of study Artificial Intelligence and Biocybernetics in the meeting room No.** .... **of the Faculty of Electrical Engineering of the CTU in Prague.**

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

Prof. Ing. Vladimír Mařík Dr.Sc.

Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study Artificial Intelligence and Biocybernetics
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6.

1. STATE OF THE ART

This chapter of the thesis statement describes the current situation of the studied problem. The area my work is dealing with is theory of anticipatory behaviour and its applications usable for Artificial Life (ALife). This area is still considered one of the so far unresolved topics of Artificial Intelligence.

Nature evolves in a continuous anticipatory fashion targeted at survival. Sometimes we humans are aware of anticipation, as when we plan. Often, we are not aware of it, as when processes embedded in our body and mind take place before we realize their finality. We can take an example from any sport or game which requires precise and fast body movement. For example in tennis the return of a professional serve can be successful only through anticipatory mechanisms. Even very fast but conscious reaction takes too long to process. With anticipation we start the action even before the event that would normally trigger this action occurs. Creativity in art and design are fired by anticipation. Before the archer draws his bow his mind has already hit the target. Motivation mechanisms in learning, the arts, and all types of research, are dominated by the underlying principle that a future state controls the present action, aimed at some goal. The entire subject of prevention entails anticipatory mechanisms. I could continue in naming all the areas of life where we can find a trace of anticipatory principles. It is true that an overwhelming part of every being's everyday behaviour is based on the tacit employment of predictive models.

1.1. Anticipation

There are several definitions and descriptions of anticipation, some of them are just broadening the initial definition of Robert Rosen [1]. These definitions are not in contradiction, they describe anticipation from different points of view. Over the last few decades research in anticipation advanced rapidly but not only in ALife domain. Experimental psychology research gradually started to accept the notion of anticipations beginning with Tolman's suggestion of "expectancies" [2] due to his observation of latent learning in rats (learning of environmental structure despite the absence of reinforcement). More recently an outcome devaluation procedure [3] has been employed that provides definite evidence for anticipatory behaviour in animals. The most recent works that inspired me in this chapter were the works of Martin V. Butz [4], Daniel Dubois [5] and Carlos Martinho [6].

I would like to point out that a significant work was done on the field of anticipation and there were several accomplishments published. As one example for all I will name a conference held each two years and dedicated to anticipation named Computing Anticipatory Systems (CASYS). This conference organized and chaired by Daniel Dubois has been held since 1998 and has become an excellent opportunity for researches in this area to exchange opinion. I consider it to be an honour that my article was accepted and published on this conference in 2009 [7].

1.1.1. *The Basics Of Anticipation*

Basic definition of anticipatory systems was published in 1985 by the biocyberneticist Robert Rosen in his book Anticipatory systems [1]. He defined an anticipatory system as follows: "*A system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a latter instant*". Rosen in his book was inspired by his observation of live organisms, namely the ones with higher intelligence. Especially by their ability to predict the future and make adaptations based on them. This ability of live beings was already discovered before. Rosen however utilizing this knowledge, created a theory which was abstracted for various systems in the following way. Rosen in his work exposed a recurring basic pattern of causality and laws, arising initially in physics and generalized over the years stating that: *"in any law governing a natural system, it is forbidden, to allow present changes of state to depend upon future features"* ([1], page 9). This law is widely followed in technical sciences such as physics or control theory. Past states are allowed (in systems with memory) but not the future states. This may seem like a denial of causality and thus it appears to be an attack on the ultimate basis on which science itself rests, while as a matter of fact it is not the case. If we consider the behaviour of a system which contains a predictive model and which can utilize the predictions of its model to modify its present behaviour. If we further suppose that the model can approximate by its predictions the future events with a high degree of accuracy then this system will behave as if it was a true anticipatory system (i.e. a system of behaviour that depends on future states). So we do not have the present state available only its estimate, and this estimate is not based on the information about the future state but on information from past and current states. This system will not violate our notions of causality, but since we explicitly forbid present changes of states to depend on future states, we will be driven to understand the behaviour of such a system in a purely reactive mode (i.e. one in which present change of state depends only on present and past states). Since we claimed that the information we can derive about future can be based only on present and past information we respected the causality. Let's describe this in a more formal way to clarify the thoughts. Let us suppose that we are given a system *S*, which is the system of interest. For the sake of simplicity let us consider that *S* is a non-anticipatory dynamic continuous system. We will associate another dynamic system *M* with system *S*, where *M* is a model of *S*. We require that *S* is parameterized in real time and that *M* is parameterized by a time variable that goes quicker than that. In this way, the behaviour of *M* predicts the behaviour of *S*. By looking at the state of *M* at time$t$, we get information about the state that *S* will be in at some time later than$t$. We shall now allow *M* and *S* to interact with each other. We shall suppose that the system *M* is equipped with a set of effectors *E*, which allow it to operate either on *S* itself, or on the environmental inputs of *S*, and change the dynamical properties of *S*. For *M* to be a consistent model, the actions operated on *S* should also be operated on *S*. Figure 1 represents such a system. If we put this system into a single box, that box will appear to us to be an adaptive system in which prospective future behaviours determine present changes of state. We will call this system an anticipatory system.

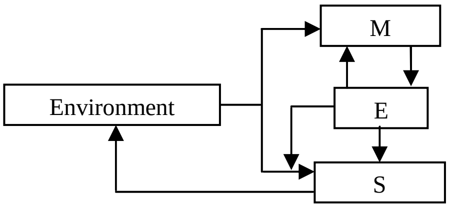

Figure 1 – Rosen's Definition of an Anticipatory System. ***S is the system of interest; M is the model of S, equipped with a set of effectors E that changes the dynamical properties of S or its environmental inputs. For consistency, these changes are also reflected in M.***

It may seem that anticipation is a matter just in biological systems and what more that it is present only in simple animals. On the contrary anticipation plays important role in all living and also non living systems. My work is focused on artificial life hence mostly concerned about the living systems. One of the researchers that noticed anticipatory behaviour even in non living systems is Daniel Dubois. The demonstration of the fundamental property of anticipation in electromagnetism is made on the well-established and well experimentally verified Maxwell Equations. It is shown that very famous physicists like Feynman, Wheeler and Dirac thought about anticipatory solutions to resolve big problems in theoretical physics. At one hand, many physical processes deal with electromagnetism, and at the other hand, many biological systems deal also with electromagnetism, like, for example, the nervous system, the brain, the heart, etc... in living systems. Robert Rosen argued that anticipation distinguishes the living systems from the non-living ones. Dubois shows that physical systems deal with strong anticipation because the anticipation is fundamentally embedded in these physical systems. Rosen's anticipatory system deals with weak anticipation, because the anticipation is based on a model of the system and thus is a model-based prediction and not a system-based prediction [8].

1.1.2. *Current Types of Anticipation*

One of the contributions my work brings is the attempt to unify and to complete the categories of anticipation. In order to build this in later chapters, it is necessary to briefly describe the current state. This chapter was composed based on the work of Martin Butz [4]. According Butz anticipations are an important and interesting concept. They appear to play a major role in the coordination and realization of adaptive behaviour. Looking ahead and acting according to predictions, expectations, and aims seems helpful in many circumstances. For example, we say that we are in anticipation, we are looking forward to events, we act goal-oriented, we prepare or get ready for expected events, etc. Despite these important approaches, it is still hardly understood why anticipatory mechanisms are necessary, beneficial, or even mandatory in our world. It might be true that over all constructible learning problems any learning mechanism will perform as good, or as bad, as any other one, the psychological findings suggest that in natural environments and natural problems learning and acting in an anticipatory fashion increases the chance of survival. Thus, in the quest of designing competent artificial animals, the so called animats, the incorporation of anticipatory mechanisms seems mandatory.

Without a conceptual understanding of what anticipatory behaviour is referring to, scientific progress towards more elaborate and competent anticipatory behaviour systems is hard to achieve. The term anticipation is often understood as a synonym for prediction or expectation - the simple act of predicting the future or expecting a future event or imagining a future state or event. Anticipation really is about the impact of a prediction or expectation on current behaviour. Thus, anticipation means more than a simple look ahead into the future. The important characteristic of anticipation that is often overlooked or misunderstood is the impact of the look into the future on actual behaviour. We do not only predict the future or expect a future event but we alter our behaviour - or our behavioural biases and predispositions - according to this prediction or expectation. Here we are moving in definition from anticipation towards anticipatory behaviour. This is the very core of ALife research, the behaviour is the main area of interest. Butz defines the anticipatory behaviour as follows: *A process, or behaviour, that does not only depend on past and present but also on predictions, expectations, or beliefs about the future.* In fact, any “intelligent” process can be understood as exhibiting some sort of anticipatory behaviour in that the process, by its mere existence, predicts that it will work well in the future. This implicit anticipatory behaviour can be distinguished from explicit anticipatory behaviour in which current explicit future knowledge is incorporated in some behavioural process. This defines two very intuitive categories of anticipation.

Implicitly anticipatory animat-type is the one in which no predictions whatsoever are made about the future that might influence the animat’s behavioural decision making. Sensors input, possibly combined with internal state information, is directly mapped onto an action decision. The predictive model of the animat is empty or does not influence behavioural decision making in any way. One of the reasons for this might be memory limitations. Moreover, there is no action comparison, estimation of action benefit, or any other type of prediction that might influence the behavioural decision. In nature, even if a life form behaves purely reactively, it still has implicit anticipatory information in its genetic code in that the behavioural programs in the code are (implicitly) anticipated to work in the offspring.

If an animat considers predictions of the possible payoff of different actions to decide on which action to execute, it may be termed **payoff anticipatory**. In these animats, predictions estimate the benefit of each possible action and bias action decision making accordingly. No state predictions influence action decision making. There is no explicit predictive model however the learned reinforcement values estimate action payoff. Thus, although the animat does not explicitly learn a representation with which it knows the actual sensed consequences of an action, it can compare available action choices based on the payoff predictions and thus act payoff anticipatory.

While in payoff anticipations predictions are restricted to payoff, in **sensory anticipations** predictions are unrestricted. However, sensory anticipations do not influence the behaviour of an animat directly but sensory processing is influenced. The prediction of future states and thus the prediction of future stimuli influence stimulus processing. As will be shown later, comparison of the expected value with the actual value can be used to focus attention as well as to produce *emotions*. Expected sensory input might be processed faster than unexpected input or unexpected input with certain properties (for example possible threat) might be reacted to faster.

Maybe the most interesting group of anticipations is the one in which animat behaviour is influenced by explicit future state representations. As in sensory anticipations, a predictive model must be available to the animat or it must be constructed by the animat. In difference to sensory anticipations, however, **state anticipations** directly influence current behavioural decision making. This means that the predicted future state(s) directly influences the actual action selection.

1.1.3. *Strong and Weak Anticipation*

This chapter was based on the work of Daniel Dubois [5], [8]. In his work he deals with some mathematical developments to model anticipatory capabilities in discrete and continuous systems. He also noticed that even non-living systems without any possibility of construction of a model (like electromagnetism and relativity transformations) exhibits some anticipatory behaviour. Dubois puts a tentative definition of anticipation: *An anticipatory system is a system for which the present behaviour* *is based on past and/or present events but also on future events built from these past, present and future events. Any anticipatory system can obey, as any physical systems, the Maupertuis least action principle.*

In view of explicitly mathematically defining systems with anticipation, Dubois introduced the concept of incursion, an inclusive or implicit recursion. An incursive system is a recursive system that takes into account future states for evolving. Some nonlinear incursive systems show several potential future states, that he called hyperincursion. A hyperincursive anticipatory system generates multiple potential states at each time step and corresponds to one-to-many relations. A selection parameter must be defined to select a particular state amongst these multiple potential states. Here we can apply criteria to select the best states from the potential states. These multiple potential states collapse to one state (among these states) which becomes the actual state the anticipation of a system can be based on a model of its environment.

In this case, the notion of exo-anticipation is introduced, with the following definition: *An exo-anticipation is an anticipation made by a system about external systems. In this case, anticipation is more related to predictions or expectations.* This defines a weak anticipation.

The anticipation of a system can be based on itself, rather than its environment. In this case, the notion of endo-anticipation is introduced, with the following definition: *An endo-anticipation is an anticipation built by a system or embedded in a system about its own behaviour.* This is not a predictive anticipation anymore but a built anticipation. In this case, this is a strong anticipation.

1.2. Anticipatory Classifier System

One of the most successful approaches using Markov chain theory is anticipatory modification of *Learning Classifier System* (LCS) invented in 1975 by John Holland [10]. All LCSs have in common that they are rule-based systems able to automatically build the rule set they work on [9]. LCSs are based on two fundamental mechanisms - *Genetic Algorithms* (GAs) and *Reinforced Learning* (RL). The anticipatory modification of these is called *Anticipatory Classifier System* (ACS).

