# FP6-511931

## Mind RACES

*from Reactive to Anticipatory Cognitive Embodied Systems*

## DELIVERABLE D5 (D4.1)

*Critical Comparison of Mechanisms based on Analogy, Proactive and Goal Directed Behavior*

Due date of deliverable:
**31 / 05 / 2005**

Actual submission date:
**13 / 07 / 2005**

Start date of project:
**01 / 10 / 2004**

Duration:
**36 month**

Organization name of lead contractor for this deliverable
**UW-COGSCI**

Revision:
**3**



| **Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)** | | |
|---|---|---|
| **Dissemination Level** | | |
| **PU** | Public | **X** |
| **PP** | Restricted to other programmes participants (including the Commission Services) | |
| **RE** | Restricted to a group specified by the consortium (including the Commission Services) | |
| **CO** | Confidential, only for members of the consortium (including the Commission Services) | |

| | |
|---|---|
| Document identifier: | **DELIVERABLE_WP4_N_1** |
| Date: | **13/07/2005** |
| Work package contributing to the deliverable: | **WP4** |
| Partner(s): | *UW-COGSCI, LUCS, IDSIA, OFAI, IST, ISTC-CNR, NBU, NOZE* |
| Lead Partner: | *UW-COGSCI* |
| Document status: | *Approved* |
| Deliverable identifier: | **WP4_D4.1** |

**Delivery Slip**

| | Name | Partner | Date | Signature |
|---|---|---|---|---|
| **Author** | MARTIN BUTZ | UW-COGSCI | 30/06/2005 | |
| **Verified** | RINO FALCONE | ISTC-CNR | 30/06/2005 | |
| **Approved by** | RINO FALCONE | ISTC-CNR | 08/07/2005 | |

**Files**

| Software Products | User files |
|---|---|
| MS Word™ | D4_1.DOC |

**Project information**

| | |
|---|---|
| Project acronym: | Mind Races |
| Project full title: | MIND RACES: from Reactive to Anticipatory<br><br>Cognitive Embodied Systems |
| Proposal/Contract no.: | IST-511931 |
| Project Manager: | ISTC_CNR |
| Name: | Rino Falcone |
| Address: | CNR-ISTC via S. Martino della Battaglia,44 00185 Rome ITALY |
| Phone: | +39 06 44 595 253 |
| Fax: | Fax: +39 06 44 595 243 |
| Mobile: | |
| E-mail | rino.falcone@istc.cnr.it |

# TABLE OF CONTENTS

# PART 1 – Management Overview

**Document Control**

This document is a co-production of all the partners mentioned above. First, system information, relevancies, capabilities, and references were gathered from the partners (initiated on 22.04.2005, all included descriptions received on 06.06.2005). Then, the responsible author put the information together and provided an overarching perspective on the systems' potentials – individually and in combination. This first draft version was sent to all authors for verification (07.06.2005). The final version was presented to the program coordinator on 12.06.2005. Comments from the coordinator were received on 04.07.2005. Final version was produced on 07.07.2005.

**Executive Summary**

This document classifies and differentiates all systems relevant to workpackage four: Goal-directed behavior, pro-activity, and analogy. The workpackage is mainly concerned with anticipatory (goal-directed) action decision making and action control that is controlled by predictions and analogical reasoning. Essentially, the document assesses the potential of each relevant model to predict the future, potentially including and incorporating own future actions, motivations, and emotions. The document examines and compares the relevant models with respect to their predictive and anticipatory capabilities. We assess weaknesses and strengths of the architectures in generating predictions and using those predictions for the generation of anticipatory cognitive functions. Solutions are suggested that will enable agents to predict at different time scales and at different levels of abstraction. Moreover, solutions are suggested how to enable effective top-down cognitive influence resulting in goal-directed behavior mechanisms and pro-activity.

It is shown that the different system capabilities cover the whole spectrum of the mechanisms necessary to generate the desired anticipatory, cognitive embodied systems. However, it is also shown that without effective collaboration and knowledge exchange between the partners, the capabilities are expected to stay limited. Although one part of the work relevant to workpackage four and the whole MindRACES project consists in the further development and evaluation of the predictive and anticipatory capabilities of each system in separation, the biggest challenge will be to combine the different capabilities, such as, for example, context processing capabilities with long term dependency learning capabilities and grammar learning capabilities, to induce maximally effective predictive environmental representations that are most suitable for the goal-directed behavior mechanisms, expected to be generated by the means of the predictive representations. The scenarios developed in workpackage two are expected to help stimulate the interaction of the partners further as well as to identify potential collaborations amongst them. However, this document suggests many potential collaborations and system combinations between the partners and may serve as the guidebook for the generation of competent anticipatory cognitive embodied systems that learn and exhibit effective anticipatory, goal-directed behavior exploiting their emergent predictive representations most effectively.

# PART 2 – Deliverable Content

## Introduction

This critical comparison of mechanisms based on analogy, proactive and goal directed behavior focuses on those systems that the partners are interested in investigating further during the MindRACES project. The aim of this deliverable is to contrast the capabilities of the systems – putting them into the broader project perspectives and comparing their potentials and drawbacks with respect to workpackage four: Goal directed behavior, proactivity and analogy. We use a general taxonomy to evaluate the systems with respect to their (1) predictive capabilities and their (2) anticipatory behavior capabilities, that is, their capabilities of using these predictions for improving action decision making and action control by analogical reasoning mechanisms as well as proactive and goal directed behavior mechanisms.

In this document, the word "prediction" refers to the capability of predicting future properties whereas the word "anticipation" and particularly "anticipatory behavior" refers to mechanisms that use predictions to improve other mechanisms including learning and behavior. Since anticipations cannot exist without the capability of predicting future properties, it is necessary to first understand and differentiate amongst predictive capabilities. Then, it is possible to evaluate potentials for anticipatory behavior.

Particularly, predictive capabilities are differentiated with respect to (1) the types of predictions represented, (2) the quality or accuracy of the predictions, (3) the time scales of the predictions, (4) the generality of the predictions, (5) the capability of incorporating context information and action decision information for improving predictions, (6) the focusing and attentional capabilities of prediction generation, and (7) the capability of predicting inner states. Anticipatory capabilities distinguish the influence of predictions on (I) learning, (II) attention, (III) action initiation and control, and on (IV) decision making.

The remainder of this document is structured as follows: First, we introduce the systems that are of interest to the project. Next, we introduce the taxonomy used to evaluate and contrast the different systems. Finally, we put the different systems into perspective by evaluating their capabilities and future potentials using the proposed taxonomy.

## 1 Relevant Systems

This section introduces the project- and workpackage-relevant systems giving a short informal overview over their capabilities and general architecture. The interested reader is referred to the provided references for details on the system architectures.

### 1.1 The Alvinn Phenomenon

An intriguing early implicitly anticipatory system architecture can be found in one follow-up version of the robot Alvinn (Pomerleau, 1989). A feed-forward neural network steers a car on the road. This is done in a supervised setting: the human teacher has steered the car, and the (visual) observations and his responses at every time step have been stored in the dataset, providing input-target samples. The crucial additional – anticipatory – feature of the system is that the neural network must not only learn the target responses at every time step, but also predict the observations at the next time step. This apparently "shapes" the internal structure of the feedforward neural network such that it is much easier to successfully learn the correct target responses. It essentially puts the burden of state discrimination (compare with a standard POMDP) on the next-step-predictions (where the neural configurations at every time step are then the "states"), reducing the action production to a relatively simple state-action mapping.

From this, two important lessons can be drawn: (1) Supervised learning of control sequences might be an option for learning behavior. (2) Anticipatory "shaping" of the network can be an effective tool to reduce the difficulty of learning supervised control. *Anticipatory shaping* mechanisms will be part of the future research in the MindRACES project using more advanced, potentially recurrent NNs.

Questions about the nature and limits of such shaping advantages will be investigated. It is expected that shaping via learning of state predictions will be helpful in all environments in which successful behavior patterns strongly depend on sensory input and feedback. Since this seems to be the case in basically all natural tasks, anticipatory shaping is expected to be effective most of the time.

## 1.2 Predictive Learning Models

Predictive models are at the forefront of anticipatory models. These models were used and evaluated in learning different types of predictions. However, they were usually – or only to a limited extend – applied for the generation or control of anticipatory mechanisms. The following paragraphs give introductions to the project-relevant systems.

### 1.2.1 The XCS System Architecture

The XCS classifier system evolves a set of rules, the so-called *population of classifiers*, by the means of a steady-state genetic algorithm (GA). A classifier consists of a condition part, an (optional) action part, and a prediction part. The condition part specifies when the classifier is applicable and the action part specifies which action, or classification, to execute. The prediction can be a real-valued reward prediction, a prediction vector in form of a weight vector, a symbolic prediction, or any other type of prediction that can be compared to the feedback available.

**XCS System Overview**

XCS iteratively updates its population of classifiers with respect to the successive problem instances and feedback. Each classifier maintains a relative accuracy estimate, which estimates the relative accuracy of its predictions in the problem. Fitness is derived from the accuracy measure so that XCS propagates higher accurate classifiers. Generalization is realized by niche-based reproduction in combination with population-wide deletion. Mutation and crossover induce search in the local, structural neighborhood of current subsolutions.

XCS has been successfully applied to many problem domains including RL problems (mainly Markov decision processes) (Wilson, 1995; Butz, Goldberg, & Lanzi, in press), classification problems (Binary problems and a large variety of real datasets, Butz, 2004a, Bernado-Mansilla, Garrell-Guiu, 2003), and function approximation problems (Wilson, 2002; Butz, in press). The broad applicability of the system as well as the recent confirmation that XCS can PAC-learn k-DNF problems with few additional restrictions confirms XCS's broad applicability and strong learning potential in diverse problem domains. Essentially, XCS evolves the condition structure and the consequent problem space partitioning in such a way that the partitioning is most effective for the required predictions. Predictions are optimized via gradient descent techniques and meanwhile evaluate the accuracy of the prediction itself. Thus, condition structure and prediction mechanisms should be combined most suitably for the problem at hand.

For the MindRACES project, the system is expected to serve well as (1) a general predictive online-learning mechanism as well as (2) a feature extraction mechanism that extracts features most suitable to generate accurate predictions.

### 1.2.2 Fractal-based Future Representations

The concept of Prediction Fractal Machines (PFM) origins from constructing language models (Parfitt *et al 1999*) and has also been used in genetics (Prum 1995), finance (Bühlmann 1998) or seismology (Brillinger 1994). The method creates a fractal map of the training data, from which state machines are built; the resulting models are known as Prediction Fractal Machines.

We model our idea from the concepts introduced by Parfitt *et al.* (1999), who construct language models with 'graded' grammaticality and well formed state trajectories. This method has also been studied in financial prediction tasks and creates a fractal map of the training data, from which state machines are built; the resulting models are known as Prediction Fractal Machines. Underlying this model is the notion of 'graded grammaticality' rather than absolute grammaticality.

**Graded grammaticality in Prediction Fractal Machines**

Tino's and Dorffner's method for constructing language models uses finite-context predictive models in spirit to Variable Memory Length Markov Models (VLMMs) (Tino and Dorffner 1998). The key idea is a geometric representation of candidate prediction contexts where contexts with long common suffixes are mapped close to each other, while contexts with different suffixes correspond to points lying far from each other. Selection of the appropriate prediction contexts is left to a vector quantizer. Dense areas in the spatial representation of potential prediction contexts correspond to contexts with long common suffixes and are given more attention by the vector quantizer.

The resulting model distinguishes between grammatical and ungrammatical utterance in natural English language. Parfitt *et al*. (1999) compare this method with Simple Recurrent Networks (SRNs) and measure the differences by generating minimal pairs and comparing their negative likelihoods (NLLs) per symbol with respect to the model. The experiments support a concept of 'graded grammaticality', implying that 'meaningless' data clearly produced different results than either grammatical or ungrammatical data. PFMs showed to be able to discriminate better between grades of well-formedness than simple recurrent networks (SRNs), shown by clearly differing NLLs. If a subsequence in a well-formed utterance occurs only rarely – or never – in a training set, it will have a highly associated NLL in the same way as an ungrammatical one does. This happens to occur mainly for very large corpora, since some grammatical structures are rare. Additionally, a recent finding is that during human sentence processing well-formedness is linked to conformity with expectation (see Coulson *et al 1998*) as measured by CLOZE scores.

Of course this notion of well-formedness and expectation could be useful for the MindRACES project.

### 1.2.3 Long Short-Term Memory (LSTM)

LSTM models are recurrent artificial neural network architectures that are endowed with neural gate-based structures. *Input gates* and *output gates* guard input/output access to the internal states of neurons, enabling the algorithm to maintain memory over theoretically infinitely long periods of time. The networks have been shown to effectively deal with the problem of *vanishing gradients*, which is usually a major limitation in other recurrent NN structures especially in problems, in which long-term dependencies need to be remembered. LSTM can remember and relate events far apart in time. It is thus expected to be most suitable as a prediction tool for anticipatory systems that need to detect long-term dependencies (in memory) or that have to deal with POMDP problems.