ACS consists of a set of rules called *classifiers* combined with adaptive mechanisms in charge of evolving the population of rules. Classical Reinforced Learning (RL) algorithms such as Q-learning rely on an explicit enumeration of all the states of the system. But, since they represent the state as a collection of a set of sensations called *attributes*, ACSs do not need this explicit enumeration thanks to a generalization property that will be described later on. This generalization property has been recognized as the distinguishing feature of ACSs with respect to the classical RL framework.

An LCS is composed of a population of classifiers. Each classifier is a triple *<c, a, p>* containing a *[Condition]* part, an *[Action]* part, and an estimation of the expected accumulated reward that the agent can get if it fires this classifier. The *c* and *a* represent the condition and action of the agent, and *p* the current estimate of the long term reward that the agent can expect from this *(s, a)* pair. Formally, the *[Condition]* part of classifiers is a list of tests. There are as many tests as attributes in the problem description, each test being applied to a specific attribute. In the most common case where the test specifies a value that an attribute must take for the *[Condition]* to match, the test is represented just by this value. There exists a particular test, denoted as "#" and called "don't care", which means that the *[Condition]* of the classifier will match whatever the value of the corresponding attribute. At a more global level, the *[Condition]* part of a classifier matches if all its tests hold in the current situation. In such a case, the classifier can be fired. After describing the representation manipulated by LCSs, we must present their mechanisms. The general goal is to design an RL system, thus there will be at its heart an action selection mechanism relying on the value of all actions in different situations. Furthermore, these systems are endowed with a generalization capability which relies on classifier population evolution mechanisms in order to reach a satisfactory level of generality. I present both categories of mechanisms in the next sections and I will show afterward that families of systems can be distinguished by the way they deal with interactions between these mechanisms. The set of classifiers whose *[Condition]* part matches the current situation is called the "match-set" and denoted *[M]*. Furthermore, we denote by *[A]*, the "action-set", the set of classifiers in *[M]* which advocate the action *a* that is actually chosen. Given the generalization property of classifiers, the *[Condition]* part of several classifiers can match at the same time, while they do not necessarily specify the same action. Thus, LCSs must contain an action selection mechanism which chooses the action executed given the list of classifiers in *[M]*. In order to benefit from RL properties, this mechanism must use the expected accumulated reward of each classifier, but it must also include some trade-off between exploration and exploitation.

Ensuring that each classifier reaches the ideal generalization level is a crucial concern in LCSs. The system must find a population which covers the state space as compactly as possible, without being detrimental to the optimality of behaviour. The mechanisms responsible for this property differ from one system to the other, but they all rely on adding and deleting classifiers. In the case of anticipation-based systems, more deterministic generalization and specialization heuristics are being used.

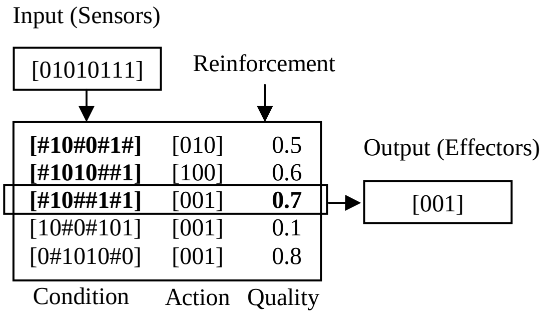

Figure 2 – Learning Classifier System Example

Although they share a number of common characteristics ACSs deviate from the classical framework on one fundamental point. Instead of *[Condition] → [Action]* classifiers, they manipulate *[Condition] [Action] → [Effect]* classifiers. The *[Effect]* part represents the expected effect (next state) of the *[Action]* part in all situations that match the *[Condition]* part of the classifier. Such a set of classifiers constitutes what is called in the RL literature a model of transitions. Since they learn a model of transitions, ACSs are an instance of model-based RL architecture. As a result, ACSs can be seen as combining two crucial properties of RL systems. First property is that they learn a model of transitions, which endows them with anticipation and planning capabilities and speeds up the learning process. The second is that they are endowed with a generalization property, which lets them build much more compact models. The first design of ACS was introduced by Stolzmann [11]. ACS was later extended by Butz to become ACS2 [12]. ACS use classical solutions to deal with the exploration versus exploitation trade-off. The agent first chooses actions bringing more information about the transitions that have not been tried enough. Then, if the best actions are equivalent with respect to the first criterion, it chooses actions bringing more external reward, as any RL system does. Finally, if the best actions are equivalent with respect to the first and second criteria, it chooses actions that have not been tried for the longest time, so as to handle non-stationary environments as efficiently as possible. In order to obtain a model of transitions as general, accurate and compact as possible, ACSs generally rely on the combination of two heuristics. A specialization heuristic is applied to inaccurate classifiers and a generalization heuristic is applied to overspecialized classifiers. When appropriate, the combination of both heuristics results in the convergence of the population to a maximally general and accurate set of classifiers. For the specialization process, all ACSs rely on the same idea. When a general classifier oscillates between correct and incorrect predictions, it is too general and must be specialized. Its *[Condition]* part must be modified so as to match only in situations where its prediction is correct. ACS randomly chooses a # test and changes it into a specialized test. The generalization process is more complex. Usually in ACS a GA is used to replace specific classifiers with more general ones.

1.3. Emotivector

Emotivector was proposed and described by Carlos Martinho in his dissertation thesis [6]. The emotivector architecture is based on four main ideas: (a) to be based on the software agent architecture, (b) to not alter or interrupt the flow of the agent architecture, (c) to be transparently addable or removable from the agent architecture, (d) to be usable in both symbolic and sub-symbolic processing models. The architecture design builds above the Russell and Norvig architecture [13], which is an approach used in most of the designs nowadays where an agent perceives its environment through its sensors (e.g. sns_i) and acts upon that environment through effectors (e.g. eff_i). This basic design is used even by me and my current architecture. It is typically composed of three phases, executed as a sequence or running in parallel. Sensing that is providing the agent with percepts translated by the sensors from the environment signals according their capabilities. Processing that is mapping the percepts and constructs into a set of effector actions and updating the current constructs. And as last step acting that is modifying the environment through the agent effector actions, within their limitations. Graphical representation of Russell-Norvig architecture together with block modification by Martinho is shown on Figure 3.

The approach that Martinho used was enriching this architecture by a semi-autonomous module he called salience module. This will perform context-free monitoring of the percepts flowing from the sensors to the processing module as well as of the action commands flowing from the processing module to the agent effectors. In more detail the information flowing from the sensors to the processing module of the agent is observed by the salience module that computes its a-priori salience. Each sensor (sns_i) is associated with an emotivector (emo_i) that computes a context-free a-priori salience for the signal and sends it alone with the signal to the processing module.

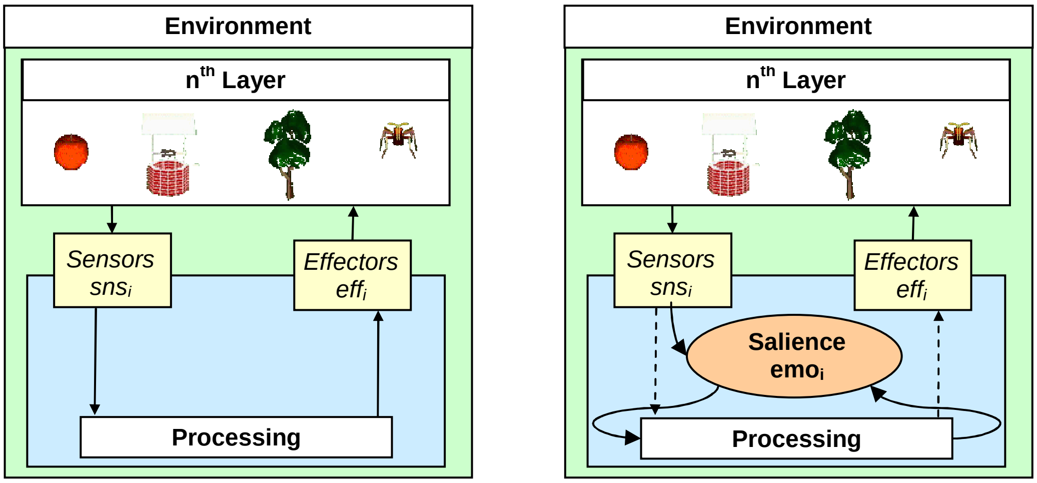

Figure 3 - Russell and Norvig Architecture (left), Martinho's Modification with Emotivector (right)

Isla and Blumberg [14] define salience as the "degree to which an observation violates expectation" $s(x) = (1 - c(x)) / c(x)$. As noted by Martinho there seems to be no need for context to estimate this a-priori salience. Salience could be performed using only the changes in percept values over time. The salience module is context-free and leaves to the processing module the responsibility of putting the salience in the context of the agent and its environment. Of course, the processing module can use this recommendation or ignore it, according to its processing resource policies. So we can conclude that to detect when the mismatch between our expectation and the percept value is significant and, when it does, tag the percept as salient, we don't need context or interpretation. When the salience information reaches the processing module, sensations are appraised in the context of the agent and its environment, and emotions may be generated and expressed accordingly by our agent. Please note that the evaluation is a context free it just measures the mismatch between expectation and value, in relation to a desired value. The code of the information flowing through a sensor is usually consistent, in the sense that it is the repeated measurement of a specific aspect of the environment on a same scale over time. We define our universe of perceptions as an n-dimensional vector space where n one-dimensional vectors (or a n-dimensional vector) define a perception in time. Each one-dimensional vector is thus the perception of a specific aspect of the environment at a certain moment in time. Note that we do not associate any a-priori semantics with the one-dimensional vector. Additionally, to ensure that our mechanism can be used in a variety of situations, every aspect of the world is reduced to a value in the normalized range <0, 1>. The normalization function may be customized according to the characteristics of each dimension of perception. We would like to model the fact that a same difference between two measurements is more relevant near the agent than far away from it. Depending on the situation we can use specific modulation function to stress out changes in particular interval of values like in the example where closer changes are more relevant that changes far from agent.

The definition of emotivector provided in Martinho's work is as follows. *Emotivector is a one-dimensional vector with a memory and mechanisms using this memory using an anticipatory affective model to assert the salience of a new value. The anticipatory affective model generates an affective* *signal from the mismatch between sensed and predicted values, providing some qualitative information regarding the salience of the new value.* The emotivector is used to generate the low-level context-free attention and also emotion.

Architecture and computational details was described and presented in Martinho's work and I have presented my use and modification in [K5]. The general principle of the emotivector is the following. Using the signal history of a sensor, the emotivector computes the next expected signal value of the sensor. Then, by comparing the expectation with the actual sensor value the emotivector is evaluated for attention potential. Afterwards, a sensation is generated. The combination of both attentional and emotional salience is then fed to the processing module to be used to support resource management. The Martinho's model of attention presented in his thesis is inspired by Posner's exogenous and endogenous systems [15] and Müller's and Rabbit hypothesis [16]. This inspiration is reflected in the two components that are used to compute the emotivector salience. The exogenous component, inspired in bottom-up, automatic reflex control of attention, and emphasizing unexpected values of a signal. The endogenous system, inspired in top-down, voluntary control of attention, and emphasizing the closeness of a signal value to actively searched values.

2. AIMS OF THE DOCTORAL THESIS

2.1. Problem Statement

The problem that my work is focused on is the **anticipatory behaviour**. Anticipation is often seen as another word for prediction especially in the *Artificial Life* (ALife) area. I claim it to be much more than that. It gives another **dimension** to the decision process – the information about the future. It is also a very elegant way to generate emotions. I find it difficult to categorize all the types of anticipation I met with using the current categories. Also the topic of **voluntary control** of anticipatory behaviour is in my opinion not well mapped. I identified all this as a problem to be solved in my work by a design of a complex but scalable architecture. Some questions that this work addresses are mentioned below:

- What is anticipation?
- What is anticipatory behaviour?
- What are the types of anticipation?
- Various definitions and categories of anticipation are given, but what is the difference?
- What is the difference and similarities of reactive and anticipatory approach?
- Can anticipation be of any help or improvement in the existing systems and how?
- What is the difference between anticipation and prediction?
- Does anticipation necessarily need learning?
- How is anticipation linked to emotions?
- Is anticipation a single mechanism in the artificial creature architecture?

2.2. Goals of the Thesis

The main goals of this thesis are stated below. There are four higher level goals, where some of them are broken down to sub-goals:

1. Survey **state of the art** in the field of anticipation, anticipatory behaviour and the associated fields, including technical (agent architecture, action selection mechanism, artificial intelligence and artificial life approaches) and non-technical (emotion, expectation, behaviour), closely related to the researched topic.

2. Suggest **own** view on **anticipation** so that
   a. the idea of anticipation playing role in more aspects of behaviour control is visible,
   b. the different anticipation approaches can be satisfactorily identified,
   c. anticipation improves the behaviour in certain criteria,
   d. levels of anticipation are explained in detail.

3. Based on the described theory **design and implement own** architecture so that
    a. each identified layer is built and tested separately,
    b. the layers are chained and the whole architecture tested,
    c. learning should be in-time and unsupervised,
    d. the simulated environment is open,
    e. it enables the generation of emotions as a result of using anticipation,
    f. implementation of anticipatory behaviour brings value in measurable terms.

4. Analyze the achieved **results** and evaluate
    a. the usability of the suggested approach,
    b. the quality of achieved results,
    c. the complexity of simulations and the effect of growing complexity on the approach,
    d. the comparison with other approaches.

3. MY APPROACH TO SOLVING THE PROBLEM

This chapter presents **the core of my thesis** and is devoted to details of **my contribution** (working methods) to the field of anticipation in artificial life domain. One of my main contributions as I see it is to propose a single architecture called **8-factor anticipation**. This term is my original term that I introduced in this thesis the design described in this chapter is my original work. Each "factor" of my architecture is described in this chapter.

What is not that obvious and sometimes even missed, anticipation is not matter of one mechanism in a living organism. Anticipation happens on many different levels in one creature. The works studying anticipation seems to overlook this fact so far, focusing on the anticipatory principles, mechanisms and their optimization. There was undeniably great progress in past years in theory and applications of the anticipation. What I miss in the deployment of anticipation in Artificial Life domain is to follow the nature's example and use anticipation principles in more design blocks. Several researchers categorized the anticipation already. Even though I embrace the categorization of anticipation Martin Butz did I was not fully satisfied with it. I missed there connection between the types and the consciousness. In addition to the existing types I added the consciousness and thus created 8 types of anticipation. This idea is the basis of my theoretical contribution to the field. My thinking here is that each algorithm is better in a different way and by combining them and properly selecting the right one I can improve the results.

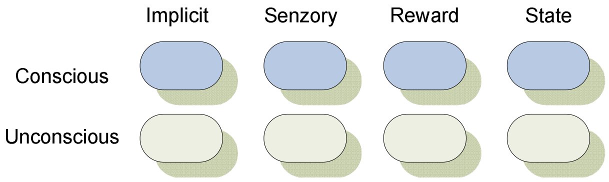

Figure 4 – The 8-Factor Anticipation

All the types are schematically shown on the Figure 4. We can say that the complexity grows in the picture from left to right and from bottom to top.

3.1. Unconscious Implicit Anticipation

Unconscious implicit anticipation (UIA) concludes the behaviour that was imprinted in the creature by nature or creator (in our case) and that is not voluntary. Under this we can understand the very basic reactions with anticipation imprinted in the design. Reactions and reactive behaviours itself is not anticipatory and in fact is very often used and understood as exact opposite of anticipation. So what exactly are reactions with anticipation? We cannot say that the reaction is associated with prediction of next sensed value, state or reward because these are subject of the other anticipation types. There classical view of implicit anticipation would be satisfied with the fact that it is the prerequisites given to the system wither by the long evaluation or by the creative mind of architecture designer.

In order to describe it we need to define a formalism to approach this is systematic manner. At this very basic level we have only the set of inputs I and set of possible outputs O. By this we implicitly assume discrete values which we typically have in a virtual environment. Please also note that we are not speaking about agent sensory inputs or actions yet. The reason is that I'm trying to generalize this description so it can be used for agents' internal blocks and not only for agent as whole. The reaction base is typically in form of projection $I \Rightarrow O$. The inference mechanism is very simple: if any of the input matches the output is executed. There couple of possibilities from binary rulebase to ACS. On the contrary the anticipatory approach would be to expect another input after the executed action $I \Rightarrow O \; x \; I$. It still might seem as nothing new one can say that everything was already presented. We must realize here that on unconscious implicit anticipation there is no mechanism to modify this rulebase other than evolution. As was said above it reflects only the non-learned behaviour. The interesting question is if the same rule as here can be then created in some other probably conscious level. The answer is yes the same rule can be inferred by the consciousness of the artificial creature but its execution takes longer path so it is less likely to be executed. We do not need to solve creation of new or forgetting of obsolete rules here because the rulebase is fixed and it is subject of only minor evolutionary changes.

3.2. Conscious Implicit Anticipation

The combination of conscious implicit anticipation (CIA) may seem illogical because as we said above implicit anticipation is something imprinted in the creature by design. How can this be consciously controlled is the right question and the moment. Here still everything depends on the design but the results are available to the higher levels and also higher levels data such as desired state (converted to the desired value in the current step) are available as inputs. This means that here we can chain the existing actions together in order to create a new non-atomic action, which would have no decision time in between and focus the attention. I will continue here with the formalism I started in the previous chapter. We still have only the set of inputs I and set of possible outputs O. In previous chapter we ended with anticipatory projection from input to output and expected new input $I \Rightarrow O \; x \; I$. We explore this further here with two modifications described below.

My first suggestion to this is to add to the expectation also expected next action. This is expected to improve the reaction time. In our formalism, we are now projecting the current input and the output to current output, expected output and expected input $I \; x \, O \Rightarrow O^2 \; x \; I$. Please note that in the agent terminology I moved from the term output to action.

Imagine the predator evasion scenario and imagine two prey agents. One of them equipped with standard prediction scheme ($I \Rightarrow O \; x \; I$) and second with the suggested modified one ($I \; x \, O \Rightarrow O^2 \; x \; I$). Both prey-agents are in the vicinity of predator which will through the sensors result in (input I) the action of both is to flee (output O). Even if the reaction process is fast it still takes some time to search through the rule base for a match. My question is what will happen in case agent will have the chance to take another action before it recalls the appropriate action from the rulebase? Since there will be no action selected yet it must wait till it the next step. On the contrary if this moment comes to my modified agent it can straight away execute the "prepared action". This is graphically demonstrated on the Figure 5.

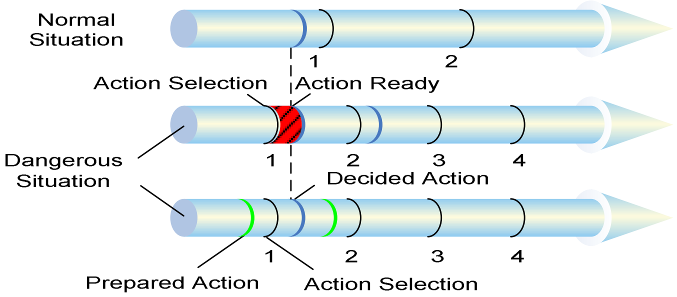

Figure 5 – Action Selection without the Action Anticipation (top) and with Action Anticipation (bottom)

At this point because it is conscious part we can introduce my second suggestion. We do not have the reward yet but we can have a rate of change for the input value. For output values, because they are typically in ALife a discreet values not expressed by numbers, statistical measure such as probability or likelihood can be measured. This is another parameter that can add value to the decision process and help to choose the right action in the correct moment. This describes the typical scenario but in fact any combination in term of discrete and continuous in the input or output can occur. So we are adding two new values the r_i and r_o which I will call rateability (the combination of word *rate* and *probability*). This enriches the projection $I \ x \ O \Rightarrow O^2 \ x \ I \ x \ \Re^2$. For example we have a proximity sensor for exploring creature that provides one input called *distance*.

Let's have a rule to change direction when the distance is lower than half a meter to avoid collision. We have two actions available "move" and "turn in one direction by a given angle". Let's also assume that the previous action was to move straight. This simple example shows that even on a very basic level the amount of information available can vary.

1. The classical reactive approach
   IF x THEN y where $x \in I, y \in O$
   Example: IF *distance* < 0.5 THEN turn(90)

2. The classical anticipatory approach
   IF x THEN y EXPECT z where $x, z \in I, y \in O$
   Example: IF *distance* < 0.5 THEN turn(90) EXPECT distance > 0.5

3. First suggested improvement – action anticipation
   IF x AND PREVIOUS_ACTION a
   THEN y EXPECT z AND EXPECT_ACTION b where $x, z \in I, y, a, b \in O$
   Example: IF *distance* < 0.5 AND PREVIOUS_ACTION move
   THEN turn(90) EXPECT distance > 0.5 AND EXPECT_ACTION move

4. Second suggested improvement – rateability evaluation
   IF x AND PREVIOUS_ACTION a
   THEN y EXPECT z AND EXPECT_ACTION b
   WITH <*r_i*,*r_o*> where $x, z \in I, y, a, b \in O, r_i, r_o \in \Re$
   Example: IF *distance* < 0.5 AND PREVIOUS_ACTION move
   THEN turn(90) EXPECT distance > 0.5 AND EXPECT_ACTION move
   WITH *<0.1,0.6>*

The approach that I find fit for this purpose is emotivector described in 1.3 and only the model of attention since the second part the model of emotion needs the information about reward too. This determines this level for attention selection. This model does not have the *rateability* factor, but this can be added to the emotivector theory. The first difference is that emotivector does not include the output (action) value estimation and evaluation. Since actions are usually not expresses in the real numbers but as worded abstraction, it would be very complicated to normalize them and calculate differences. It is not even required, the only thing that is required is to have an expected output stored (in other words) the prepared action. As mentioned this estimation will be based on previous action therefore $\hat{a}_t = a_t$. The other difference is the *rateability* evaluation. For input value it is simple, the speed of change (velocity) for discrete values is calculated as a difference $v = \Delta x / \Delta t$ in one step we calculate the velocity $v = \Delta x = x_{t-1} - x_t = r_i$. However I will argue that there is a better measure of how the object is interesting and that is the salience computed by the emotivector so we will use it instead of simple change speed $r_i = salience_t$. For output rateability will be counted as frequency of occurrence of the output across the whole actions $r_0 = n_0/N$.

The model of attention is implemented as follows. Using its history at time $t-1$, the emotivector estimates a value for next time $t(\hat{x}_t)$ and predicts that its value will change by $\Delta\hat{x}_t = \hat{x}_t - x_{t-1}$. At time t, a new value is sensed (x_t), and a variation $\Delta x_t = x_t - x_{t-1}$ is actually verified. The newly sensed value triggers the computation of the emotivector components. The exogenous component at time t (EXO_t), is based on the estimation error and reflects the principle that the least expected is more likely to attract the attention. EXO_t is computed as follows $EXO_t = |x_t - \hat{x}_t|$.

If the emotivector has no associated desired value, the exogenous component will be the only factor contributing for the emotivector salience. However, if there is a desired value (d_t), then the endogenous component of the emotivector is also triggered by the newly sensed value. Whenever a desired value is present within the emotivector at time *t*, the endogenous component *(END_t),* is computed. It is a function of the distance of the sensed value to the desired value (Δs_t) and of the estimated distance of the expected value to the desired value $(\Delta \hat{s}_t)$. END_t is computed as follows $END_t = \Delta \hat{s}_t - \Delta s_t$ where $\Delta s_t = |x_t - d_t|$ and $\Delta \hat{s}_t = |\hat{x}_t - d_t|$. Together, the exogenous and endogenous components define an a-priory salience for the emotivector. This salience can computed by adding the absolute value of both components as $salience_t = EXO_t + |END_t|$. Of course, other emotivectors are being evaluated at the same time, each one with its own salience computed based on the described process.

3.3. Unconscious Sensory Anticipation

Moving on to the sensory anticipation on the unconscious level (USeA) concludes all the sensory input gathering, pre-processing and data filtering. Basically here we can meet all the functions that cannot be voluntarily influenced. In broader sense by this we can simulate the situation where the input magnitude is so huge that it cannot be processed all by the conscious processes. This information is collected, processed, stored or disregarded based on the attention and other factors. In my work I will not go that far to implement this in full scope and I will stay just with the input gathering and pre-processing. In anticipation we talk all the time about some estimated future value but less we speak about the means how to get this value. In most implementation very basic approaches such as the no-change rule are used. However statistics provides a wide variety of very powerful and complex methods to estimate the future values based on arbitrary long history data. In my opinion these have place exactly in this part. For anticipation purposes they are just tools that present us with the estimated value that we can in other levels use and further process.