LSTM NNs use "memory units" that us a "constant error carrousel" (CEC) to backpropagate error theoretically infinitely back in time. The memory units are protected by an input gate and an output gate that multiply incoming activation and outgoing activation to effectively "gate" the memory information so that it can apply only when necessary. In later papers, an additional forget-unit is learned that makes the timing of the memory unit more precise and allows to learn from continuous streams (this especially makes the previously necessary network reset obsolete in many problems) (Gers, Schmidhuber, Cummins, 2000). Additional "peephole connections" can be used to focus the memory unit further to apply really only when necessary developing precise timing by periodical activity or counting-like activity. The addition of Kalman-filter enhanced learning (Perez-Ortiz, Gers, Eck, Schmidhuber, 2002) increases learning speed by orders of magnitude but also – of course – increase the learning complexity of the system.

### 1.2.4 Combined Bottom-Up / Top-Down Hierarchies

Adding hierarchies to anticipatory systems is expected to enhance the system's ability to abstract and generalize, while reducing the search space enormously. Hierarchically combined recurrent neural network systems, in which lower layers feed into higher layers, have been shown to effectively enhance certain abstraction and generalization properties of such systems as a whole (Riesenhuber, Poggio, 1999).

However, systems where higher layers do also influence the lower layers are even more interesting. Imagine a cascaded RNN where every layer is predicting its own state at the next time step. Higher level layers try to model the behavior of the lower level layer, correcting lower layers when the lower levels do not have a high-enough level of abstraction to reliably predict their own state. In other words, we envision systems where high-level components, where possible, anticipate and correct the state of lower level components. For example, Rao and Ballard (1997) implemented this idea by means of a form of hierarchical Kalman filter implemented in a hierarchical recurrent neural network. In their method, uncertainty about anticipations by components of other components is estimated, affecting the amount of influence the levels have on each other (when the higher level is very certain that an event is going to happen in a lower layer, it has more control over the lower layer than when it is not certain at all). Additionally, we imagine the inclusion of attention in the framework in that attention may increase the top-down influence dependent on current motivations, emotions, or intentions.

More advanced variants of such systems are expected to be implemented during the project, using techniques such as the gating mechanisms in LSTMs and the information combination mechanisms used in Kalman filters. Additionally, several hierarchies should be implemented, such as one for action and motor control and one for visual processing, as proposed recently (Poggio, Bizzi, 2004). Interactions between the two can be manipulated with Rao and Ballard's original proposition (1997), or more recent approaches, such as the mechanisms in Grimes and Rao (2005).

## 1.3 Schema Mechanisms

Schema mechanisms refer to systems that create context-action-effect structures. The structures are proposed to be used to generate action-dependent predictions effectively and to cause anticipatory behavior competence. The original ideas stem from the work of Kant (1781) and Piaget (1954). Piaget proposed to represent an object as a set of expected interactions with the environment. Besides Piaget, also Gibson always stressed that perception without action is basically meaningless (1979).

Drescher's *schema mechanism* (Drescher 1991) is a constructivist mechanism inspired by the work of Piaget (1954); it represents an object as a set of expected interactions with the environment. The schema mechanism learns to anticipate the consequence of its actions by creating three parts schemas "context/action/result"; contexts and results are items (propositions) that can assume the values "on" or "off". The schema mechanism performs mainly two operations: Empirical learning and concept invention. Empirical learning consists in finding a result item that is relevant to an action, as well as context items that make the result reliable (it uses a statistical method called marginal attribution). Concept invention consists in postulating/reifying synthetic items (objects, not directly sensed) as the "cause" of the perceptions; this property permits to improve the ontology of the system.

### 1.3.1 Schema Theories of Arbib, Arkin, and Roy

Related schematic theories (Arbib 1995, Arkin 1998) share resemblances with the concepts of *frames* in Minsky (1975) and *scripts* in Schank and Abelson (1977). Schematic theories are strongly motivated by biological and ethological models; some of the first works are in fact the attempt to replicate into robots the behavior of cockroach, mantis and rana computatrix. Schemas are conceived as coarse grained modeling unities and opposed e.g. to neural networks that are more fine grained (in some implementation they are however decomposed and ultimately realized as neural networks; however, it has to be stressed that the descriptive level is coarser). Schemas include perceptual and motor elements to form coordinate control programs: a schema can be activated by sensory input or internally generated goal states, and many arbitration mechanisms are described in literature for selecting the more appropriate one among the active ones (they share some resemblances with some action selection mechanisms (e.g. the behavior networks, Maes 1990) as well as with the subsumption architecture (Brooks 1991)). The implementation of Arkin is based on potential fields, while Arbib uses mainly (recursive) neural networks; Roy (2004, submitted, see below) uses several machine learning techniques for different, interacting modules in his approach.

Schematic theories stress the procedural knowledge (versus declarative): "a schema constitutes the long term memory of a perceptual and/or motor skill, or the structure coordinating such skills". They are especially well suited for parallel and distributed systems, since they can be seen as concurrent computing unities. Lyons and Arbib (1989) describe a language for implementing schemas in robotic applications. Schematic theories were not originally conceived for anticipation; in fact, Arkin describes their use mainly in the perspective of reactive and behavior based robotics. However, schemas embed a predictive component that is used for action selection (it could be in principle even used for offline planning and for online control of action, although these aspects are not stressed in the available literature).

The recent work of Roy (also inspired by Piercean semiotic) extends the schematic approach by coupling sensorimotor engagement, expectations, and active perception. Schemas are also more sophisticated: they are actually networks of beliefs (either analog or discrete), linked by six possible actions, that are called projections (sensors, actions, transformers, categorizers, intentional projections, and generators). The anticipatory nature of schemas is stressed: "Beliefs are both memories of what has transpired, and also predictions of what will transpire (contingent on action)". Schemas are parametric and can be concatenated or abstracted. The whole apparatus is constructive and follows the 'active perception' paradigm: perception depends on understanding the effects of movements on sensory stimulation, thus it is intrinsically based on anticipations. The robot "Ripley" (as well as other robots) is equipped with (hand-made) schemas that provide paths from actions to anticipated sensations. For example, the following picture shows a schema for active sensing of compliance through grasping.

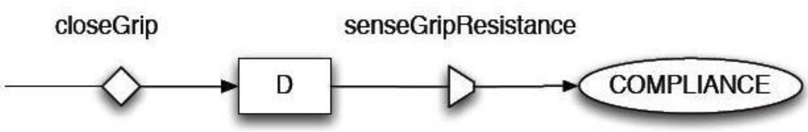


**Figure 1**

Roy introduces schemas for objects, properties, events and situation; all these representational primitives expressed in the same formalism: in this way "beliefs about concrete objects (e.g., cups) [can] be efficiently translated into expectations with respect to actions". For example, in a way that is similar to Drescher's SM, objects are represented as the set of possible interactions with it (also capturing in a natural way the notion of affordances, Gibson 1979).

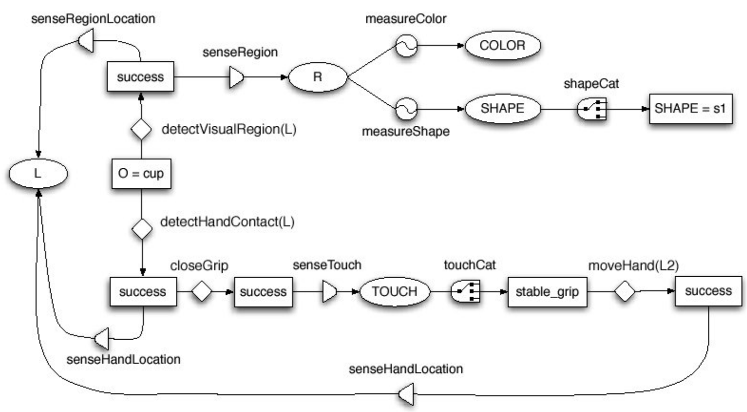


**Figure 2**

Roy aims at providing a means for grounding language by proposing "a computational path from sensing and motor action to words and speech acts". Grounding itself is defined as a causal-predictive loop, stressing the anticipatory nature of action and event sensing.

### 1.3.2 DYNA-PI models

The subsequent systems control their behavior by the means of a schema-related representation. One of the simplest ways of implementing an anticipatory system is to use models for training the controller. For example, if we have an evolutionary algorithm for evolving a (neuro)-controller, we could replace the "real world" by a predictive model of the world rather than the real world itself, and evolve on that instead of executing real actions. This could speed up learning very much. Care must be taken to occasionally execute actions in the real world in order to continually update the world model.

The basic architecture of a DYNA-PI model (Sutton, 1990) is shown in the figure below. The core of the model is the actor-critic architecture (Sutton and Barto, 1998). The evaluator implements the "evaluation function" and the actor the "action policy". The novelty is the "model of the world". This model is composed of two functions, the state transition function ST, and the reward function R (these functions might be stochastic, cf. Sutton and Barto, 1998):

$\mathbf{y}_{t+1} = ST(\mathbf{y}_t, a_t)$ $\qquad$ $r_{t+1} = R(\mathbf{y}_t, a_t)$

The world model is used to predict the consequences of actions, in terms of reward and future state of the world, one step later. The key idea of the DYNA-PI models is that an agent endowed with a world model can produce "simulated experience" and train the evaluator and actor *in the same way* it would train them through real experience. Assuming that the world model is enough accurate, if the actor is trained with simulated experience its potential performance in the real world improves, hence this process implements a form of planning in the case the actor is actually used to act in the real world.

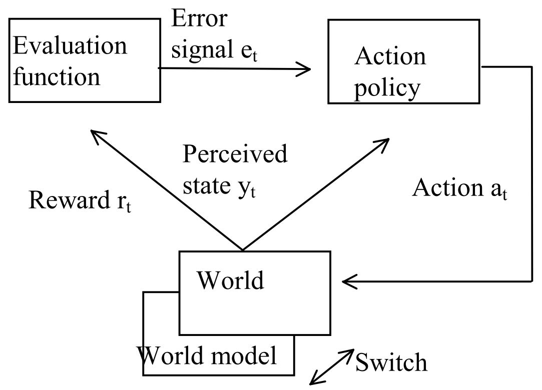


**Figure 3**

One key system that is able to generalize over states and uses an internal, generalized state population to predict reward is the XACS system (Butz, Goldberg, 2003). The system does not only evolve (via gradient-based techniques and genetic algorithms) a highly general model of its world but also approximates a state-value function with a maximally general representation. The system allows internal simulation of events that showed to improve learning significantly. Additionally, the problem of a potentially over-general model for accurate reward prediction is overcome.

It should also be noted that predicting the utility of actions has been shown to be a useful technique in many applications of which the most prominent example is the use of regular action-value-estimators in reinforcement learning. However, such predictions are just a simple form of implicit anticipation. More elaborate are techniques where a whole discounted future utility distribution is anticipated, as for example McCallum (1996) has done in order to be able to do state discrimination. Further research into this subject might be helpful, where such distributions are estimated using more advanced RNNs instead of using HMMs or decision trees.

### 1.3.3 Planning with neural networks

Baldassarre (2001, 2003) designed, implemented and investigated some predictive-planning controllers built with neural-networks and inspired by DYNA-PI architectures (Sutton, 1990). The controllers are tested with a simulated robot with a 1D camera that solves stochastic path-finding landmark navigation tasks (the robot acts in an arena with white walls and black pillar landmarks by selecting one out of 8 actions per time). The controllers, unlike the DYNA-PI architectures, are taskable: they can be assigned arbitrary goals and are capable of achieving them in an efficient way from the first time. The controllers build a "partial policy" focused on efficient start-goal paths, and are capable of deciding to re-plan if "unexpected" states are encountered. In several experiments the generalization capacity of neural networks proves important for learning.

The simple "forward planner" version of the controllers implements planning by iteratively generating "chains of predictions" from the position currently occupied by the robot, on the basis of forward models. The pseudo-experience so generated is used to train the reactive components of the system as in DYNA systems. These forward models are composed of neural networks trained to predict the consequence of execution of actions in terms of perception.

A modular version of the controllers allows the system to store information about achieved goals, and to recall such information, so as to decrease the planning burden when the same goals are assigned more than one time. In this case the goals are not only used to plan but also to furnish a "motivation" to the reactive components of the system.

Another version of the controllers implements a simple form of neural abstract planning that poses enhanced exploration and evaluations' updating capabilities. Abstraction is implemented in terms of planning on the basis of macro-actions (actions composed of n actions of the same time, e.g. north-north-north) and action execution at the primitive level.