The second function that we sometimes require to get closer to the animal world is filtering the information in order to reduce the information value so more less informative data can be processed at the same time. Let me demonstrate on an example what I mean by this. We can take again our robot. The robot has some hardware limitations in terms of data size it can process. So it is in our best interest to give it more accurate data about the objects it has focus on (we already know where the focus is from the previous level) and other objects data can be reduced to some approximation. Let still continue with the distance measure but this time the distance is measured in four directions. In the direction we are close to the wall we would be interested in the number, how close are we. In other directions the information if the distance is short, medium or long would suffice. Of cause one can object that in narrow spaces we would need detailed information in all the directions in worst case scenario. Yes that is correct and that is why it can take then more than one time unit to process the data and the robot would slow down.

Here we meet first possible conflict of the levels if we want to combine them together. For the focus attention mentioned in the previous chapter we would need the exact data to measure the changes and decide about the object we want to focus on but here I hid this data in some intervals. We have two resolutions to this situation, leave it as it is and accept the fact that objects can be in the focus only if the cross borders of the intervals and if they are close. This variant does not seem to bring value in certain sense it would impede the previous two factors significantly. Even if it is more probable to have closer object in focus than the further one we still would want our agent to be able to focus even to object in long distance if they are interesting enough so we need another solution. The second solution is to bypass this filtering for attention focus. That would solve our problem with focus, but neglect the reason why we used the filtering so this does not seem to be optimal as well. The solution is at hand, the sensors alone have the data and these are being pre-processed by the unconscious layers, so these can have access to the full information and present only results to fit with the above scheme of a limited data size that can be processed at one time.

For the implementation I continued using Martinho's emotivector and the suggested simple predictor. In this simple predictor the prediction of the next value is the weighted sum of two parameters, the previous prediction $\hat{x}_{t-1}$ and the sensed value x_t. Both compete for influence in the computation of the new prediction $\hat{x}_t$. In a certain way, the weight w_t accounts for the certainty of the system in its previous prediction. When there is no desired value, the exogenous component EXO_t is used for the value of $w_t = |x_t - \hat{x}_{t-1}| = EXO_t$ and the prediction is then $\hat{x}_t = \hat{x}_{t-1}(1 - w_t) + x_t \cdot w_t$. When there is a desired value in the emotivector the learning rate is set to the intensity of the current sensation associated with the END_t. As such, the change of w_t (Δw_t) at each step is computed as $\Delta w_t = \xi_t \cdot (x_t - \hat{x}_{t-1}) \cdot w_t$ where $\xi_t = |END_t|$.

3.4. Conscious Sensory Anticipation

As in every chapter I will discuss first what I understand under this category and support it by examples. We are on the *conscious sensory anticipation* (CSeA) level at the moment, so we have access to the sensor data and from the previous level even to the estimate of future data.

What else would we need at the sensory level? There are still many things that would be very helpful for the artificial creature to derive that would require some higher level processing than just having several expected values for each sensor. A dog hunting a rabbit does not need to sense the hare continuously. If the hare, for example, disappears behind a bush the dog predicts the future location of the hare by anticipating where it is going to turn up next and continues its hunt in this direction. This behaviour described below needs little bit more than a pure sensory input. It requires recognition of the objects (rabbit, bush) and making projections of a sensory data that cannot be directly measured at the moment by sensors. Also some knowledge about the rabbit and the environment has come into play. This means that we would need some already stored data to be recalled from the memory and associated with the recognized objects. As we can see, the situation is complicated and requires mechanisms that I did not described so far. It is obvious that model of some kind would be very helpful at this level. The solution to this is that we are on conscious part of the architecture and the consciousness has access to the memory, plans, active knowledge etc... This knowledge is shared across all my anticipatory levels. This means that I need to introduce a shared part for the conscious levels that they can either utilize or contribute to. It is schematically shown on Figure 6. For the lack of better expression I will call it **memory**. This is only a logical design in the physical design of the whole architecture several components will use their own memory that is not shared or use this shared memory. With the use of memory the sensory anticipation can be used now in our example to predict even more complex events that just sensory input. It can then abstract objects and predicts future sensory input for these objects. We are still on the sensory anticipatory level so we cannot derive yet any other observations than future sensory inputs. Filling the memory, building models and beliefs, planning and other cognitive tasks, will be subject of further levels.

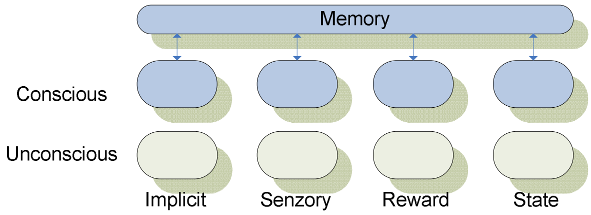

Figure 6 – Shared Media for Conscious Factors {original drawing}

Now I will get more in details about how to implement this. One of the suitable approaches was suggested by Isla and Blumberg in their work [14] the Probabilistic Occupancy Map (POM). This is simple yet efficient approach to track objects even when they are not visible (hidden behind another objects) and estimate the probable position. It is based on separating the environment to hexagons (also called nodes) and assigning each of them for each object the probability of its presence in it. This probability is diffused using simple isotropic diffusion

$$p^{t+1}(n) = (1-\lambda)p^t(n) + \frac{\lambda}{8} \sum_{n' \in neighbors(n)} p^t(n') \qquad 1$$

where λ is a diffusion constant in the range [0,1] and $p^t(n)$ is the probability of the node *n* at time *t* to reflect the last motion pattern of the observed object. The diffusion constant can be modified

$$\lambda_i = \lambda_c + \max\left(0, \frac{v \cdot l_i}{l_i^2}\right) \qquad \mathbf{2}$$

where v is the velocity vector, l_i is the position offset between the current node and the node's *i-th* neighbour, and λ_i is the diffusion rate along the *i-th* connection. λ_c is a constant diffusion rate, ensuring that some probability is diffused to every neighbour, even if that neighbour does not lie in the direction of the velocity vector. One of the main contributions of this work is the fact that the levels can support each other. In the USeA I implemented for the sensors several estimators of the future values. These can be used instead of using the method of altering the diffusion constant to aid the algorithm. The formula for diffusion constant helps only to propagate the probability in the map in the right direction and to decrease it for more distant nodes. But we can use estimators to aid this process and provide estimated position based on the history of measured values.

3.5. Unconscious Reward Anticipation

We are now finally approaching the area that almost all current anticipation behaviour designs operates with (sometimes with combination with lover levels in the sense of my description) the *unconscious reward anticipation* (URA). The reason is at hand, the reward or better said reinforcement to include also punishment is a powerful way to learning. Reinforcements together with the expectations (anticipation) also serve to generate emotions. My contribution to this area is to argue about the categorization from point of consciousness to fit this into my framework and to select the appropriate approach to implement it. As in every level of design I will also introduce my own improvements to the design.

The generation of emotion is in general achieved through comparison of the expected reinforcement with the received reinforcement. This is a basic principle used in most of the current works. The absolute value of the difference can drive the strength of the emotion. Multiple emotions such as happiness, surprise, disappointment, frustration, sadness etc... can be generated. This means that without the algorithm to verify the expectation against the real reinforcement, also the set of emotions needs to be defined or in more general case the set of rules for emotions and their generation. My first **contribution** to this level is in definition of three main emotion elements to consider the reinforcer (generated by received reward or punishment), expectation difference (generated by comparing the expected and received reward) and the surprise (evaluates the expected versus real value).We can normalize each of these to an interval *[0, 1]* and then draw an **emotion cube** with these values on each of the axis Figure 7.

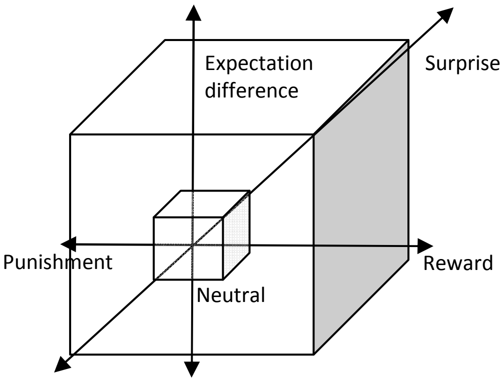

Figure 7 – The Emotion Cube. ***Reward/punishment on x axis, expectation difference on y axis and surprise on the z axis***

Adding names of all the emotions to the cube would make it difficult to read. The Table 1 below gives the mapping of the emotions and contains my own definition of the combination of the emotions. The cells of the table maps to the cube as the intensity of each emotion grow. The intensity of the final emotional state is created by superposition of the intensity of the three parts of emotion. Their intensity is given for reward and punishment by the amount of the reward/punishment received and for the surprise by the distance of the expected value from the observed value. As I mentioned already above, a complex creature pursues more goals at once and hence the creature also has multiple reinforcement (reward/punishment) expectations. The final emotional state is generated by a superposition of all the current emotions.

The described approach to emotion uses very simple mathematical approach. The only complexity is the superposition and evaluation of the emotion to be generated from the emotional state. There is a value x observed by the agent's sensors and its expected value $\hat{x}$. This value can be connected with reinforcement. I define r as the reinforcement that was actually received and $\hat{r}$ the as the reinforcement expected. The difference between expected and received reinforcement will be denoted as Δr. The surprise factor is Δs.

$$\Delta r = \frac{\hat{r} - r}{\max(|r|,|\hat{r}|)} \qquad \Delta s = \frac{\hat{x} - x}{\max(|x|,|\hat{x}|)} \tag{3}$$

	More reward received	Received as expected	More punishment received
Reward expected	Joy + Surprise (Pride)	Joy	Sadness + Surprise (Suffering)
Negligible	Joy + Surprise (Happiness)	Neutral	Sadness + Surprise (Disappointment)
Punishment expected	Joy + Surprise (Relief)	Sadness	Sadness + Surprise (Anger)

Table 1 – My Designed Emotion Mapping on the Nine Sensation Model

All these values need to be normalized in order to be comparable and projectable to the cube. I use both the actual observed value and the associated reward in order to be able to capture situations where there is small increase in reward/punishment, but still a significant difference between expected value and the actual value leading to higher value of surprise.

The evaluation of final emotional state is done as mentioned above by superposition of the partial emotions. The resulting vector depicts the sum of all the emotions and its position in the cube then dictates the final emotional state. There is another aspect to emotion that I want to capture in my work as well. Usually in simulations the emotion is in effect until it is changed by another emotion. In a simulated world that has many agents and objects this is a good approximation. However what if there is no reinforcement for a longer period of time, the emotion will not definitely stay the whole time with the same intensity (i.e. the emotion intensity decreases over time). This is another addition I made to the emotion approach implemented. Each emotion that has occurred is counted in, but it loses intensity with each time step. This way even if there is a bigger reward received followed by minor punishment, the emotional state will still favour the outweighing contribution of the reward, but if the punishment comes few steps later, where the joy of the reward should wear off, if can influence the emotional state.

3.6. Conscious Reward Anticipation

On the contrary to the previous chapter, on the conscious level there is the advantage of the consciousness shared media so the possibilities are much wider. At this level we are finally reaching full capabilities of the current architectures, plus my design has two additional levels above this one thus leaving space for further advancements.

At this stage we are looking for framework that works with reward, and is able of working with the observations and gained knowledge including creating, modifying and deletion. We need to keep in mind that these might serve for other conscious levels to work with and hence they need to be compatible or abstract enough so all levels can understand them. This also means that the system should be open enough in terms of inputs it requires and outputs it provides co I can easily integrate it in the complex architecture. In my opinion ACS is ideal algorithm for the conscious reward anticipation due to its generalization and specialization properties. In my implementation I selected one of the basic ones that gave the idea to others i.e. the work of Stolzmann [11].

3.7. Unconscious State Anticipation

This is the last of unconscious levels at the same time the most sophisticated one and most complex one. This level has a similar problem to the conscious implicit anticipation. The combination itself seems at the first sight confusing. However it is important part of the architecture and has its meaning. All the state creations manipulations, and estimation of next states that are not brought to consciousness right away or at all have place here. One example is the internal state of the creature. It is monitored through internal sensors, it is regulated and working without external actions needed but some unusual states should be reported to the conscious levels. The motivation was taken from the nature as always. As long as all the internal variables are within the certain boundaries there is no need to alert consciousness Once some of them drops or exceeds the threshold and external action is required to get it back within the safe range (energy is for example low – i.e. hunger). The consciousness controlled action is needed as the situation needs to be evaluated and proper actions executed in order to address the situation. So the artificial creature internal state can be monitored and partially controlled from this level. Second consideration is if models of other agents or environment can be created on the unconscious level. I can think of one possibility – latent learning. It is surely subject for discussion if the latent learning is triggered consciously – learning something “just in case” we will need it sometime. But I would here say that it is not. My argument will be based on the available storage space and processing speed. It seems very unlikely to create models and states of the whole environments and store them for a long term and then for every new goal going through them if they can be used or not. More likely the models are created subconsciously in the short term memory and when the proper reinforcer appears, and the model is proven useful then it is kept in the conscious long term memory. This scenario seems to be more efficient and reasonable from the resource optimization point of view unfortunately I’m not aware of a nature experiment to support my theory. Nevertheless I’m going to follow that theory in my work.

What exactly is anticipatory about keeping the internal state and latent learning? I will start from latent learning because the answer is straightforward, creation of knowledge and maps about the situations we are going through and storing them anticipates that they might be useful in the future. So latent learning is anticipatory in its very nature. With the internal state it is not that easy as the principle seems more or less reactive (value drops to certain level – alert is triggered). But the rate internal values decrease is not same under different circumstance (more energy is consumed when running than when exploring etc…). So anticipatory monitors using the information about current actions and observed internal values behaviour would help to optimize the system and bring it again from reactive to anticipatory.

For the implementation of this level, the ACS framework described already in previous section can be used. Instead of creating an environmental map in its explicit representation (even if that is also possible). I’ve decided to capture the latent knowledge in terms of the ACS rule base. The advantage is, that my architecture has support for ACS already build in and as such can work with it. Another such advantage is that degradation over time of such knowledge representation is then trivial and means removal of random classifier from the rule base. The disadvantage is that the next level than would not be able to use for example planning directly on such knowledge representation. I have decided to stay with the ACS latent knowledge representation in my work for this level.

3.8. Conscious State Anticipation

The most complex and therefore the most interesting factor of my architecture is conscious state anticipation. Basically all the classical AI approaches can find their place here starting from state space search through different methods of planning (for example based on Markov decision chains) up to the reasoning about others and self. These tasks typically require more time to process. This can be imagined as a state of the agent where there is no urgent internal need (food, sleep, etc..) and also the agent external goals are satisfied. In that case the agent can select the action to be to build, review, updated or evaluate the model of its own state or of others. This type of meta reasoning case still have anticipatory basis. In my work I focused on the interconnection of several levels together connected by memory as described above.

Working on the previous two levels where I used the ACS algorithm I noticed and pointed out several weak spots of the approach. Thanks to the probabilistic approach and low degree of specialization of the newly created classifiers the behaviour even after long learning cycle is still random. While this greatly promotes the environment exploration it lacks the deterministic use of the gained knowledge. In my design the agent creates parallel to the rulebase of the ACS a map of the environment in the memory. This map is however tight to the ACS very closely as is it created by the ACS exploration phase and is composed of the applicable actions successfully executed. In a dynamic environment the applicable actions can change, thus this map needs to be able to adapt to these changes. The second modification is a priority queue of the goals. This queue has multiple levels and priorities as we have a planning (deliberative approach), exploration (reactive approach) and internal state (hysteretic approach) competing for actions. Please note that this is a simplified situation. In my architecture there will be up to 8 competing layers of the priority queue.

The last modification is the actual planning approach. The planning is done on the map of the environment created through the discovery of the environment. This means that the agent is not able to plan action that has not yet been applied as there will be no knowledge in the agent map and rule base about such action. This fact helps to balance the exploration and the planning phase. The state space is then created by a position and the applicable actions. Such state space can be searched for goal state by different algorithms from depth or breadth state space search, through *A** up to approaches based on Markov decision chains such as dynamic programming. I chose the *A** algorithm.

4. RESULTS

For ALife in most cases simulations in the virtual environment are the method how to test theories and compare effectiveness of results with others. I have conducted my experiments in the REPAST simulation tool [17]. The presented simulation scenarios are intentionally made as simple as possible to clearly demonstrate and articulate the functionality.

4.1. Unconscious Implicit Anticipation

The setup of the experiment is placing a robot (agent) into a world with several objects. There are walls obstructing way and the beverage which the agent has to reach. The goal for the agent architecture is to reach the goal in effective manner. There are three layouts of the environment, one without obstacles, and two containing different types of obstacles. In each simulation there are always two agents, each trying to reach different goal location. This experiment was designed to show in practical sense the meaning and differences in the implicit anticipation and I understand it. The setup of this experiment counts two instances of an agent with similar sensors, effectors and action selection mechanism. Each agent has a “proximity sensor” of the 8-neighbourhood. It gives the agent information about presence of objects. In order to ensure agents follow goals, one agent is thirsty and goes to water and second agent is hungry and searches for food. Both agents for the sake of simplicity know the location of the object they are searching for. The task is to get there. To complete the task the agents have a set of nine possible actions and those is movement in all possible 8 directions and “do nothing” action. In each scenario the agent is given a set of rules that maps the inputs to the outputs. In the first experiment the agents moves randomly, the rule base has just one rule that tells the agent to stop when the food is found. There is little anticipation in this scenario. In the second experiment reactive behaviour was implemented with the implicit anticipation of “right angles” which means that when meeting the wall rotation by 90 degrees will help to avoid it. The rulebase here has the rule from previous agent plus 8 other rules that gives appropriate action to the met walls. In the third experiment, I enriched the reaction base with different mechanism for navigating in space with obstacles which is called “wall following”. This means that we have even larger reaction base as more situations of wall presence in the base is needed to successfully navigate along the wall.

All these agents were tested in three different scenarios where first was without obstacles, second contained only straight obstacles and the third contained also curved obstacles. In the first scenario it takes long until the agent randomly stumbles on the food regardless the obstacles (please note here that some single cell organisms use this method of navigation). In the second experiment the 3rd scenario is not achievable as the rules do not allow the agent to cope with the obstacle. In the last experiment we can see that it is quantitatively better than the previous one, not only it can reach the goal in fewer steps, it also can complete the third scenario. The Table 2 shows the number of steps necessary to complete the scenario.

Experiment	Scenario 1	Scenario 2	Scenario 3
(a)	Rand	Rand	Rand
(b)	26	38	∞
(c)	26	35	38

Table 2 – Results of the Experiment with Unconscious Implicit Anticipation

4.2. Conscious Implicit Anticipation

The aim is to test and prove the emotivector attention focus features. For this purpose I designed scenario including several types of agents. The predator agent shown as "*wolf*" is observing the environment and its task is to pick a target of interest based on their salience, there are three agents to be observed. Two "*piglets*" agents both with similar characteristics, except the move pattern, while one of them uses a random move method to navigate through the environment, the second one moves in a constant cyclic pattern. The last agent depicted as "*flower*" is a static agent. It was confirmed by this experiment that the moving agents are more interesting for the observing agent than the static ones, which was expected based on the fact that emotivector is sensitive to the observed value change in time. This reveals the strong and weak sides. For the attention focus only the changes in the environment are relevant on this level. This is acceptable on the basic "reactive" level.

One of my suggested improvements to the emotivector approach was introduction of the **rateability**. In the second experiment I demonstrated that it can be beneficial and lead to improvement. I used the same setup as in the first experiment but I added one more agent this time with the enhanced emotivector. In order to maintain the same conditions I kept both agents on the same position. For visualization purposes I show one of the agents above the other and depicted the additional agent as a "*bear*". In this experiment I **compared** the performance of emotivector **improved by my own design** below referred to as "enhanced emotivector" against the Martinho's original version of emotivector below referred to as "standard emotivector".

Due to the random movement of one of the agents in this example I evaluated the results statistically. I simulated the change in attention over 1000 steps of the simulation, and repeated the same simulation 3 times.

Experiment	Enhanced Emotivector	Standard Emotivector	Stability Increase
1	12.7%	14.9%	2.2%
2	11.6%	13.3%	1.7%
3	11.0%	14.3%	3.3%

Table 3 – Enhanced Emotivector Test

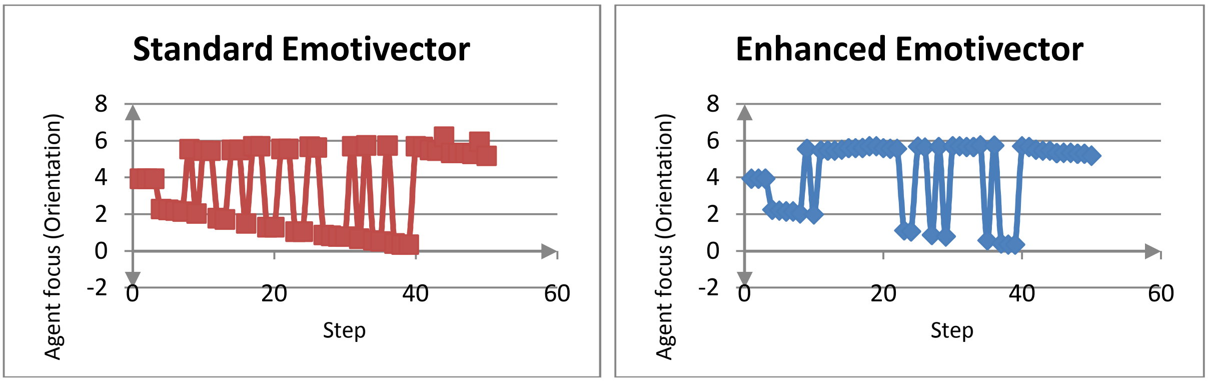

Figure 8 – Standard (left) and Enhanced (right) Emotivector Attention

The results are summarized in the Table 3 above and also a fifty steps sample is shown on the two figures below. In each of the simulations my enhanced emotivector exhibited better **stability of**

attention (in the experiments conducted in average by 2.2%) while still being able to change the attention focus if the other moving object is more interesting.

The Figure 8 show the difference in both approaches, while my approach exhibits stable attention focus areas (shown above), the standard emotivector approach shows unnecessary oscillations between the objects of attention.

4.3. Unconscious Sensory Anticipation

The main experiment here is to test the estimators and evaluate their qualities. In the referenced work [6] there are some conclusions about them, but none of these is shown or proven. The predictors tested are referenced by the abbreviations *Simple Predictor* (SP) uses the equations described in 3.3, *Limited Simple Predictor* (LSP) uses the same equation, but also keeps history of the input values calculates the mean and the deviation and limits the prediction if outside the statistical range and lastly *Desired Limited Simple Predictor* (DLSP) uses also the desired value. I decided for this experiment to set the desired value to 0.6.

Both the predicted value and also the predictor error of all 3 estimators are shown on the following Figure 9. It is clearly visible that the LSP and the DLSP are nearly identical.

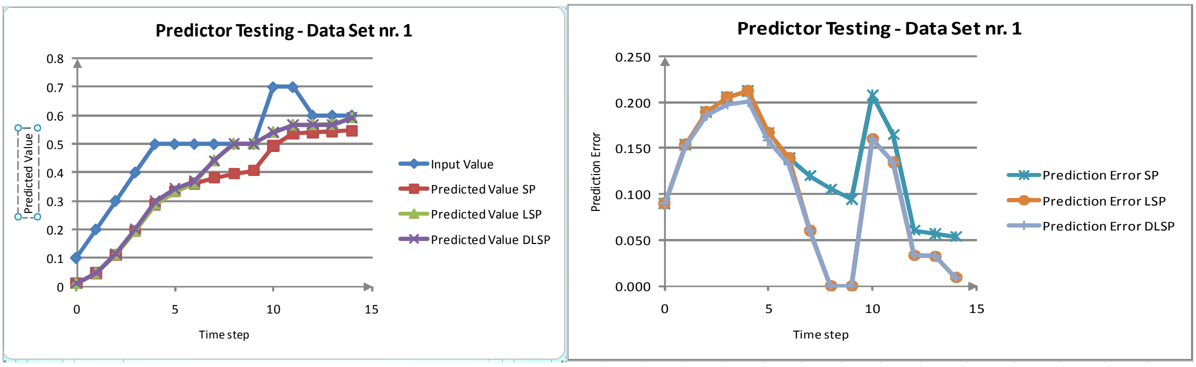

Figure 9 – Results of the Predictor Testing – *Predicted Value (left) and Prediction Error (right) on Data Set nr. 1*

There are several conclusions that can be drawn from the results above. The *Simple Predictor* has poor results. It is not shown in the table above, but the convergence speed was very slow. In this experiment the convergence to value 0.1 ± 0.01 was 87 steps. The other two predictors showed comparably better performance. As can be seen above they are both able to converge in 5 steps. In situation of oscillating values they are not able to adapt and they oscillate too.