In addition to the original DYNA system capability of executing internal rehearsal and consequently speeding up behavioral learning to gain external reward as fast as possible, Baldassarre's (2001, 2003) NN planner can implement planning with respect to the achievement of any externally *or* internally generated goal, thanks to the additional internal generation of rewards. Indeed, while DYNA learns to predict rewards associated to states, the NN planner possesses a "reward generator", named "matcher", which generates an internal reward when the system achieves its current goal.

An earlier, similar NN planner was published in the early nineties by Schmidhuber (1990b, 1990d, 1991b). Schmidhuber learned a recurrent NN model and could show capabilities of reinforcement learning and planning in dynamic environments. He also investigated the capabilities of simulating curiosity and boredom with the architecture. Interestingly, most recently Alexander Gloye won the robot soccer world cup (FU-Fighters Small Size 2004) applying parts of the NN planner in some of the modules of his modular and hierarchical control architecture (Gloye, et. al., 2004).

In yet another context, Kersting et. al (2004) applied Dyna-based ideas to a logic-based, relational world model framework. Using dynamic programming techniques and a matcher mechanism similar to the one by Baldassarre's and Schmidhuber's approaches, desired goal states are activated and then propagated through the logic-based relational world model. The first-order logic-based abstractions in the world model showed to improve behavior and planning capabilities significantly also enabling generalization to similar context.

The inclusion of matcher modules that generate internal rewards and link them to the appropriate, correlated state properties should be investigated further in the MindRACES project and also incorporated in other, related system architectures such as the XACS system.

### 1.3.4 The Anticipatory Classifier System ACS2/XACS

Anticipatory learning classifier systems (ALCSs) are learning systems that learn a generalized predictive model of an environment online. Predictive knowledge is stored in rules, called classifiers, which together represent the model. ALCSs do not only implement the schema mechanisms principle but also realize the learning principle of anticipatory behavior control outlined by Hoffmann (1993). An English version of the basic principle is available in Hoffmann (2003).

The ACS2 system (Butz, 2002) combines heuristic search with genetic generalization mechanisms to learn the model online. More recent publications successfully combined the model learning capabilities of ACS2 with the evolutionary online generalizing RL mechanism XCS (Wilson, 1995). The resulting system XACS (Butz, 2003) learns a generalized state value function using XCS-based techniques in combination with the model learning techniques of ACS2. The system shows to robustly learn optimal behavioral policies. Policy learning was improved by exploiting the knowledge of the predictive model using DYNA-based update techniques (Sutton, 1990).

XACS combines the model learning capabilities of ACS2 with the reinforcement learning capabilities of XCS. In the next paragraphs, first both systems are introduced; next the combination in XACS is described.

ACS2 learns a predictive model of an encountered environment specifying the consequences of each possible action in each possible situation. The predictions are represented explicitly in so-called classifiers, or rules, which specify condition-action-effect triples where the effect specifies which sensory inputs are expected to change after executing the specified action, given the specified conditions are satisfied. In reinforcement learning terms, ACS2 learns the state transition function of an MDP. While interacting with an environment, the population of ACS2 increasingly approximates the state transition function of the MDP. Usually, the agent starts without any prior knowledge except for the knowledge implicitly included in the coding structure and the provided actions. Initially, classifiers are mainly generated by a covering mechanism. Later, an anticipatory learning process (ALP) generates specialized classifier offspring while a genetic generalization process produces generalized offspring. Similarly, over-specialized as well as over-generalized offspring is deleted by the evolutionary mechanism.

Since ACS2 learns a predictive model, i.e. the state transition function of an MDP, dynamic programming techniques can be applied using the learned predictive model to approximate the Bellman equation representing the state value function of an optimal behavioral policy. To learn the state value function, the XCS classifier system is used (Wilson, 1995). XCS's learning is based on the accuracy of the reward prediction values of classifiers. Each classifier in XCS (in the XACS framework) represents a condition and a reward prediction. The XCS module learns an approximation of the state-value function represented in its classifier population using evolutionary learning techniques. Fitness of XCS classifiers is derived from the accuracy of their reward predictions. Reproduction in XCS is based on fitness reproducing in action sets (problem niches) and deleting from the whole population. A complete overview over the advanced learning theory on XCS can be found elsewhere (Butz, 2004).

For the interaction with ACS2 in XACS, parameters in the XCS module are updated each time XACS perceives a situation and an immediate reward, that is, they are updated each learning iteration. The expected discounted future reward is determined by using the ACS2 module to determine all possible future states and by the XCS module to determine the expected reward in these future states.

Together, it was shown that XACS learns optimal behavioral policies in problems in which ACS2 failed previously as well as in problems in which a tabular approach would take significantly longer to succeed. Recent gradient-based update mechanisms in XCS (Butz, Goldberg, & Lanzi in press) can improve performance of XACS so that XACS promises to serve as a robust learner in large, high dimensional MDP problems. With respect to psychological plausibility, it was shown that ACS2 can be used to simulate learning patterns and behavior consequences previously observed in rats (Butz, Hoffmann, 2002). Moreover, since XACS is a system that learns online and from scratch, the implementation of an enhanced XACS system is possible that comprises multiple, interacting reward learning modules that may be additionally controlled by motivational and emotional constraints. It is, for example, imaginable that – dependent on early learning experiences – the emotional patterns of the animat may evolve differently resulting in, for example, very “shy" or very “bold" animats. Future research will show to what extend the XACS system is suitable for such further investigations.

### 1.3.5 The Artificial Immune System Architecture

Also artificial immune systems (AISs) bear many resemblances with schema mechanisms. These immune system inspired architectures were originally introduced by Ishiguro and Watanabe (Ishiguro and Watanabe 1998). They are based on Jerne’s immune network theory (Jerne 1974).

The basic idea is that antibodies are formed into a network that successfully arbitrates between simple behaviors on a real robot. An antibody consists of a *paratope* defining a desirable condition and related motor-action called an *idiotope* which identifies other antibodies to which it is connected. Connection between the idiotope of antibody *x* and the paratope of an antibody *y* stimulates the antibody *y*. Links between antibodies in the network can either be evolved by a genetic algorithm (Watanabe et al. 1998) or formed via on-line adaptation mechanism which provides reinforcement signals to links (Michelan and von Zuben 2002).

The AIS architecture for autonomous robot control handles behavior arbitration constructed in such a way that its links express *sequences of actions.* A path in the network represents both a past history of robot actions (i.e. an episodic memory) and information useful for planning and anticipatory capabilities. A related line of research to AIS is that of the application of *classifier systems* to robot-control. Webb et al. (2003) have used a classifier system to control a robotic agent in a simulated robot environment. Hart et al. (2003) realized an implementation of the AIS approach on a simulated Khepera robot and OFAI on the other hand developed a real robot implementation (Rattenberger *et al.* 2004).

**Further Details on the Architecture**

In the beginning, the system is driven by basic instincts such as a "*desire to avoid collisions*" and a "*desire to seek novelty*". The robot then should learn through experience, and the learned behaviors should gradually take over control from the instinct-driven initial system. The robot therefore needs to capture some minimal details of its experiences. In the proposed model this information is held as a collection of *rule-like associations* (RLAs).

Each RLA is a node in a network and consists of a (partial) description *C* of sensory information, a robot action command *A,* and a partial description of the sensory effects *E* of doing the action. After creation, an RLA therefore expresses some of the expected results of doing action *A* in a context C. Weighted network links express the sequencing information. For example, a sub-path involving strongly positive weights would express an episode.

In immunological terminology, antibodies correspond to these RLAs, and antigens correspond to sensory data (not necessarily just raw data); the *C* and *E* parts of an RLA can be regarded as paratope and epitope. Much as in Jerne's immune network hypothesis, connections are formed and adjusted by a process of *recognition* between the paratope of one antibody and the epitope of another, and result in stimulation and suppression of one antibody by another, according to a dynamical equation suggested by Farmer in (Farmer 1986). For further details see (Hart *et al*. 2003).

Pathways through the network can be interpreted as an episodic long-term memory. This memory maintains a record of relationship between sensory conditions, actions taken, and the consequences of those actions. This episodic memory enables the robot to learn how to interact with the environment and to naturally form and ground concepts within the RLA network. For example, when passing a corner, certain RLAs trigger and as the learning process continues, a certain episode of RLAs emerges, that will always trigger while interacting with the corner.

As described in experiments in (Hart et al. 2003) during the learning phase, the robot learns the consequences of its actions within its environment and also learns which actions lead to desired behaviors.

The RLAs also appear to capture some episodes (sequences of sensory events) in a reasonably stable manner, so that the robot could be said to have a long-term memory that maintains a record of the relationships between sensory conditions, actions taken and consequences of those actions.

Planning and anticipatory behavior has to emerge from the network: this could occur as a dynamic cascade of internal events. For example, a goal is represented as an antigen which is injected into the system. As in the immunological system, the network must respond to its antigen – the antigen (goal) remains in the system until it is satisfied. At any point in time, the external environment will consist of multiple changing data items, representing goals, sensory information and maps and internal memory states. The resulting course of action results from a chain of RLAs firing, determined by the dynamically changing concentrations of antibodies. Thus, the network effectively records chains of events that can allow a desired goal to be achieved. This leads to the emergence of more complex behaviors and to anticipatory capabilities.

Another additional approach could be explored: The RLAs associate expectations with states, therefore in theory a *virtual antigen* could be injected into the system, representing some potential goal or action, and the dynamical equations applied to determine what could be the result of such an action. By comparing the result of a number of such virtual experiments, a plan could then be selected. The network thus provides a blackboard for 'thought' experiments by the robot.

For MindRACES the AIS we will be using a multilayered RLA-Architecture. The first, RLA network layer will be purely interactive, realizing simple behaviors, driven by basic motivations. Other layers will enable higher behaviors and also anticipation. Just to give a brief preview and introduction

– a second RLA layer, supporting the "*grounding of actions to objects*" will be added and operate on top the first. This second network is then responsible for learning and grounding the objects the robot encounters.

The primary purpose of the first layer is to capture "*how to*" knowledge, which is sequences of RLAs that after a specific amount of time and training learn through reinforcement which interactions environmental features, or more specific, objects allow. The "how to" knowledge level gets its input not only from the sensor values of the AIBO robot but also from so called "virtual sensors", which are functions computed for single sensors or over a certain sensor cluster.

The second layer can be seen as description layer and in principle is a generalization of the first layer. Here the "how to" knowledge which is represented by RLAs of the first layer, is anchored in a process of supervised learning. This means that sequences of RLAs that are responsible for specific object interactions are recorded in the second layer as these specific objects. Additionally this allows the robot to demonstrate the characteristic behavior describing an object, just by performing the RLA sequence it saved for that object (e.g. "*AIBO show me what you do when you interact with a ball*") as well as the creation of a topological map of the environment which corresponds to the topology of the second layer RLA network in association with input from virtual sensors of the first layer. In this process the object sequences, and therefore object representation in the second layer, should be kept as short as possible. Figure 4 gives an example for a short sequence. This is due to the fact that the longer they get the more difficult it is to find general, valid descriptions which allow to be recognized again later in the sequences of the RLA network as the robot interacts with its environment. Of course, there are several mechanisms and algorithms that allow finding common objects even if the RLA sequences are different to some degree, and therefore also allow finding a "common denominator" of such objects, but we will not go into detail on that here.

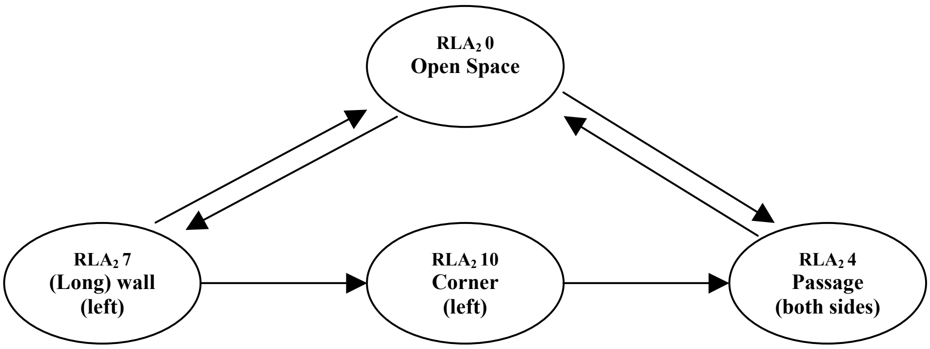


**Figure 4: Topological map of the environment corresponding to the RLA network of layer two**

### 1.4 Controllers and Emulators

Besides model building mechanisms that partially use reinforcement learning to develop competent adaptive behavior, control structures and emulators are important part of the MindRACSE project. Hereby, controllers are used that generate control signals given current input and current goal representation. Such controllers are also often referred to as *inverse models*. Emulators are used to predict the action consequences. They are also often referred to as *forward models*.

#### 1.4.1 Emulators and Simulators Proposed by Grush and Barsalou

Grush (2004) describes a mechanism based on Kalman filters (mainly used in control theory): an emulator exploits efferent copies of the control signal for producing a feedback to the system. The emulator, running simultaneously with the perception-action cycle of the system, provides *"expectations in the sense of expected results of the possible actions of the system"*; both the input and the output of the emulator are in the same format of the representation used by the system to perceive and act (ant thus they can be directly compared), as shown in the following picture (from Grush, 2004).

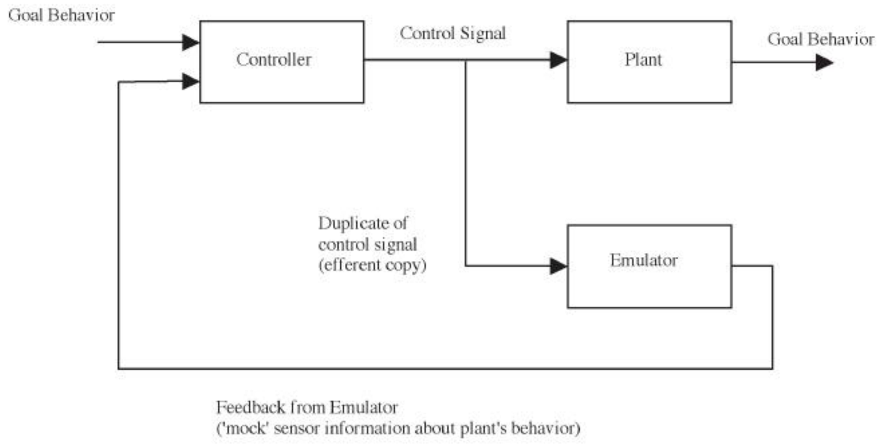


**Figure 5**

The emulator can also be used even for generating pseudo-proprioceptive information that is at the basis of motor imagery; and, assuming that the data produced by the emulator can run even the sensory areas, even visual imagery can be produced.

In a similar spirit Barsalou (1999) described (although he did not implement) the *perceptual symbol system* that is heavily based on the idea of producing modal representations by using simulators, that re-enact sensory patterns. Simulators can also produce expectations by running offline: "perceptual symbol systems cannot be implemented adequately until it is possible to run simulations either in isolation or simultaneously with perception".

Recently, the discovery of mirror neurons in neurobiology (Rizzolatti et. al., 1996, Gallese et. al., 1996, Gallese, Goldman, 1998) has strongly revitalized and influenced all the research about "productive" systems such as simulators and emulators, focusing on the strong coupling between motor activity and perceptual stimuli.

### 1.4.2 Previous Cognitive Control Structures

Numerous other, related computational models for motor learning and control have been proposed. Most of them address specific stages of movement generation, for example trajectory formation (Cruse, Steinkühler & Burkamp, 1998; Hirayama, Kawato & Jordan, 1993) or coordinate transformation (Salinas & Abbott, 1995). Others are tracking reference signals, relying on IMs and feedback controllers (Kalveram, 2004; Kalveram, Schinauer, Beirle, Richter & Jansen-Osmann, 2005; Kawato, Furukawa & Suzuki, 1987), which might be combined in a single control structure (Stroeve, 1996, 1997). Some approaches gate a number of single control structures to be able to quickly adapt to changing limb properties (Wolpert & Kawato, 1998; Haruno, Wolpert & Kawato, 2001) or to combine motor primitives (Berthier, Singh & Barto, 1992).

While each model has interesting properties on its own, none can be considered as a truly cognitive system. The model of Kawato, Furukawa & Suzuki (1987) contains three different levels but does not accept goals in arbitrary modalities. Other controllers (e.g. Cruse, Steinkühler & Burkamp, 1998) accept underspecified goals but do not include hierarchical layers. Many models contain neural networks that learn by cognitively implausible mechanism like back-propagation. It is intended to bridge these drawbacks effectively during the MindRACES project by creating a hierarchical, anticipatory cognitive model that is suitable to process any goal representation flexibly and hierarchically.

### 1.4.3 Hierarchical processing models and controllers

Many neurological and psychological studies and models suggest that cognitive information is processed hierarchically. Powers (1973) already stressed the importance of hierarchies in behavioral control and consequent computational models of cognitive systems. Just recently, Poggio and Bizzi

(2004) pointed out that hierarchical structures very likely are the key to not only sensory processing but also motor control. Available hierarchical models in vision (Riesenhuber & Poggio, 1999; Giese & Poggio, 2003) are suggested to be extended to motor control. Hierarchical top-down influences showed to have advantageous structuring effects (Rao & Ballard, 1999).

Computational motor control models showed advantages of hierarchical structures. Considering the hierarchy of the musculoskeletal system, the spinal cord, and a controller in the CNS at the top, Loeb (1999) demonstrated that the spinal cord is able to counter most perturbations on its own. However, the spinal cord also receives task-depending input from the CNS to adjust its behavior. Thus, the spinal cord makes the control task easier for the CNS because not every single muscle has to be addressed. It is sufficient to set an overall strategy to deal with most perturbations.

Hierarchical processing models were proposed by Kawato, Furukawa, & Suzuki (1987), who applied a hierarchical controller to a robot arm. The lowest level contains a simple PD-controller that can in principle handle any task. The controller is not very efficient, because the delayed feedback results in a slow control process. A second layer improves performance. As soon as a direct model of the plant is learned, it updates the control signal using the expected feedback, which is available much faster. However, it is still necessary to adjust the signal iteratively. A third level consists of an inverse model (IM) that calculates a control signal for any given goal. When the IM is accurate, the controller selects a feasible control signal instantly. In case of a failure, the lower levels induce the (slower and less effective) control. The more accurate the models in the higher levels, the more they influence the control signals. Since IMs directly determine the action necessary to obtain a desired goal, very efficient controllers can be derived (Kawato, Furukawa & Suzuki, 1987).

Despite the ubiquitous hints on the importance of hierarchical processing and the first model from Kawato and colleagues, it remains somewhat unclear why hierarchies are advantageous. One advantage may be the general decomposability of our environment due to time and space constrains (Simon, 1969; Gibson, 1979). Another advantage may lie in the potential coding invariance in higher abstract layers. Clearly, pinpointing and exploiting such properties is yet another challenge for the MindRACES project.

### 1.4.4 Gradient Inverse Models

In most of the above models, general action strategies are learned or represented in the controller. It is hardly considered what the best learning strategy is when only sparse reinforcement feedback is available. While a significant amount of research has been directed towards solving POMDPs optimally within discrete action spaces, cases with continuous actions have not been researched to a satisfactory extent.

One promising technique is, among others, *neuroevolution*, but evolutionary methods have the disadvantage that they can become very rapidly harder to solve with the size of the genome. The approach of inverse models for reaching sub-goals seems necessary instead. We need inverse models that can compute continuous actions that lead to certain sub-goals. One way of directing the learning of the controller further could be to use action gradients in recurrent neural networks. In an RNN, we want to use action gradients towards desired results instead of just error over simple prediction. Desired results could be predicted by a world model-RNN (that is, a forward model), while an action producing RNN feeds in actions into this model. Action gradients can be computed across the two networks. Action gradients could be computed towards a concrete sub-goal, or towards reward-maximization as predicted by the model. A similar approach was taken in the work of Jordan and Rumelhart (1992) who use a forward model to backpropagate and convert a location (goal-dependent) gradient into an action gradient. Schmidhuber and Huber (1991a) used similar ideas backpropagating target gradients to generate fovea trajectories. Similar backpropagation could be useful in converting reward gradients into action gradients.

Actual anticipated sub-goals could be produced by a hierarchical reinforcement learning technique, or they could be produced by something like an "optimistic predictor" predicting feasible future states with high predicted utility.

## 1.5 Context Sensitive Learners

Most learning mechanisms above do not distinguish between different (sensory, motor, context, abstracted, etc.) inputs. Although several systems have the potential of processing different inputs in different ways, usually all systems are pre-wired and mostly fully connected. Inputs are provided in the form of an input vector that does not distinguish information. The pre-processing of inputs proposed for the PFMs or XCS's and ACS's feature-based structuring could be used to evolve more abstract features. However, different inputs might have completely different properties. An intuitively very appealing distinction is the distinction between *sensory input* and *context information*. Context information is expected to evolve only once sensory input can be processed in a sufficiently competent way. Context information may then be helpful to direct sensory processing as well as action decision making towards currently relevant sensory stimuli and other, more abstracted environmental properties.

### 1.5.1 Context Sensitive Reinforcement Learning

In many problems, it seems to make sense to divide the input to a (reinforcement) learning system into two parts: one that codes for the stimulus and one that codes for the context. Current thinking in animal learning theory suggests that the stimulus and the context do not play symmetric roles in learning. Initial learning appears to be insensitive to the context, while relearning makes behavior increasingly context sensitive. By using an asymmetric learning rule of this kind, a reinforcement learning system can be designed that initially generalizes maximally between contexts and later restricts the selection of actions to contexts where they are successful. If the stimulus and context are selected and represented in an appropriate way for the task, this scheme can lead to very fast learning. It also avoids catastrophic forgetting if the context inputs are capable of coding the different learning situations.

These ideas have been applied to Q-learning to develop ContextQ which learns using a context sensitive linear approximation system (Balkeniuse, Morén, 2000, Balkenius, Winberg, 2004, Björne, Balkenius, 2005). Initial learning operates as an ordinary linear approximator, but relearning invokes the context to cancel out inappropriate associations. Apart from making the learned behavior context sensitive, the method also limits catastrophic forgetting in cases when the context can accurately predict which set of behaviors is appropriate in the different learning instances. Emotional reinforcement triggers have been investigated in this context (Morén, 2003, Balkenius, Morén, 2000, Balkenius, Björne, 2001).

### 1.5.2 Reinforcement Learning in Attention Control

Balkenius and coworkers have developed a model of attention control based on reinforcement learning (Figure 2). The goal of the model is to explain how different learning processes contribute to the control of visual attention. There are five main components in the model (Balkenius, Hulth, 1999; Balkenius, 2000; Balkenius, 2003). A *sensory buffer* codes the visual input in different ways. This coding ranges from the detection of oriented contrasts to object identity. The sensory buffer also codes the visual input using a spatial code that allows attention to be directed toward the location of a stimulus. A fixed *response system* R reacts to the location coding in the sensory buffer to produce overt orienting reactions.

The shift of attention is controlled by two basic learning systems. The *stimulus evaluation system* S* assigns a value to each stimulus based on its reward history and the *response evaluation system* R* assigns values to stimulus-response pairs. Together, the S* and R* systems implement an actor--critic (or two-process) architecture for learning (Sutton & Barto, 1998, Mowrer, 1960/1973). The learning process in S* is classical conditioning and in R* it is instrumental conditioning.

In addition, the three modules, S*, R* and R, are influenced by the context system that codes the current visual context or situation (Balkenius & Morén, 2000). The inhibitory role of the context (Balkenius, 2000, Balkenius & Morén, 2000, Morén, 2002) has been investigated as well as the excitatory influences of the context on the response evaluation system R* (Balkenius, 2003, Chun & Jiang, 1989, Chun, 2000).

The model can produce many attention phenomena including positive and negative priming, and anticipatory saccades. The model can also operate both in top-down or bottom-up mode.

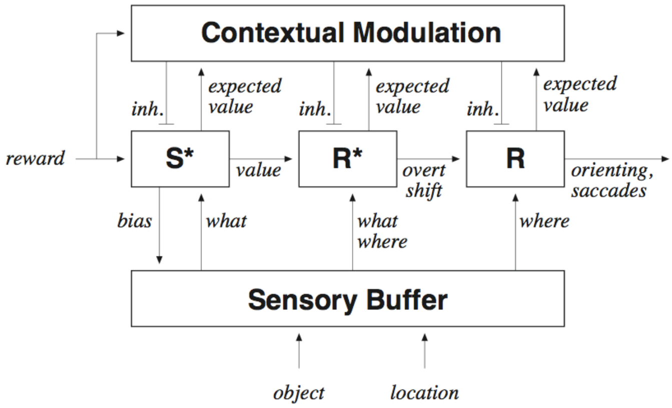


**Figure 6 Overview of the Model**

The bases for the model are four principles of attention (Figure 3). (1) Attention to object B is inhibited which disengages the focus of attention (*attention-as-inhibition*). (2) Shifting the attention and gaze g from object B to A is an action s (*attention-as-action*). (3) Object A is selected for the action a by directing the focus of attention, and gaze, toward it (*selection-for-action*). The focus of attention is used as an implicit argument for the action. (4) The focus of attention refers to the object B without explicitly representing all of its properties (deictic reference).