4.4. Conscious Sensory Anticipation

This level has only one experiment and that is the object persistence scenario. This enables the agent to be able to estimate the position of another agent which is hidden behind obstacle and thus cannot be directly perceived by the sensors. The experiment compares the original work with my suggested modifications. For this experiment I used similar setup to the one used in 4.2. I still kept a stationary observing agent. This is purely not to bring another variable into the experiment. The algorithm works even if the observing agent is moving. The observer shown again as "*wolf*" is trying to follow up the movement of a moving agent shown as "*piglet*" similarly to the previous experiment. This scenario contains also a wall which hides the observed agent (*piglet*) and thus renders is hidden for the observing agent. When the moving agent disappears the probabilistic occupancy map is initialized and probabilities are diffused each step to estimate the position based on the known last position speed vector. In one experiment the probabilities are diffused using the modification of the diffusion constant. In another my suggested modification with also the position estimation was used.

The original approach had problems to keep propagating the probability in the right direction and in some experiments tend to follow up in the last observed direction and speed but after few steps the probabilities were so dispersed that the estimated position stopped being propagated in this direction. In my approach the estimator was used to estimate the next position based on the same values but also the observed history values. However this position has only a certain degree of reliability, for this I used the probabilistic occupancy map to reflect the fact that other positions next to this have also some probabilities of occurrence of the observed agent because the agent could have changed the direction of the movement or even stopped.

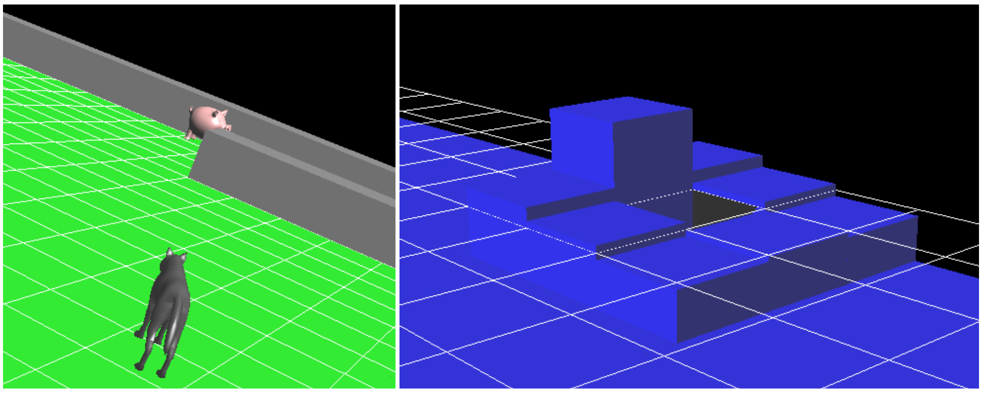

Figure 10 – Object Persistence - 3D Visualization of the Detail of Agent (*Pig*) Starting to be Hidden for the Observer (*Wolf*) Behind a Wall (left) and a Probabilistic Occupancy Map created (right)

Figure 10 shows the exact moment when the agent is about to disappear behind an obstacle (*wall*) and it also shows the generated occupancy map, where the height of a rectangular prism represents the probability of the occupancy of that position by the hidden agent.

4.5. Unconscious Reward Anticipation

This level focuses on emotion generation and superposition. For this the scenario with the predators "*wolf*" and prey "*piglet*") was still used. The experiments are focused on emotion generation and to confirm that the correct emotions are generated in the correct situation. Emoticons are the widely accepted for of expressing the emotion in a simulation.

I still have an agent "*wolf*" observing another agent "*piglet*" and estimating its position after it disappears behind the wolf. Once the observed agent is visible again and the observing agent can verify his expectation an emotion can be generated, also the intensity of the emotion is evaluated and the final emotional state generated. For this purpose the behaviour of the observing agent was modified and once it is hidden, it can decide to turn back and continue its motion counter clockwise, which will lead to surprise in the observing agent. The Figure 11 and show two different situations with either confirmed or unconfirmed expectations and the corresponding emotion.

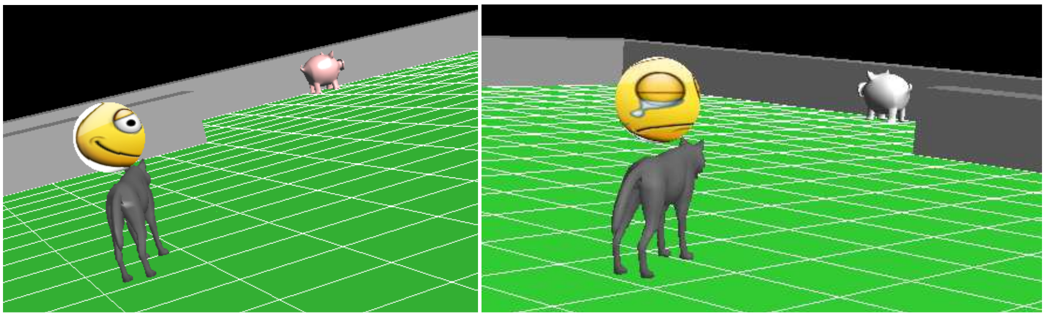

Figure 11 – Emotion Generation - 3D Visualization of the Agent (*Pig)* Reappearing Where Expected (left) and where NOT expected (right) by the Observer (*Wolf)* and the Positive (left) or Negative (right) Emotion Generation

4.6. Conscious Reward Anticipation

I have copied the scenario from work of Kadleček [18] in order to compare my results and to show strengths and weaknesses of my approach. In this scenario the main actor is a Taxi agent, shown as a "*yellow van*" (or dot). This agent's goal is to pickup client agents shown as a "*red woman*" (blue triangle) and take them to their destination, the desired destination is shown as a "*white house*", the rest of the houses are shown in "*brown*" (stars). The client agent is generated at random intervals (the probability of client appearance is 1/6). At a time there can be only one client agent until the client is delivered to the final destination. The scenario contains also a "*filling station*" (red cross) as the taxi agent consumes energy by moving and transporting client agents. Both the 2D and 3D view of the scenario is shown on Figure 12.

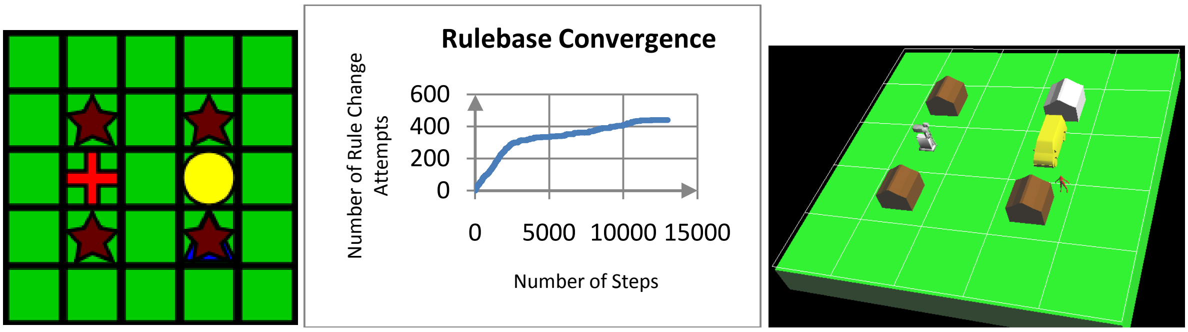

Figure 12 – The Taxi Problem Scenario Layout in 2D View (left) and 3D View (right) and The Convergence Speed of the ACS (middle)

This experiment shows that the ACS approach is capable of learning to navigate in multi-goal scenario. However it also shown that the ACS approach alone has many weaknesses. First of all the learning phase greatly depends on the random behaviour. It happens quite often that in the early phase a certain element of behaviour such as *fill up* or customer *pick up* is not learned. Then in a later phase due to better strength of the already learned behaviour these elements have smaller chance of being selected and strength improved. I have analyzed the algorithm, and I believe the root cause is in the new rule generation step. The rules that are generated by this approach are still not specific enough and do not allow to unlearn conditions under which the action has no effect. This can partially be remediated by deleting rules however that step does not help to create more specific rules. The convergence speed of the learning phase shown on Figure 12 is quite slow. As is shown on the graph below it can be also misleading. Stopping the learning process after 5000 steps would suggest a good convergence, but running the simulation for longer time revealed that there were additional 100 rule base change attempts in the next 7000 steps. The rulebase in the scenario fully converged after 12000 steps. It is due to say that thanks to the high count of #-symbols, it can even happen that a rule that is connected with drop off or pick up action is deleted and then it can never be executed again. This shows another weakness of the ACS approach that I wish to highlight. It can be concluded that ACS alone is not the optimal driving mechanism, and is rather limited. While it can help to build latent knowledge as will be shown below, it should not be used as the only decision making mechanism. But as a part of my architecture it serves its purpose as one of the eight levels.

4.7. Unconscious State Anticipation

This level focuses on latent learning. The experiment I used similar setup to Stolzmann, and even he took the experiment from an ethology example of rat learning. In this scenario an agent (in my implementation depicted as *piglet*) is placed in a simple E shaped maze. The agent starts in the middle, and has a choice to go left or to go right. The end boxes of each branch have different colour *black* and *white*. The agent is allowed a free run in the maze pre-learning phase. After a certain period the agent is placed in the left side (*black*) and is presented with reward (*food*). Then the agent is gain placed in the starting point. If the latent learning is correct the agent should be able to run straight to the left black box with anticipation of a reward there. The scenario is shown on a Figure 13 where the agent in the starting position is depicted as red circle.

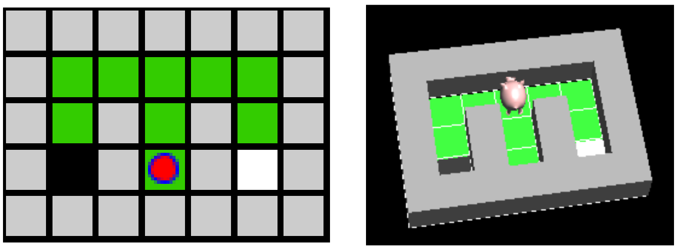

Figure 13 – Latent Learning Scenario Layout in 2D view (left) and 3D view (right)

To compare the results with Stozlmann’s work, I have conducted the same experiments. This means after the learning I executed ten times a 30 tries to observe, how many times the agent will turn left to reach the reward. The results of these experiments are shown in Table 5.

Trial number	1	2	3	4	5	6	7	8	9	10	Average
Go left (action 1)	28	26	26	28	22	27	29	24	24	26	26.0
Go right (action 2)	4	6	6	4	10	5	3	8	8	6	6.0

Table 4 – Latent Learning – Stolzmann’s Results

Trial number	1	2	3	4	5	6	7	8	9	10	Average
Go left (action 1)	24	24	25	15	26	24	19	21	20	25	22.3
Go right (action 2)	6	6	5	15	4	6	11	9	10	5	7.7

Table 5 – Latent Learning - The Statistical Evaluation of the Experiment

There are two conclusions to be made from this table. One is that the latent learning was successfully tested and gave reasonable results. Second is that compared with Stolzmann’s result, my results are slightly worse, but since that heavily depends on how the rule base was trained I cannot make a definitive conclusion on the second point.

4.8. Conscious State Anticipation

For this complex scenario I again chose to compare with work of my colleague David Kadleček [18]. Except the taxi problem already introduced among many other he also used so called Treasure Problem. Again I aim to compare my results and to show strengths and weaknesses of my approach.

In this scenario the main actor is an agent, shown as a “*robot*” (or purple dot). This agent’s goal is to reach and open the *treasure chest* (yellow triangle). This chest is not reachable because it is behind *door* (brown cross). To open the door the agent needs to place *heavy stones* (gray star) on the *pressure pads* (black rectangle). The stone needs to remain on the pressure pad for the door to open. There are three such stones available in this scenario. On top of this goal the agent needs to satisfy its own requirements for self preservation. The robot needs to supply energy in terms of food and also water. There are two static food sources shown as *red apples* (red circle) and two static sources of water shown as *blue buckets* (blue circle). Agent can refill his level of food/water by executing appropriate action of these sources (*apple/bucket)*.

The hybrid ACS-planning algorithm as will be shown below can successfully solve the treasure problem. Unlike the pure ACS approach it also significantly decreases the run time after the initial exploration and convergence of the ACS rulebase. Thanks to the learned environment the planning can then significantly reduce the run time while at the same time being able to satisfy the internal needs.

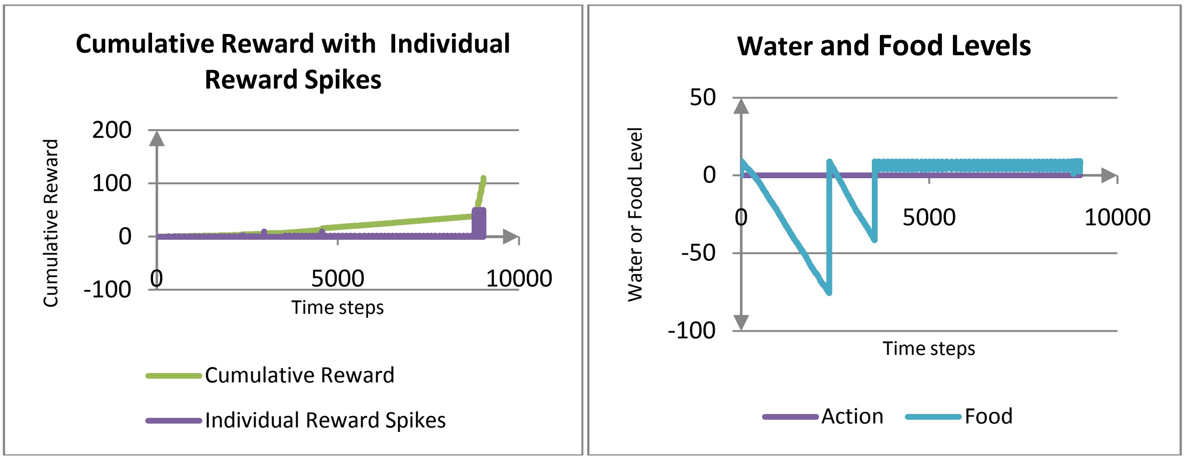

Figure 14 – Treasure Problem - The Cumulative Reward over Time with Individual Reward Spikes (left) and The Level of Water and Food during the Simulation (right)

The Figure 14 below shows the cumulative reward over time (green line). The training takes place for 8772 steps when the treasure is reached. The individual reward spikes were created by multiplying the actual received reward by 5 so it is visible and to scale with the cumulative reward. The smallest spikes show fulfilment of the internal needs of eating and drinking. The middle spikes (in steps 2961 and 4559) show the placement of the stones on the pressure pads. The highest spikes identify when the treasure is found. As previously mentioned once the treasure is once found and the environment is successfully learned the planning algorithm takes precedence and reaching the goal again becomes very fast. As mentioned it took 8772 steps to reach the goal for the first time after that it took in average 93 steps to reach the goal again.

For the same simulation the level of water and food is shown on the Figure 14. During the learning phase it is shown that the levels go deep into negative values. What is also interesting that the water level dropped below zero even after the first learning cycle at the step 8823. It is obvious that reaching the treasure (goal) for the first time does not guarantee that the algorithm covered the environment fully. Even if planning is helping significantly to keep the needs for food or water satisfied during exploration the agent can wonder so far from the source that it is not possible to the back in time. The coverage of the environment in this case is shown on the 3D mesh on Figure 15, where for each of the position I used only the action with the highest number attached. It is simplified but shows clearly the positions of the stones, the pads and the treasure and that they have been successfully mapped.

Emotions can be also generated here by plugging in the appropriate layer of my design. Since the emotion cube was already prepared then it is not a problem to add it as another property of this agent.

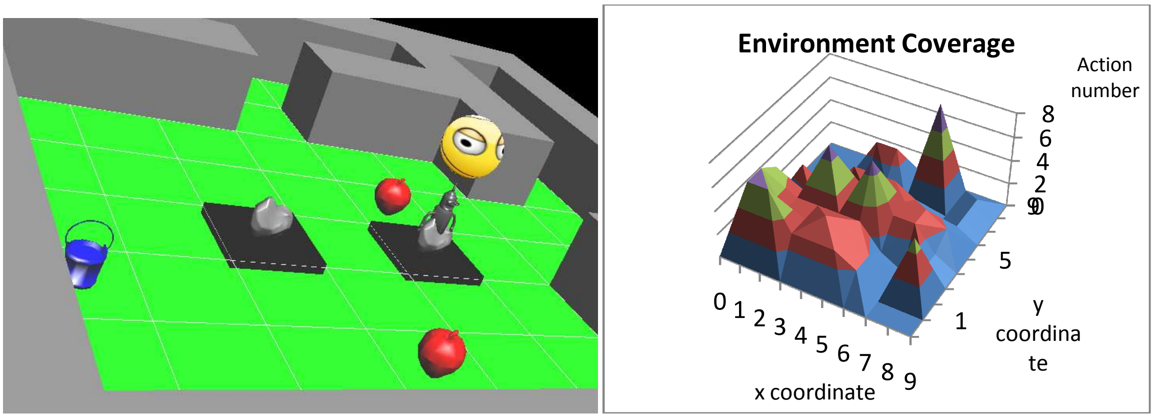

Figure 15 – Treasure Problem - Emotion Expression (left) and The Coverage of the States of the Environment (right)

Since the scenario counts only with reward and not punishment and the agent in this scenario is not focused on anticipating the reward then there is just one emotional state to generate, still to show how each factor influences each other I have embedded the emotion cube here. Figure 15 shows the situation described.

5. CONCLUSION

My research and study of “the state of the art” regarding the field of anticipation convinced me, that this topic can be a valid subject of dissertation thesis. Anticipation is perceived as not just plain prediction or estimation of the future. Anticipation in ALife sense is much more than just prediction it is utilizing the obtained information about the future for the cognitive processes such as decision control and planning. It is also about generating emotions, controlling attention and many other things described in my work.

The simulation and visualization methods used may create an impression that this work is focused on improving the artificial intelligence for computer game industry. It is due to say that this has never been the goal of this work. While my work has a value in this industry as well it is not the primary one. The main industries to apply my research in are power distribution (*Smart Grids*), Robotics (HRI) and prevention and protection of health and safety of human beings.

5.1. Fulfilment of Goals of the Thesis

I claim that all the goals set for my work were achieved and completed. I will demonstrate that by commenting how the goals were achieved and pointing to the chapter that completes the goal.

1. State of the art was mapped to a sufficient level of detail not only on technical side but also on an ethology and psychology side (namely behaviour and emotions).
2. I came up with original multi level anticipatory approach. In my work I introduced the 8-factor anticipation architecture in and described each factor of the architecture including the selected implementation. I demonstrated also the interfaces and information exchange between the different levels.
   a. The idea of anticipation playing role in more aspects of behaviour control is clearly visible in the multi level approach including eight factors.
   b. The difference between my approach and already existing anticipation approaches is highlighted thorough the work.
   c. I claim in my work that anticipation gives another dimension into the input of the action selection it was shown on experiments that adding anticipation improves the behaviour.
   d. Each level (in my work called factors) is explained in detail and each one of them has a subchapter dedicated to its description.
3. I've implemented the anticipatory behaving agents in the REPAST environment. The approach to implementation taken was to build the architecture from the lowest levels and test functions of each level separately before plugging in with the previous ones and integrating in the single architecture. The described theory, design and implementation of artificial life animat architecture have the following features.
   a. Each identified layer (factor) is build and tested separately in dedicated sub chapters.
   b. The layers are chained and resulting architecture is tested on several occasions namely in employing emotion generation.
   c. Learning implemented with help of ACS is in-time and unsupervised.
   d. The simulated environment is open.
   e. The architecture enables to generate emotions as a result of using anticipation.
   f. The implementation of anticipation brings value in small and larger scale.
4. The conducted experiments not only prove the correct function of selected algorithms, but also where possible compare with either the algorithms I based my approach on or any other comparable algorithms. The growing complexity of the approach is visible thorough the work.
   a. The suggested approach as shown in all the simulations is usable for Artificial Life domain.
   b. The achieved results exhibit in some cases better qualities than the original approach.
   c. The complexity is growing in the design and the simulations. The whole 8-factor approach is build with growing complexity.
   d. The comparison of the design with other approaches was sometimes difficult to achieve. Where it was feasible I have compared my results with results of others namely in the chapter 4.2 and 4.7 where I compared with the original approach and in chapters 4.6 and 4.8 I compared with the similar scenario but different approach.

5.2. Main Findings

My design is still unique on this field and this is mainly because two main ideas. The first idea is that anticipation is not a matter of just single mechanism (similarly to any living being). This is why I came up with my idea of 8-factor anticipation which is multi-level architecture of anticipatory behaving creature. Second idea is the introduction of consciousness into the categories of anticipation. There are multiple secondary findings as a product of implementing and evaluating my architecture.

- Probabilistic occupancy map enriched with the estimated value can provide better results in propagation of the probability.
- Emotion generation can be achieved thanks to the anticipation. I have proposed a method of Emotion Cube that can be used not only to generate emotions, but it is capable of superposition of emotions with generation of the resulting emotional state.

- The ACS approach is capable of completing the different scenarios other algorithms are able to cope with however more focus should be done on the rule specialization techniques as the future research.
- Even basic planning approaches when plugged in into the 8-factor design can significantly improve the behaviour if correctly learned. The planning algorithm is not expected to provide any exploration capabilities and as such cannot be beneficial when the rule base is not learned through other mechanisms.

5.3. Known Limitations of My Design and Future Work

On higher level the work as such is very focused on the Artificial Life simulations and as such it is not straightforward connected with real life industry applications. There are several possibilities, but the fact that the work is without the evaluation of the approach in the real life scenarios is one of the main limitations of the work. On the level of the algorithms selected and the weaknesses there I'd like to mention several things. The Anticipatory Classifier System selected and implemented in my work exhibited multiple limitations that I pointed out in the respective chapters. In a summary the low level of rule specialization leads to several disadvantages such random behaviour even after the rulebase is fully trained, removal of rules for actions that have never been tested leads to non-optimal coverage of the state space.

The field of anticipatory behaviour for Artificial Life is a dynamically developing area and it still puzzles researchers and presents more questions than one work can answer and I believe that my work is a contribution not only to the ALife area. There are multiple areas that deserve more attention and further research. Among the first ones the 8th factor of my design was leaves space for future work and improvements of the whole architecture. This is the main area for future work namely on deliberation techniques and reasoning about self and other objects. The cooperation of the factors (levels) was in my work almost inherent and given by design, a more formal interface description would aid replacement of one algorithm with another without drastically changing the rest of the architecture. I'm also already considering the possibility of adding other factors and thus extending the work even beyond 8 factors.

List of literature used in the thesis statement

[1] ROSEN, R. Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations. 1st edition. Pergamon Press, 1985. 436 p. ISBN 0-08-031158-X.

[2] TOLMAN, E. Purposive behavior in animals and men. 1st edition. New York: Century, 1932. ISBN 978-0891975441.

[3] RESCORLA, R. Associative relations in instrumental learning: The eighteenth Bartlett memorial lecture. Journal of Experimental Psychology, 1991, vol. 43, no. 1. p. 1 - 23. American Psychological Association, ISSN 1747-0218.

[4] BUTZ, M. - SIGAUD, O. - GÉRARD, P. Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories and System. 1st edition. Springer Verlag, 2003. 303 p. ISBN 3-540-40429-5.

[5] DUBOIS, D. Mathematical Foundation of Discrete and Functional Systems with Strong and Weak Anticipation. In Lecture Notes in Computer Science No. 2684 - Anticipatory Behavior in Adaptive Learning Systems. Heidelberg: Springer Verlag, 2004. p. 110 - 132. ISBN 3-540-40429-5.

[6] MARTINHO, C. Emotivector: Affective Anticipatory Mechanism for Synthetic Characters. Lisbon: Instituto Superior Técnico, 2007. PhD Thesis.

[7] KOHOUT, K. - NAHODIL, P. Reactively and Anticipatory Behaving Agents for Artificial Life Simulations. In Computing Anticipatory Systems. Editor Dubois M.D. AIP, 2010. p. 84 - 92. ISBN 978-0-7354-0858-6.

[8] DUBOIS, D. Review of Incursive, Hyper incursive and Anticipatory Systems Foundation of Anticipation in Electromagnetism. In Proceedings of 3rd International Conference on Computing Anticipatory Systems (CASYS 1999). Editor Dubois M.D. New York: American Institute of Physics, 1999. p. 3 - 30. ISBN 978-2960017960.

[9] SIGAUD, O. - WILSON, S. Learning classifier systems: a survey. Soft Computing - A Fusion of Foundations, Methodologies and Applications, 2007, vol. 11, no. 11. p. 1065 - 1078. ISSN 1432-7643.

[10] HOLLAND, J. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 1st edition. Cambridge: MIT Press, 1995. 228 p. ISBN 978-0262581110.