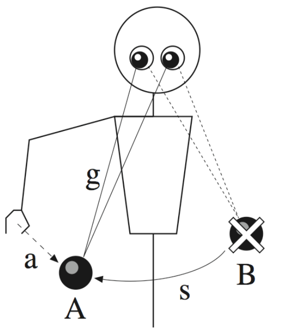


**Figure 7**

Although currently limited in its ability to produce actions, the goal of the model is to design a system that can control both attention and action in similar ways. Thus, this action-based, contextual learning system is highly relevant not only for attention processing in the sensory realm but also in the realm of action decision making and action control.

### 1.5.3 A Context Processing Model

The goal of the context processing model first described by Balkenius & Morén (2000) is to build codes for the current context based on sequences of sensory stimuli. One way to build the context codes is to use an attention mechanism. The current focus of attention acts as the stimulus while sequences of attention states make up the context. This implies that the context can be controlled by choosing how attention is allocated.

The initial model used distinct events to build bindings coding for stimuli and place relations, which where then combined into context codes. The context in turn was used to form expectations of the stimuli that would be present in each environment. These predictions were subsequently used to create new bindings when unexpected stimuli where detected.

In the later development of the model the binding is done using tensor product coding which removes the creation of context codes from the behavioral situation. In this model, contexts can still be created when expectations are not met but they can also be constructed passively from sequences of input stimuli.

## 1.6 The AMBR Model

The AMBR model (an acronym for *Associative Memory-Based Reasoning*) is a cognitive model of analogy-making based on the Dual cognitive architecture (Kokinov, 1988). A number of predictions have been generated with this model and later on confirmed by psychological experiments.

Dual (Kokinov, 1994) is a general-purpose cognitive. It comprises a unified description of mental representation, memory structures, and processing mechanisms. All these aspects of the architecture are organized around a small set of principles including (1) emergent computation, (2) micro-level hybridism, and (3) dynamics and context-sensitivity.

All processing and knowledge representation in the architecture is carried out by a cohort of small entities called *Dual micro-agents*. There is no centralized executive that controls the whole system, or makes large-scale decisions, allocates resources, resolves conflicts and so on. Instead, small-scale Dual agents interact with one another, locally, and the global behavior of the system *emerges* from the self-organizing pattern of these interactions.

Dual integrates the symbolic and connectionist approaches in a horizontal manner at the level of micro-agents, this means that each micro-agent has two parts: the symbolic part of the agent represents its knowledge while the connectionist part of the agent represents the current relevance of this piece of knowledge. The micro-agents are dualistic in another respect as well: they represent some declarative knowledge, but at the same time they are also processing units that act by sending messages, activation, etc. and in this way embody some procedural knowledge.

An important feature of Dual's operation is that it is constantly changing in response to influences from the environment. This is possible due to the emergent nature of the processing that underlies Dual's operation and to the lack of rigid and pre-programmed specification of the computation. In particular, there is no sharp boundary between the 'task' or the 'problem' given to the system to solve, and the 'context' that accompanies this problem. One and the same problem can be solved in different ways during two successive runs of a Dual-based model, in spite of the strictly deterministic character of the architecture.

The AMBR model, based on the Dual principles, integrates memory and reasoning by implementing both retrieval of episodes and mapping between the base episode and the target episode within a single model. The system exhibits interesting internal dynamics implementing a mechanism of state evolution that allows the previous states of the system to influence the specific computation that will emerge with given input. In this way the model explains and predicts priming effects.

Additionally, AMBR is context-sensitive in that changes in the environment that are reflected in the input of the system can influence the emerging computations even if they are not directly relevant to the task. The model has predicted certain context effects that have been confirmed in psychological experiments. Moreover, the system is parallel and interactive in that various subprocesses (like retrieval and mapping) run in parallel and influence each other thus allowing complex interactions to be explained.

Current most severe restrictions in the AMBR model are the scalability of the system – the system has only been applied to toy problems so far. Moreover, the perceptual abilities so far are all hand-coded. Also action and control influences were not considered so far. However, the close relation to actual cognitive processes and the interesting analogical-based activation spreading and consequent reasoning capabilities are worth to by considered further during the project and be potentially combined with other learning mechanisms of the project such as the AIS or the XACS architecture as well as recurrent NN-based methodologies such as LSTMs or hierarchical models.

## 1.7 General Believe Desires Intentions Architectures

Several computational and logical models of deliberative (intentional) and goal-directed action have been proposed. In all those models a goal is considered as an anticipated mental representation of a result that is the starting point for selecting the appropriate plan or action for achieving it, guiding planning, guiding performance, stopping the behaviour (see Rosenblueth et al., 1960). It is clear that a major aim of this project is to realize these principles within the respective architectures.

The believe desires intentions (BDI) models of deliberative agents represents yet another, more symbolic representation oriented evolution of models of goal-directed systems. The main interest in developing BDI models is the characterization of the functional properties of different kinds of mental states such as beliefs, desires, goals and intentions and the analysis of their interrelationships. BDI models been developed both from a computational perspective (Rao & Georgeff 1991b) and from a logical perspective (Cohen & Levesque 1990, Singh & Asher 1993, Meyer et al. 1999). In the BDI framework the process of selecting an action for achieving a certain number of goals is modelled as well as the way an agent gets committed to execute the selected action (intention formation) and persists in having the intention until the time of action execution is achieved. In BDI models several strategies of commitment with respect to the selected (intended) action have been implemented (Kinny & Georgeff 1991) and computational solutions to the problem of intention reconsideration are proposed (Schut et al. 2004).

BDI models are not anticipatory systems in strict sense. They have not been developed for modeling anticipatory mechanisms and functions. But since they represent a rich theoretical and computational framework to analyze mental states dynamics of rational agents as well as belief and intention formation and revision, they are well suited for studying the properties of anticipatory mental states such as predictions and expectations and their relations with intentions and goals.

A very particular and anticipatory mental construct (well represented in the BDI models) is the trust (Castelfranchi and Falcone, 1998a; Falcone and Castelfranchi, 2001) both on other agents and on the world. It is the basis for delegating part of its own plan (Castelfranchi and Falcone, 1998b) for achieving the final goal to other cognitive or not cognitive agents. In this view any practical action is fact a ‘delegation’ to some process in the world: the agent (aware or not) is relying on some environmental mediation for producing the aimed result. There are implicit forms of ‘confidence’ in the environment (while acting or moving around) and explicit evaluations and expectations (trust), including an evaluation of possible dangers and risks. One of the effects of surprise is reducing implicit confidence and revising explicit trust. We also predict a relationship between the monitoring activity and the level of confidence or trust: the more one is confident the less he will monitor the process.

BDI computational models and BDI logics are well suited to analyze and implement rational and deliberative agents with the anticipatory capability to exploit different kinds of sources of information (inference, perception, questions to memory) in order to form predictions and expectations; with the capability of reasoning with predictions for selecting the best course of action for achieving a certain number of goal states and for reconsidering intentions (previously chosen actions and plans) when the chosen action is not more expected to be suitable to achieve the goal states. The necessity to integrate expectations as an additional mental state type in the BDI framework has already been addressed by Miceli & Castelfranchi (2002) and Castelfranchi & Lorini (2003) as well as by Corrêa & Coelho (1998) and logics of expectations in the BDI framework have been proposed (Vu Trân et al. 2004). A relevant issue concerns the way expectations should be conceived: either as primitive mental states with their specific functional properties in the pragmatic reasoning or as compositional mental states (formed by goals/intentions and beliefs) whose functional properties are based on the functional properties of the components. Once expectations are integrated in a richer model of motivational (intentions, desires and goals) and informational states (beliefs and knowledge) dynamics, it becomes relevant to model the top-down influence of expectations on the perception process of a rational agent. The role of anticipatory mental states in “driving” the execution of knowledge producing action (Scherl and Levesque 2003), that are aimed at verifying them, has already been stressed in previous research in active perception (Shanahan & Randell 2004; Shanahan 2005).

## 2 Taxonomy

To evaluate and contrast the different systems, we now put the systems into perspective by using taxonomies of system properties. A fundamental distinction can be drawn between predictive capabilities and anticipatory capabilities. Thus, one part of the taxonomy focuses on predictive capabilities and another part focuses on anticipatory capabilities. This section introduces the taxonomy and the questions relevant in the taxonomy for a proper system evaluation. Section 3 then investigates system capabilities and potentials with respect to the taxonomy.

## 2.1 Predictive Capabilities Taxonomy

### 2.1.1 Types of Predictions

As a first criterion we intend to distinguish which types of predictions each system is able to generate. First, it is necessary to distinguish between discrete and continuous predictions. Some systems are pure symbolic systems that depend on discrete, symbolic input. Others usually handle real-valued inputs or even combinations of either. Moreover, some systems predict (potentially discounted) reinforcement values while others predict sensory values. Besides the predictive input and output processing capabilities, it is necessary to specify if the predictive system can predict exact next input or if it has multiple levels of predictive abstractions available. Moreover, a distinction can be made between systems that are able to predict partial aspects of the successive inputs and systems that always predict all successive inputs.

### 2.1.2 Quality of Predictions

Besides the nature of the predictions available, it is necessary to distinguish between different qualities of predictions. The most important question in this respect is if the predictions are concrete predictions of one next state or if they are able to code a number of potential predictions or a range of predictions. Similarly, it is necessary to distinguish systems that are only able to produce deterministic predictions with systems that can generate noisy predictions. In relation to this, predictions may be endowed with a confidence measure is available. Finally, it needs to be assessed if the system can learn a predictive model successfully given an MDP or also a POMDP problem. In the latter case, the system needs to learn internal state representations in order to bridge ambiguous inputs.

### 2.1.3 Time-Scale of Prediction

Most systems predict the immediate, next input property. However, there are others that can predict long-term dependencies. Again others are able to generate multiple predictions on various levels of abstraction in time. For example, a hierarchical predictive system may be able to predict immediate next sensory input on the lower level, but may predict the abstract flow of the sensory input on a higher level (such as that the flying ball will hit the approaching wall and the consequences thereof).

### 2.1.4 Generalization Capabilities

Yet another highly distinguishing criterion for predictive learners is the capability of generalizing their predictions to similar events, sensory inputs, or other regular input properties. Hereby, similar situations may be characterized by sensory inputs that have a certain amount of stimuli or stimuli properties in common. Similar sequences may be generated by an identical underlying grammar. Finally, it needs to be assessed if the systems are able to identify abstract environmental properties or generalize to object-oriented predictions ignoring background information and such.

### 2.1.5 Incorporation of Context and Action Information

Although it seems very important for cognitive systems to handle (1) action information and (2) context information in different ways than bottom-up, sensory information, only few current predictive learners actually induce such distinctions. Thus, it needs to be assessed how the systems handle action information and context information. Similarly, reward information may be handled in different ways.

Finally, it needs to be clarified how these different types of information are merged, if handled separately.

### 2.1.6 Focusing Capabilities

Besides the generalization capabilities with respect to similar inputs, generalization capabilities or rather focusing capabilities are also highly important for the generation of predictions. For example, it might be very easy to predict certain parts of a visual scene (for example, constancy) but much more difficult to predict other parts in the scene (for example, a moving object or agent or the somewhat random patterns of the leaves of a tree on a windy day). Thus, a predictive system should be able to ignore or generate noisy predictions for inputs that are hard to predict whereas concrete, near exact predictions for other parts that are predictable easily. Systems will be evaluated on their capabilities of ignoring irrelevant or distracting inputs. Moreover, they will be evaluated if predictions can be generated for parts of inputs only – either spatially or in a more object-related or property-related fashion.

### 2.1.7 Internal State Predictions

Besides the predictions of inputs, it might be highly advantageous to be able to predict own internal states. More importantly, future behavior and future relevancies may give advantages when evaluated for anticipatory behavior. Also, it might be advantageous to predict future motivational and emotional states to be able to exploit opportunities and avoid upcoming disasters.

## 2.2 Anticipatory Capabilities Taxonomy

Once the predictive capabilities are assessed, we evaluate the anticipatory capabilities of the systems. Essentially, we are interested which mechanisms may exploit which predictive capabilities in which ways. We distinguish the following anticipatory capabilities.

### 2.2.1 Learning Influences

Predictions may be used to improve learning of predictions and behavior itself. We may distinguish (1) indirect learning influences, (2) direct model learning influences, and (3) direct behavior learning influences. In the first case, predictions may shape internal learning structures in certain ways that may be advantageous for future learning. In the second case, model learning may be improved by focusing learning, ignoring irrelevant or unpredictable inputs, delegate learning responsibilities more effectively, estimate feedback reliability, and bias learning to explain unexpected events. Finally, the predictive model may be used directly to structure behavioral patterns offline or use it as a planning mechanism.