[11] STOLZMANN, W. An Introduction to Anticipatory Classifier Systems. In Lecture Notes in Computer Science No. 1813 - Learning Classifier Systems. Heidelberg: Springer Verlag, 2000. p. 175 - 194. ISBN 978-3-540-67729-1.

[12] BUTZ, M. An algorithmic description of ACS2. In Lecture Notes in Computer Science No. 2321 - Advances in Learning Classifier Systems. Heidelberg: Springer Verlag, 2002. p. 211 - 229. ISBN 978-3-540-43793-2.

[13] RUSSELL, S. - NORVIG, P. Artificial Intelligence: A Modern Approach. 2nd edition. Prentice Hall, 2002. 1132 p. ISBN 978-0137903955.

[14] ISLA, D. - BLUMBERG, B. Object persistence for synthetic characters. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems. Editor Castelfranchi C. and Johnson W. L. New York: ACM Press, 2002. p. 1356 - 1363. ISBN 1-58113-480-0.

[15] POSNER, M. Attention before and during the decade of the brain. Editor Meyers D. E. and Kornblum S. M. MIT Press, 1993. p. 343 - 350. ISBN 0-262-13284-2.

[16] MÜLLER, H. - RABBIT, P. Reflexive orienting of visual attention: time course of activation and resistance to interruption. Journal of Experimental Psychology: Human Perception and Performance, 1989, vol. 15, no. 2. p. 315 - 330. American Psychological Association. ISSN 0096-1523.

[17] Recursive Porous Agent Simulation Toolkit [online]. URL: <http://repast.sourceforge.net> Last Revision 2010-12-16. [cit. 2011-4-10].

[18] KADLEČEK, D. Motivation Driven Reinforcement Learning and Automatic Creation of Behavior Hierarchies. Prague: Czech Technical University in Prague, 2008. PhD Thesis.

List of candidate's works related to the doctoral thesis

[K1] NAHODIL, P. - KOHOUT, K. **(50%)** Víceúrovňové pojetí anticipačního chovaní agentů (invited paper). In Cognition and Artificial Life (in Czech). Editor Jozef Kelemen et al. Opava: Silesian University, 2011. p. 163 - 172. ISBN 978-80-7248-644-1.

[K2] NAHODIL, P. - KOHOUT, K. **(50%)** Types of Anticipatory Behaving Agents in Artificial Life. In Proc. of the 24th European Conference on Modelling and Simulation. Kuala Lumpur: European Council for Modelling and Simulation, 2010. p. 145-168. ISBN 978-0-9564944-0-5.

[K3] KOHOUT, K. **(90%)** - NAHODIL, P. Reactively and Anticipatory Behaving Agents for Artificial Life Simulations. In Computing Anticipatory Systems. Editor Dubois M.D. AIP, 2010. p. 302 - 309. ISBN 978-0-7354-0858-6.

> **Response:** Based on the submission of the paper we were asked by the organizer and proceedings editor prof. Dubois to extend the paper. We were directly asked to provide our view on the weak and strong anticipation.

[K4] KOHOUT, K. **(90%)** - NAHODIL, P. Anticipation and Artificial Life. In Proceedings of CTU Workshop 2008. Prague: ČVUT, 2008. p. 90 - 91. ISBN 978-80-01-04016-4.

[K5] KOHOUT, K. **(75%)** - NAHODIL, P. Design of Anticipated Agents in Artificial Life Domain. In Kognice a umělý život VIII. Opava: Slezská univerzita, 2008. p. 163 - 172. ISBN 978-80-7248-462-1. (in Czech).

[K6] NAHODIL, P. - KOHOUT, K. **(40%)** Simulation Environment for Artificial Life Agents. In Proc. of the 22nd European Conference on Modelling and Simulation. Brussels: European Council for Modelling and Simulation, 2008. p. 131 - 136. ISBN 978-0-9553018-5-8.

[K7] KOHOUT, K. **(80%)** - NAHODIL, P. Simulation Environment for Anticipatory Behaving Agents from the Artificial Life Domain. In Lecture Notes in Computer Science No. 4722 - Intelligent Virtual Agents. Berlin: Springer Verlag, 2007. p. 387 - 388. ISBN 978-3-540-74996-7.

[K8] KOHOUT, K. **(70%)** - NAHODIL, P. - ELMAHALAWY, A. Anticipatory Behaving Creatures in the Artificial Life Domain - Preliminary Research. In Proceedings of Abstracts of the 8th International PhD Workshop (Young Generation Viewpoint). Budapest: Computer and Automation Research Institute, 2007. p. 169 - 174. ISBN 978-963-311-365-3.

Citation: KANG, Y. Electrical power anticipation for Smart Home Product based on evolutionary algorithms. In Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT). Chengdu: IEEE, 2010. p. 523 - 526. ISBN 978-1-4244-5537-9.

[K9] KOHOUT, K. Analysis and Synthesis of Behavior of Intelligent Agents in Context of Selected Parts from Artificial Life Domain. In Proceedings of CTU Workshop 2007. Prague: ČVUT, 2007. p. 106 - 107. ISBN 978-80-01-03667-9.

[K10] KOHOUT, K. **(90%)** - NAHODIL, P. Simulation of Anticipatory Behavior in the Artificial Life Domain. WSEAS Transactions on Information Science and Applications, 2007, vol. 4, no. 3. p. 568 - 575. Greece: WSEAS Press, ISSN 1790-0832.

[K11] KOHOUT, K. **(90%) -** NAHODIL, P. Simulation Environment for Artificial Creatures. Acta Polytechnica, 2007, vol. 47, no. 6. p. 46 - 50. Prague: Acta Polytechnica, ISSN 1210-2709.

[K12] ELMAHALAWY, A. - KOHOUT, K. **(25%)** - NAHODIL, P. Simulation of Behavior in ALife Domain. In Proceedings of CTU Workshop 2007. Prague: ČVUT, 2007. p. 144 - 145. ISBN 978-80-01-03667-9.

[K13] KOHOUT, K. Anticipatory Behaviour for ALife Domain. Prague: Czech Technical University in Prague, 2006. PhD Minimal Thesis.

[K14] KOHOUT, K. **(90%)** - NAHODIL, P. Simulation and visualization of anticipatory ALife agents. In Proceedings of the 7th International PhD Workshop: Interplay of societal and technical decision-making, Young Generation Viewpoint. Prague: Institute of Information Theory and Automation, 2006. p. 21 - 22. ISBN 80-903834-1-6.

[K15] KOHOUT, K. **(90%)** - NAHODIL, P. Simulation Environment for Selected Parts of Artificial Life Domain. In Proceedings of the WSEAS International Conferences. New Jersey: WSEAS Press, 2006. p. 170 - 175. ISBN 960-8457-56-4.

[K16] KOHOUT, K. **(80%)** - NAHODIL, P. Visualization of Behaviour in Artificial Life Domain. In Kybernetika a společnost na prahu XXI.století. Brno: University of Technology, 2005. p. 77 - 82. ISBN 80-214-3058-3.

[K17] KOHOUT, K. **(50%)** - NAHODIL, P. - KADLEČEK, D. - ŘEHOŘ, D. Simulation of Selected Part of Artificial Life Domain. In Proceedings of CTU Workshop 2005. Prague: ČVUT, 2005. p. 222 - 223. ISBN 80-01-03201-9.

[K18] NAHODIL, P. - KADLEČEK, D. - KOHOUT, K. **(20%)** - SVRČEK, A. Teorie a robotické aplikace umělého života. Working lecture notes. 1st edition. Prague: CVUT, 2004.

[K19] NAHODIL, P. - KADLEČEK, D. - KOHOUT, K. **(20%)** - SVRČEK, A. Theory and Robotics Artificial Life Applications for Mobile Robots Control. In Proceedings of CTU Workshop 2004. Prague: ČVUT, 2004. p. 368 - 369. ISBN 80-01-02945-X.

[K20] KOHOUT, K. Simulace chování animátů a jeho vizualizace. Diploma Thesis. Prague: ČVUT, 2004.

[K21] NAHODIL, P. - KADLEČEK, D. - KOHOUT, K. **(20%)** - SVRČEK, A. Artificial Life Simulation. Proc. of CTU WORKSHOP 2003. Prague: ČVUT, 2003. p. 364 - 365. ISBN 80-01-02708-2.

Response / No response and reviews

Responses are mentioned directly below the work that they relate to. In addition to the mentioned references my diploma thesis was cited in works of my successors in the Mobile Robot Group.

SUMMARY

My work is occupied with a specific area of Artificial Intelligence. This so far outstanding problem is anticipation theory and it’s applications in the Artificial Life. My thesis on the topic of anticipatory behaviours is employed with design of anticipatory behaviour architecture. This architecture builds on

top of the current state of the art on the field of anticipation namely works of Daniel Dubois, Martin Butz and Carlos Martinho. My work was motivated by two main ideas that can be summarized in two sentences. The first key sentence is “anticipation is not matter of a single mechanism in a living organism - it happens on many different levels”. The second pillar of my work is the fact that “anticipation happens even without voluntary control”. The combination of these two sentences is the most important point in my work and together forms my original approach to anticipation. I believe that this brings novelty for the anticipation theory. Naturally research of anticipation at the beginning of this decade was focused on the anticipatory principles bringing advances on the field itself. This allowed me to build on those, look at them from a higher perspective, and use not one, but several levels of anticipation in a creature design. In this work I will present my unified theory for anticipation and describe eight different types of anticipation ordered in a semi-hierarchical structure. I call this approach the 8-Factor Anticipation. This presents another original idea in this work and that is the agent architecture that has anticipation built in almost every function. My 8-factor anticipatory behaviour approach is based on the current understanding of anticipation from both the artificial intelligence and biology point of view. I strongly believe that some of my findings will be used and applied in specific industry applications such as Smart Grids. These are electricity networks that can efficiently integrate the behaviour and actions of all users connected to it (generators, consumers and those that do both) in order to ensure an economically efficient, sustainable power system with low losses and high quality and security of supply and safety.

RÉSUMÉ

Moje práce se zabývá specifickou oblastí umělé inteligence a to konkrétně jednou z dosud stále nedořešených problematik, totiž teorií anticipace a jejími aplikacemi využitelnými v umělém životě. Má disertační práce na téma předjímaného (anticipačního) chování se zabývá návrhem architektury realizující anticipační chování. Tato architektura těží z poznatků na poli anticipace, zejména však z prací Daniela Dubois, Martina Butze a Carlose Martinho. Výzkum v oblasti předjímaného chování v uplynulém desetiletí byl přirozeně primárně zaměřen na funkční principy samotného předjímání, jejich implementace a experimentální ověření. To mi umožnilo na těchto již probádaných principech stavět a přistoupit k nim z jiné perspektivy tedy nejen jako k jednomu mechanismu v živém organismu ale jako k souboru několika vzájemně provázaných mechanismů. Výzkum a experimenty mnoha biologů a etologů ukazují, že se děje se na mnoha různých úrovních i u velmi jednoduchých životních forem. Já k tomuto přidávám další aspekt a to že předjímané chování se děje jak vědomě tak i bez vědomého ovlivnění. Práce presentuje originální přístup k teorii anticipace samotné, kde se objevují dva hlavní nosné názory. Jedním z nich je myšlenka začlenění anticipace do více úrovní či bloků tak, aby předjímané chování se projevilo ve více aspektech i na bazálních úrovních jako je reaktivní chování. Jako celek toto dá vzniknout emergenci předjímaného chování. Druhou z myšlenek motivovanou zejména přírodními vědami je rozlišení vědomé a nevědomé anticipace. Spojením těchto dvou myšlenek vznikla v této práci původní osmi faktorová anticipační teorie. Každý z faktorů (úrovní) je detailně rozebrán jako samostatný celek a následně začleněn do celkové architektury ve spolupráci s ostatními úrovněmi. Takováto architektura je schopna zapojit předjímané chování jak ve smyslových vjemech, tak při generování emocí na základě porovnání odměny očekávané (předjímané) s odměnou fakticky obdrženou. V neposlední řadě umožňuje architektura tvorbu modelů prostředí a to jak cíleně tak i tzv. učením latentním. V prvém případě (cíleně) je například odhadovaná poloha agenta, který je smyslům skryt za překážkou. Ve druhém (latentní učení) je model naučen nevědomě na základě provádění činnosti a krátkodobě zachován pro případ potřeby. Práce se zabývá i výběrem a modifikací algoritmů pro implementaci zmíněných úrovní a diskuzí o vhodnosti zvolených algoritmů a účinnosti navržených modifikací. Pevně věřím, že některé ze závěrů práce budou dál využity v konkrétních realizacích. Jednou z možných oblastí aplikace předjímaného chování je energetika a tzv. chytré sítě (Smart Grids).