### 2.2.2 Attention Influences

Since attention is more the responsibility of workpackage three and deliverable D3.1, attention will be not considered in as much detail as the other points in the taxonomy. However, it will become clear that attention is most likely not only relevant for directing sensory processing but also very relevant to controlling action decision making. Recent publications even suggest that basically the same mechanisms can produce either effect (Poggio, Bizzi, 2004). We will evaluate the systems with respect to their capabilities to use the predictions to bias their sensory processing, use context to facilitate such processing, and predispose processing taking internal emotions, motivations, or internal goals into account.

### 2.2.3 Action Initiation and Control Influences

Much more relevant for this deliverable are the potentials of using the predictive capabilities to improve action initiation and control. Hereby, we intend to distinguish between inverse modeling capabilities, that is, if the system is able to directly trigger actions or motor programs given current sensory input and current goal representations. Next, we are interested in the capabilities of action control influences in that sensory feedback might be combined with predicted sensory states to produce optimal resulting control signals. Moreover, the differences between predicted and actually occurring sensory feedback may be used to induce more effective stabilization strategies via behavior adjustments.

### 2.2.4 Action Decision Influences

Finally, action decisions will be influence by the predictive capabilities of the system. Note that we refer to influences that do not only predict simple reward values by e.g. model-free reinforcement learning (Sutton, Barto, 1998) but rather refer to actual input predictions that are used for an effective action selection. Clearly, the most obvious anticipatory action decision making influence is that of planning systems. General planners, however, are of minor relevancy to our project, since general planning is exponential in time so that general plans are not very helpful. More important is the prediction of potentially partial future aspects and the capability of adjusting behavior according to these predictions. Second, motivational influences need to be assessed. Hereby, utility information might be merged with predictions to be able to generate opportunistic behavior, shape action preparations, and predispose action decision making. Essentially, motivations may be linked to certain environmental properties. The activation of these properties may then induce the activation of action patterns that may lead to such properties and consequently bias action decision making towards executing actions that lead to such properties.

Finally, the systems may be able to use currently available knowledge of their environment and current goals not only to achieve the goal itself but also to induce epistemic, pro-active actions that are aimed at gaining information from the environment. Such behavior is typical for any search scenario in which the agent perceives only partial information about its environment and needs to execute action to gain further, currently hidden information (such as moving a magazine to see if the searched key is located under it). Similarly but more generally, curious behavior may be induced in the general search for, for example, food but also for general knowledge gain – which can result in learning improvements.

To be more precise, such curiosity-based actions are actions intended for gaining information, namely actions of which we assume that they lead to a better understanding of the world; more precisely, anticipation of information gain. When implementing curiosity, we consequently need to design a scheme where actions are selected that lead to an optimal information gain, i.e. actions of which we expect the results to best improve our own predictor. We should therefore not concentrate on just trying to improve our predictions of extremely noisy events but instead on actions that really improve our predictions.

## 3 Predictive and Anticipatory Capabilities of Relevant Systems

The project-relevant systems introduced in Section 1 are now evaluated using the taxonomy. After considering each system separately, the next section then contrasts the different system capabilities identifying many possible system combinations and enhancements highly promising and project relevant.

### 3.1 The Alvinn System

Although the Alvinn system can be considered more a principle than a particular system, its predictive capabilities and anticipatory capabilities suggest a baseline of the minimal amount of anticipation the project is interested in.

### 3.1.1 Predictive Capabilities

Two types of predictions are generated in the Alvinn system: the observations at the next time step, and the action-prediction of what the teacher-human would have done in the same case.

The current system is predicting exact actions but could certainly also be used to learn probability distribution over possible actions instead. Moreover, it predicts the next observation. Also for predicting the observations we could easily imagine producing probability distributions instead of simple predictions per se, which have higher information content.

Predictions are currently restricted to one-step predictions. Of course, chaining is possible, though.

This would have to be researched in detail. However, since Alvinn is a simple NN, it should have the typical NN generalization properties.

The focusing capabilities are error-function dependent. The gradient-descent focuses on minimizing the error, so it really tries to predict exactly what the human would have done. Unfortunately, it will also try to predict exactly the next observation, which might not be possible. So some inefficiency might arise as a result of that. Use of probability distributions as outputs might help alleviate this problem.

### 3.1.2 Anticipatory Capabilities

This method uses the prediction of the next step as information to shape the same network that produces actions. This is a form of indirect predictive learning, where the predictivity "shapes" the internal structures for learning the right actions.

Although the process of the production of actions tries to ignore irrelevant inputs, the anticipatory predictions try to predict all predictable observations. The predictive system therefore does not focus. This remains a matter of future research. There is, in the anticipatory part, no selective attention; there are no subgoals except implicitly as learned by the RNN.

As we have seen, control is generated by an RNN that is taught by supervised gradient descent. The method does not utilize any more complicated goal generation or inverse modeling mechanisms.

Behavior is (indirectly) adjusted according to predicted external states. Although there are no explicit internal goals, back-propagation of action-error makes the system pro-active, while unexpected events produce new neural configurations that lead to other actions that are (hopefully) more appropriate in the new situation.

The main lessons learned from the Alvinn system show that it is interesting to learn (internal) representations that are used both (1) for proper predictions of next states and (2) for a proper generation of next actions. Thus, goal-directed behavior may be learned easier, if the internal structures are not only used for learning predictions but also for generating action decisions. Degrees of such advantageous will be assessed throughout the remainder of this project.

## 3.2 The XCS System

### 3.2.1 Predictive Capabilities

XCS is a very general predictive learning mechanism, which can handle discrete as well as continuous predictions, can predict reward as well as sensory inputs or sensory changes, is highly noise tolerant and does not rely on determinism. The implementation of predicting sensory changes or successive sensory inputs still awaits successful implementation and application. The only restriction is that XCS was applied only to very restricted POMDP problems so far. The evolution of inner states seems quite complicated and awaits further research. Most probably, heuristic methods for an inner state representation are necessary to apply XCS successfully to complex POMDP problems.

XCS essentially generates fuzzy predictions with a confidence interval. Usually, reward values are predicted but this is not restricted as mentioned above. Currently, XCS is a flat architecture but combinations of multiple XCSs in a hierarchical fashion are under investigation. Any prediction in XCS always has a confidence measure attached to it. An XCS module predicts usually one aspect. Multiple modules may be used to predict multiple aspects or modalities. Object-oriented predictions remain to be investigated. XCS's learning in Binary classification problems is polynomial in the number of input attributes with an order of the polynomial of the size maximally of the size of the number of relevant attributes (in the extreme case of a very symmetric problem, such as an XOR problem) but usually linear in the number of input attributes. Given fitness guidance (which is present in most problems), XCS is linear in the number of attributes irrelevant for accurate predictions.

Currently XCS is flat – combinations of multiple XCSs on multiple time-scales are under investigation (Butz, 2004b).

XCS is a strongly generalizing system that ignores irrelevant attributes and identifies attributes (or activities) relevant for an accurate prediction reliably. The capability of using different condition structures also enables different types of generalizations. Sequential learning needs to be investigated further since inner states were not integrated successfully in XCS so far.

Focusing capabilities are possible but have not been further investigated so far. Niching techniques and importance-based learning mechanisms can clearly be incorporated by tuning the GA application as well as the inherent accuracy threshold. Distracting attributes are already now effectively ignored.

Actions are currently only discretely includable in XCS classifiers. Alternatively, actions may be coded as part of the condition input. However, in this case, action choice is not straight-forward any longer. Motor programs may be included coding them either discretely in advance or in an hierarchical structure over time as proposed in Butz (2004b). Predictive clues are currently handled identically. However, the condition representation of classifiers allows the incorporation of many diverse structuring tools (such as different basis structures) for different sensory inputs and potentially internal state information. All these issues strongly depend on the design and connectivity (that is, which inputs are handled in what way) of the XCS system.

Finally, the modular and hierarchical combination of the XCS system awaits the successful design and implementation.

### 3.2.2 Anticipatory capabilities

Anticipatory learning influences have not been investigated in XCS so far.

Attention processing has not been investigated any further in XCS but may clearly be included in many ways. Predictions have confidence values so that predictions with low confidence may be processed in further detail. Additionally, the modularization is possible in that different XCS modules may predict different aspects. Another module that determines task-relevancy could be used to determine which XCS modules are currently used for prediction generation.

XCS could be used as an inverse model but has not been so far. Predictions could be combined in a Kalman-filtering fashion and predictive control would be possible similarly but awaits research. Direct faster control or stabilization mechanisms may be more difficult to implement but could be realized by prioritizing classifier processing in XCS.

Curious behavioral mechanisms showed to improve XCS's learning capabilities in MDP problems. Other direct behavioral influences await future investigations and require the combination of several XCS modules or XCS modules with other learning modules such as motivational modules. One approach is outlined below for the XACS system.

In sum, XCS has strong potentials as a predictive learning system with anticipatory capabilities. Due to its flexibility in representation and potential application, it might serve as a predictive learning module in many circumstances and in different architectures and combinations with other learning modules.

## 3.3 The Prediction Fractal Machine

### 3.3.1 Predictive Capabilities

The PFM is a purely symbolic system that learns sequences of symbols and can use these symbols for predictions. Hereby, the system partially relies on determinism. The dependency can be alleviated by an effective discretization of the input stream and thus strongly depends on the input pre-processing.

The system is able to generate fuzzy predictions on multiple levels of abstraction. Confidence values for the predictions are available. Moreover, partial future aspects may be predicted dependent on the discretization mechanisms used and the number of sequential similarities.

With respect to time scales, the system can be used to predict several steps in advance but in its current state, it cannot bridge states predicting for example, the next significant event. In this sense, long term predictions are not available but can only be generated iteratively. However, the system has a clear potential of forming hierarchies, clustering similar sequence structures and thus generating higher level sequential pattern predictions.

Generalization capabilities are strong when similar sequences occur. Dependencies are detected fast in this sense, if sub-sequences are expected to have identical symbolic structures. Thus, also grammars are essentially learnable dependent on the pre-processing mechanism used and how pre-processing depends on sequence learning. Dependent on the discretizations used, abstract environmental properties can be generalized upon.

Also the incorporation of action information and context information depends on the pre-processing mechanism. In general, though, any input is handled identical in the current system and thus no distinctions are employed between different actions and sensory inputs.

Focusing mechanisms are currently not used and the system essentially tries to predict all sequence information available.

The system can predict future sensory patterns but also behavioral patterns dependent on the used coding. Thus, the system is able to predict its own future behavior. Finally, it is planned to incorporate motivational and possible emotional modules using multiple PFMs.

### 3.3.2 Anticipatory capabilities

Actual anticipatory capabilities have not been investigated so far in PFMs. However, we expect that the fractal-based structural shaping may also be helpful for action generation mechanisms. The model has the clear capability of improving behavior control programs offline executing hypothetical actions.

Attentional influences may cover all relevant points but requires more detailed investigations. It seems hard to allocate and delegate learning responsibilities in the current PFM architecture. However, if multiple PFMs are used, then a responsibility delegating module may come in handy.

Clearly, this is not an inverse model type. However, action control influences may be possible combining predicted with actual feedback information. Faster control, stabilization and behavioral adjustment may be a consequence of such processing mechanisms.

Finally, the system has clear planning capabilities potentially modifying the action decision system. Utility and predictions may be merged and action may be pre-disposed.

Similar to the XCS system, the PFM is a predictive system that is currently used to learn good state predictions and predictive sequences matching up sub-sequences. The strength of the system lies in its effective sequential representation and its generalization capabilities with respect to sub-sequences. The continuation of the project will show how many of these capabilities can be effectively used for anticipatory mechanisms as outlined above.

## 3.4 The Long Short Term Memory

### 3.4.1 Predictive Capabilities

The LSTM system is a multi-purpose predictive system with the ability to handle continuous inputs and outputs. It could be used for sensory prediction, but also for utility prediction (as has been done for reinforcement learning purposes), among other things.

With respect to the exactness of predictions, LSTM can be used in multiple ways. Since the system essentially is an RNN, standard methods from RNN literature can be used, such as error functions that allow us to output probability distributions instead of pure predictions per se, or methods to control regularization.

The method has the capability to capture long-term time dependencies, but in order to predict events far in the future or in order to predict future abstract states, we probably need to combine this method with hierarchical approaches.

The generalization capabilities of the systems show very strong capabilities especially with respect to generalizations in time and in grammatical structure. Many experiments, among others speech processing and learning grammars, showed that the system generalizes very well.

The capabilities of including action influences are shown in an application in a reinforcement learning setting, where each output estimates a Q-value for a particular action. This would be the simplest way to encode actions. However, a more interesting goal would be to let outputs encode continuous actions, using gradients to estimate how action changes would influence future discounted reward in a model of the world. Capability of including / incorporating / merging other predictive clues will have to be researched.

Just like it is the case with other RNNs, if one uses a proper error function, LSTM networks can ignore irrelevant inputs effectively. However, focusing capabilities are somewhat limited and future research is necessary to assess if the networks can be structured in such a way to generate object-oriented focusing mechanisms or property-based focusing mechanisms. In this case, again, hierarchical and further modular structures seem inevitable.

Finally, while capability of predicting future internal states, behavior, or internal relevancies have not been investigated so far, it seems plausible that LSTM nets can and should be used for future internal state prediction. Especially long-term dependencies can be expected to be detected rather easily compared to other RNN approaches.

### 3.4.2 Anticipatory Capabilities

LSTM nets are another predictive system that can be used to investigate the shaping of internal structures via backpropagation. Recently, a new method, called *Evolino*, was developed that learns LSTM weights by using evolutionary methods. This method can overcome some difficulties that arise in gradient-descent methods due to local minima. Explicit model learning or behavioral learning influences, however, have not been investigated so far, except for the capability of ignoring irrelevant inputs using effective Kalman-filter based update techniques.

Although attentional mechanisms have not been investigated in detail, the LSTM network is a usual (R)NN. Thus, irrelevancies can be addressed by the use of a proper error function. A wide range of standard (R)NN techniques can apply here.

Similar to the other predictive learners, LSTM could be used as a tool in action-gradient-based inverse modeling techniques, or used for the supervised learning of policies. The possibility to compute gradients offers many opportunities for such methods, but we need to research especially those algorithms that function in cases where no teacher information is available, only reinforcements.

While LSTM networks have not been used to influence action decision making, it seems clear that the strong predictive capabilities will also be relevant for anticipatory decision making. Sufficient information is available to trigger curious behavior, potentially epistemic actions as well as merge the predictive information with utility information or motivations.

## 3.5 Top-Down, Bottom-Up Predictive Systems

Also the last predictive system has its current strength in its predictive capabilities. The anticipatory capabilities will have to be assessed in detail throughout the project.

### 3.5.1 Predictive Capabilities

Although this system features continuous predictions, the hierarchical structure enables some form of discretization. For example, Boden (2004) discretizes the internal state of the lower-level network according to prototype vectors before feed-forwarding it to the next level. A more sophisticated approach, where discretization is achieved by, for example, a gate-based structure like LSTM has, seems more general than this approach.

Rao and Ballard (1997) present a probabilistic system. Hierarchical structures should be able not only to predict one possible next state, but a probability distribution over next possible states. Using RNNs, we can use extensions of standard adaptations of the error function in order to achieve this.

The hierarchical top-down, bottom-up interactions could enable the architecture to operate on multiple time scales, for example, lower levels may feedforward their activation to the next level when there is significant change in its state, or when it fails to predict the next state, or when there is currently high entropy in its state. This way, different levels operate on different time scales and higher levels may predict states or state transitions many steps in the future.

Hierarchies can achieve a form of abstraction and generalization. Although this is less general than simply using bigger networks (since we are constraining the architecture), the probability of finding appropriate representations might be higher – a promising aspect of the future research taken on these networks. Essentially, it is expected – as often claimed before – that hierarchically, decomposed structures are ubiquitous in our environment (Simons, 1969; Gibson, 1979). Since our project is concerned with embodied, cognitive systems – as long as sensory representations are chosen wisely – the typical structural constraints in the environment should carry over to the internal hierarchical structural predispositions.

Currently these networks do not incorporate action influences. However, Poggio and Bizzi (2004) propose an approach in this direction, suggesting forming hierarchies for the action pathway. Context information is already included in the top-down information stream. Other connections may be added with similar properties. Most recently, Grimes and Rao (2005) show an interesting enhancement of the network with respect to forward predictions.

With respect to focusing, higher level layers focus on different aspects of the prediction than the lower levels. The lower level predictable properties of observations are mostly ignored by higher levels. Moreover, modularity and connectivity in each layer may enhance the focusing capabilities induced from higher layers to lower layers further.

Finally, the system is continuously predicting changes in its own internal states – top-down influence predicts and predisposes lower level activity. If and in what way other types of internal states – such as motivational states – might be represented and predicted remains for future research investigations.

### 3.5.2 Anticipatory Capabilities

Anticipating its internal state at the next time step, higher level layers may start to influence lower level layers to enhance their modeling capabilities. Combine actions using an estimation of prediction confidences at several levels, should shape the system such that it can handle more complex abstract representations than a single-level system could have handled.

Higher levels focus their „attention“ on those predictions that the lower levels cannot handle themselves. In a different aspect, attention may actually be induced in the system by increasing the top-down activity predisposing the lower lever activities in certain ways.

It is hoped that action influences and control may be accomplished similarly to the sensory processing pathway learning action gradients. Attention would then not only predispose sensory processing but would similarly incur action control.

Similar to other predictive systems, the hierarchical system may be used to modify and influence action decisions. Similar to the attention mechanism on sensory processing and control, action decision mechanisms may be influenced. However, to accomplish this, sensory processing and action control need to be coupled effectively – maybe the greatest challenge for the MindRACES project.

## 3.6 Schema Mechanisms

### 3.6.1 Predictive Capabilities

The original SM accommodates only binary-valued sensors; but recently it has been extended to continuous values. It has also no parameters: each schema is unique (this poses many scalability problems). It can learn in POMDP environments, as shown in Holmes and Isbell (2005). It focuses on sensory rather than reward predictions. Schemas can perform discrete predictions; some components of Roy's schemas are also analogical. They focus on sensory rather than reward prediction.

Schemata focus on sensory rather than reward prediction. A schema also has no parameters: each schema is unique (this poses many scalability problems). It was recently also shown, that schemata can be learned in POMDP environments (Holmes and Isbell, 2005). Schemas can perform exact sensory prediction; in principle there can be more abstract predictions (e.g. by using hierarchies of schemas), but this use is not jet explored.

Normally schemas predict only the next step; however, hierarchies of schemas could scale up to long-term predictions – a promising direction for future research in the project.

The SM is able to generalize and to build synthetic objects. Schemas should generalize quite naturally as also seen in the XACS mechanism and the predictive fractal machine. Roy focuses on the capability of representing e.g. objects in terms of their affordances.

Schemas include perceptual and motor elements. For example, in the work of Arbib there are actually always two coupled schemas, a perceptual and a motor one. This includes also continuous actions. The main biases seem to be the goal context. In principle, schemas can accommodate other predictive clues, but this capability is not jet explored.

Except in the recent successful application to POMDP problems, predictions of internal states are not considered explicitly. However, the predictions lead to other activation triggers which can be considered as predictions of states of other internal modules. Thus, the internal activity, especially in Roy's model, is highly interactive and modular, continuously predicting states of other internal modules.

### 3.6.2 Anticipatory Capabilities

The learning mechanism is based on marginal attribution that estimates the reliability of some results after an action versus without this action. In the work of Roy, many parameters are learned, but the general structure of the schemas (e.g. object or event schemas) remains the same. Thus, the different modules do not learn interactively so that predictive capabilities are not employed to shape learning in another module. Rather, modules are trained, if at all, in a stand alone fashion (like a controller) and then are hooked together appropriately.

The essence of an SM is to perform active perception considering only those sensory aspects relevant for its activation. The sensory and goal context is used to activate some schemas, but many mechanisms exist for their selection

The SM can associate goal states to chains of actions that are believe to accomplish these goals. Drives and Goals can activate appropriate schemata. Depending on the action selection policy, there can be arbitration, selection, merging, superimposition, etc. There is also an online control phase using the anticipatory elements of the schemas to induce active control.

The SM is driven by goal states and has not special mechanisms for surprise, curiosity, etc. However, schemas can be used both for planning and online action control. At the moment the goal influences are not fully explored and mainly limited to direct action selection and triggering.

## 3.7 Neural Network-Based DYNA-Like Systems

The DYNA architecture (Sutton, 1990), and its correlated NN-based approaches (Baldassarre, 2001, 2003) taken up by the partners, reveal strong predictive and anticipatory capabilities.

### 3.7.1 Predictive Capabilities

Whereas the original DYNA architecture is a tabular lookup system that operates on discrete symbolic input, the NN-based architecture is a continuous predictive system that predicts feature-like sensory inputs. Both systems have problems with POMDP as all standard RL systems.

Both systems can use stochastic predictions and act in stochastic worlds, but the forward models of the architectures proposed so far produce deterministic predictions. In the case of the NN planner, as in all NN methods, confidence values and fuzzy predictions can be employed by using an appropriate error function and related weight update mechanisms.

Both, Sutton's original DYNA architecture as well as the NN-based versions represent concrete predictions in terms of sensorial input. Currently the system architectures are flat enabling only immediate predictions (but see Baldassarre, 2003, for a preliminary investigation of a NN planner whose predictions span two to ten times ahead). Chaining of predictions is of course possible.

While there was no generalization in the original tabular system, the NN-based systems can generalize to similar events and similar sequences but they have neither object-oriented representations (no clustering and such) nor any type of grammar representation (no recurrence).

Actions are represented in a discrete number and are used to select expert forward models dedicated to them. If continuous actions should be included, then actions should be considered as an additional input, which however would make the planning capabilities much harder. Single actions (e.g. a displacement of a given measure towards north) need to be executed by servo-motor-programs. There were no investigations so far of including other context-based information.

Within the context of original DYNA architectures, forward models have been mainly used, as a memory of past experience, to carry out extra training so as to speed up learning of specific external reward-related tasks (e.g. Lin, 1992). Instead, the NN planner generates reward internally in correspondence to (externally or internally generated) goals by means of a specific module. This module generates the internal reward by comparing experienced or "imagined" states with pursued goals. Indeed, while DYNA learns to predict rewards associated to specific tasks (specific states), the NN planner can self-generate internal reward, which is associates with any possible state that the system might happen to pursue as a goal.

During planning, the NN planner focuses on states that lie between the starting position and the goal, and those around them. However, the system does not focus on partial aspects of the sensory information.

In both original DYNA and NN planners, when in planning mode the actor is a model of itself acting in reality. In this sense, the system actually predicts its own choices in states potentially experienced in the future.

### 3.7.2 Anticipatory Capabilities

While the DYNA system is a very simple planning system that only plans (potentially stochastically distributed) concrete next states, the NN-based systems plan on predictable states. Unpredictable things are automatically not predicted. However, partial predictions are currently not possible. Planning is quite robust in the sense that it involves all states that might likely be visited during action execution. Planning takes place precisely as an offline improvement of the control policy in relation to the assigned goal.

There are no attention influences except for the planning part mentioned above.

With respect to action initiation, the system learns to associate goals with actions once a planning step has been applied successfully. In this case, goals trigger actions / motor (control) programs especially in the multi-goal version of the system when same goals are assigned more than one time. The planning process “compiles” goal-related information into the reactive components of the system.

Action decision is influenced in a planning manner in that hypotheses are generated looking ahead and distributing reward and assigning maximally effective actions to NN-based states. Goals can become motivations in the multi-goal version of the planner. Epistemic actions were not investigated so far. Curious behavior can be included easily by directing the behavior during exploration to NN regions in which the predictions have high uncertainty.

## 3.8 ACS2 / XACS

As the NN-based DYNA approach, XACS is a DYNA-based system. The system learns a predictive world model as well as behavioral policy. The flexibility of the XACS architecture may prove very useful for the MindRACES project.

### 3.8.1 Predictive Capabilities

The XACS is currently a purely discrete, symbolic learner that was applied to n-nary alphabets (where n is a small number). It combined model and RL learning to learn sensory predictions and reward predictions, for which the sensory predictions are used. The sensory predictive module relies on determinism in the environment but can ignore irrelevant, fluctuating attributes. The reward learning part is rather noise robust as several more recent studies on XCS have shown (Butz, Kovacs, Lanzi, Wilson, 2004; Butz, Goldberg, Tharakunnel, 2003). Currently, the predictive learning capabilities are restricted to MDP problems, since no internal states are used.

XACS comprises concrete predictions but also provides a certainty of its predictions for state predictions as well as for corresponding state value predictions. The predictions are generalized in that irrelevant attributes can be ignored. The predictive model is usually significantly smaller than a completely specified model. Currently, the architecture is flat without hierarchies. Irrelevant attributes are ignored and can be explicitly identified as irrelevant for accurate predictions or as unpredictable.

The experimental evaluations showed that the system can ignore irrelevant attributes and, in this case, beats learners that learn a tabular problem representation. The predictive models are always generalized and are usually much more compact than tabular approaches. The comparisons to learning results of XCS suggest that ACS2 can PAC-learn k-DNF problems as XCS can.

There are no hierarchies and predictions are currently on one time scale only. However, predictive chains can be generated so that longer term predictions are possible in a limited sense.

The generalization mechanisms in ACS2 and XCS showed to be able to focus on the attributes that are relevant for accurate predictions of the next sensory inputs and the consequent reward, respectively. Thus, regularities are detected and object clusters are expected to be identifiable by the mechanisms. Sequence learning capabilities and POMDP problem types in which internal states need to be maintained were not investigated with XACS so far.

Actions are directly included in the classifier structure which forces the usage of discrete action codes. More sophisticated actions or hierarchical, option-type action structures or motor programs were not investigated so far. Merging of sensory/reward and predictive information were not investigated so far. Contextual information was not handled so far (or treated separately from pure sensory inputs).

Distracting stimuli can be ignored as mentioned in the system description. Also the behavioral policy can be improved to cause curious (improving model and policy learning) as well as greedy behavioral patterns (improving and speeding-up policy learning). Focusing capabilities that explicitly ignore sensory aspects in a goal-oriented fashion were not investigated so far.

The system has no internal states and thus was not tested on predicting internal properties (except for the prediction of reward triggers).

### 3.8.2 Anticipatory Capabilities

Anticipatory learning influences are used in that the comparison of predictive and resulting sensory inputs is combined to shape internal structures and to derive fitness for the evolutionary generalization component. The ACS2 system combines heuristics and genetic generalization to gain the best out of heuristic machine learning principles and genetic mechanisms.

Behavior is directly influenced and improved by the learned model so that DYNA principles are applied. A list of currently least accurate predictions combined in a priority list, similar to the work done by Moore and Atkeson (1993) in their prioritized sweeping mechanism, showed to improve behavioral learning even further.

Attention is not directly implemented but realized in the classifier structure in that generalized attributes are irrelevant for prediction and are effectively ignored. Context information is not used to improve attention or sensory input processing. Also motivations and similar things are not included but may be used to prioritize the generation of predictions, possibly mediated by the XCS-based reinforcement module.

Goals can trigger actions, possibly mediated by the XCS-based RL module. However, these capabilities were not investigated further so far and the current XACS system was only tested on problems starting from the sensory input. Kalman-filtering mechanisms were not investigated so far.

The current XACS system relies on discrete inputs so that fuzzy inputs cannot be handled with the current system.

XACS uses its predictions for behavioral decisions. Currently only one RL module was implemented but different RL modules for different motivations are easily interpretable and combinable in the XACS framework. Thus, the system has strong potentials to study multiple motivational influences and emotional integrations. Utility and predictions consequently can be combined to yield opportunistic behavior. Curious behavior is implemented in the system. However, epistemic actions and pro-activity are currently out of reach because no internal states were used so far.

In sum, the XACS system has a lot of potential for the project but may be combined with other learning mechanisms and the modules in XACS may be substituted by other modules (such as the ACS2 module by another, sensory-predictive XCS-based module), which may be more noise robust or may also be able to handle POMDP problems.

## 3.9 Artificial Immune Systems

As a schema-based mechanism, the system seems to have significant potentials for effective anticipatory mechanisms for decision making, goal-directed behavior, and control.

### 3.9.1 Predictive Taxonomy

The RLA-network on the first layer captures basic "how-to" knowledge, works with rewards (internal reward, by reinforcement learning) and by external reward, given by the system, if expectations of sensory predictions occurs.

The AIS-system is using a queue of the recent sensor history. Taking a moving average of the sensory data is a useful way of reducing or canceling the effects of noise and error and can provide a form of short-term memory. The past data is summarized by maintaining a short, fixed-length queue of sensor values, and calculating the mean of the last few sets of sensor values. The process continues by advancing one time step and calculating the mean of the next few sets of sensor values, dropping the first set of sensor values. A moving average history length of 5 sets was in the version implemented on the KURT2 robot found to give fairly good results (Rattenberger et al. 2004); large history lengths (8 or above) caused the moving average values to change too slowly to be of much use, while very small history lengths (3 or less) were not so effective at smoothing sensor noise and error. The contents of the queue provide raw material from which candidate RLAs can be built. A moving average history of 5 time steps was seen to give good results.

On higher levels, the AIS will also support object-oriented and/or spatial focusing capabilities. As soon as the concept of a "ball" (e.g. the pink AIBO ball) has been formed, by interacting with it, the activation of the according concept on the second level could trigger object-oriented focusing capabilities, by generating a hypothesis, that the object appearing and interacting with is already known.

The introduced architecture follows the principles according to interactivism. Action codes will not simply be included as an additional input, but will constitute a special status in the predictive architecture.

### 3.9.2 Anticipatory Capabilities

When determining robot behavior on the first level of the RLA-architecture, a hypothesis is made by choosing an antibody, fitting the current sensory situation and executing the action, represented by that antibody. After executing the antibody (which is not restricted to one time step), the antibody is being evaluated for accuracy. Therefore it is being determined if the paratope (expected sensory outcome) of the previously chosen antibody was accurate, by comparing the degree of matching between the current antigen and the paratope of the previously selected antibody. If the accuracy is satisfying, then the paratope of the previous antibody was regarded to be an accurate prediction of the next sensory condition. In addition to checking the accuracy of the outcome a number of factors are also considered in determining whether the antibody should receive positive reinforcement. This is done mainly by giving reinforcement according to the accuracy of the performed behavior considering the robots current motivation.

On higher AIS network levels, concepts will be formed (e.g. the concept of a ball). Once such concepts are formed, hypothesis will be formed when interacting with the environment about the nature of an object the robot starts interacting with. The agent might then reach a purely action-based perception starting from an experience-based perception model. Predicted inputs will be filtered and unexpected inputs will then be processed further in detail.

Once the robot evolves, higher behaviors will take over the initially instinct-driven behaviors. This also leads to selective attention, when for example the robot's task is to or to seek a ball - anticipating it behind a wall. When approaching an location where the ball is expected, all pink objects will primarily focus the agent's attention.

Individually, connections between RLAs can express a temporal association between RLAs. A strong positive connection between two antibodies x and y means that if RLA x fits the current situation then RLA y is a possible candidate to describe a subsequent situation (this applies for all layers in the hierarchy). Thus they can capture some aspects of episodic memory. Strongly connected paths in the network represent a history of robot actions i.e. an episodic memory. This selectively improves processing as for already known (and thus predicted) situations, less processing is necessary, until unexpected input arrives. Bandura describes this as follows: "If one had to think before carrying out every routine activity, it would consume most of one's attention and create a monotonously dull inner life. Efficient functioning requires a mix of routinized and mindful action". (Bandura 1999). A well known example from daily life is the escalator. When approaching a deactivated escalator we often experience some bafflement before switching from routine to full attention and mindful actions.

Action initiation and control influences may be incorporated in several ways. Briefly summarized, the virtual antigen would allow the agent to associate goals with actions, by processing the antigen and generating a sequence of antibodies, leading to the successful handling.

RLAs code (at least on the first hierarchy level) for an environmental situation C, the action A to be performed and an expectation E what will happen after executing action A under condition C providing feedback whether the expectation and the performed action (depending on the motivation/goal/task) were viable.

The episodic nature of the RLA system inherently enables exploitation of differences between predictions and sensory feedback for faster behavioral adjustments and faster stabilization, stabilization on lower levels, when actions and consequences of simple interactions are learned, and behavioral adjustments when higher levels are learned.

As described above, when pursuing higher behaviors, e.g. finding and retrieving an intruder/object, action decisions and explicit planning for action decisions is necessary. Unexpected events will be closely examined in the project, as they are part of the proposed scenario, where at a certain level the robot will learn to deal with surprise, e.g. a ball not being at the expected place and then learning to set actions to deal with that surprise.

## 3.10 Hierarchical Controllers and Emulators

The section on hierarchical controllers and emulators comprises many systems so that an actual taxonomy-based evaluation is not as straight forward as for the other systems. We still discuss each of the potential contributions in the following.

### 3.10.1 Predictive Capabilities

The predictive capabilities strongly depend on the type of emulator and controller employed. Several mechanisms introduced in the previous sections are of clear relevancy and can be used as forward model emulator or also as inverse model controller. The predictive capabilities then have the same properties as the systems themselves. Modularity and hierarchies should be employed to enable forward models on many time-scales detecting local dependencies as well as further-reaching, more abstract dependencies in time and space.

Much more important, though, are the anticipatory capabilities embedded in the principle of combining controllers and emulators – potentially hierarchically.

### 3.10.2 Anticipatory Capabilities

The potential anticipatory capabilities of the systems are enormous but require much future research to be combined and employed effectively. Learning influences can be expected to be effective in many ways including shaping of internal structures by shared predictive responsibilities (e.g. predicting motor activity and consequent states), the Kalman-filtering based updates promise to effectively ignore irrelevant inputs, filter noisy inputs, and merge sensory inputs with sensory predictions mediated by corresponding confidence values expressed in the Kalman gain measure. Although the emulator will also be used to improve the performance of the controller indirectly, DYNA-like updates seem to have been neglected in these architectures as well as actual planning mechanisms – probably because actual full-scale planning does not seem to be necessary for competent adaptive behavior.

The Kalman-filtering based mechanisms clearly induce attentional sensory processing. However, actual top-down attentional mechanisms remain to be effectively employed.

Most important in the architectures that comprise controllers and emulators are the implications for action initiation and control. Action initiation is mediated by the combination of forward predictions, sensory inputs and current goal representations. This combination is then fed to the inverse model-based controller that triggers the corresponding, most effective action (dependent on the controller used). Although differences between predicted inputs and actual inputs have not been used so far to incur faster behavioral adjustments or stabilizations, this seems to be another challenging future research direction that will be considered in the further MindRACES project.

Finally, action decision can be influenced in many ways although most approaches so far had only one behavioral module or only one type-of goal (such as move the gripper to a certain position) that was generated. Many future challenges and potentials for successfully employing more competent patterns of adaptive behavior and cognitive systems capabilities can be expected, if the different modules of controllers and emulators are combined most effectively. This should not only lead to the capability of curious behavior patterns and epistemic actions to decrease uncertainty in the network but also can be used in combination with motivational modules shaping action decision making potentially yielding opportunistic behaviors and predisposing action decision making in general.

## 3.11 Gradient Inverse Models

Also gradient inverse models can be considered more as a general architectural principle than an actual system implementation. Thus, also in this case, the predictive capabilities depend on the predictive system employed. The anticipatory capabilities are mainly improved due to the capability of converting reward gradients to action gradients.

### 3.11.1 Predictive Capabilities

The effects of contemplated actions are implicitly predicted and backpropagated using gradients. This is done in the model-part of the system. The predictions are for concrete aspects of the environment – observations and rewards given actions. However, even more interesting would be to predict the change in internal state given a contemplated action change. Which systems are most suitable for each predictive aspect needs to be investigated further in the MindRACES project. The gradient propagation is clearly limited due to the problem of vanishing gradients (Schmidhuber, 1997), which might be alleviated by using an LSTM-based system. Other predictive capabilities strongly depend on the predictive architecture used.

### 3.11.2 Anticipatory Capabilities

The main influence of action gradients is on learning behavioral policies more effectively. Action gradients could constitute a promising learning technique for learning continuous actions by anticipating their potential effects on either future rewards or concrete goal-reaching.

In its simplest form, action gradients take in all inputs, but using RNN techniques, try to ignore irrelevant data by adjusting weights appropriately. The gradients towards goals or states of high reward could be considered a form of implicit attention toward goal states, but not real attention.

Action gradients inevitably associate goals (combined with observation/action histories) with (continuous) motor programs. (Sub-) goals trigger actions, so care must be taken to produce sub-goals in a sensible way (i.e. a form of hierarchical reinforcement learning or using optimistic predictors).

Action gradients constitute a form of implicit planning during learning (using the RNN world model as a tool during the planning process). During execution, behavior is adjusted according to previous implicit predictions of the action-gradient during learning.

### 3.12 Context Sensitive Learning Systems

Several aspects were discussed in the context sensitive systems introduced above. Although these systems have been used more intensively to guide attention, these mechanisms are clearly strongly correlated with action decision and control processes. Action control or a particular, goal-oriented action can be expected to require attention to avoid, for example, random switching of action control problems during action execution (*attention as action*). Action decision making can be expected to be strongly dependent on which sensory features and goals the system currently pays attention to. Essentially, a system that correlates a sensory processing module with a motor control module will most probably choose desired abstract sensory properties first and put attention onto them. The attentional mechanisms may then trigger the right action program to generate the sensory properties (*selection for action*).

#### 3.12.1 Predictive Capabilities

In its current form, the predictive capabilities of the context-sensitive systems are somewhat limited but certainly may be substituted by other learning mechanisms discussed above.

Currently, the systems employ discrete reward and sensory predictions. There is some noise tolerance but determinism makes the system most effective. The predictions are concrete and are endowed with multiple approximation levels of predictions. However, in its current form, there are no confidence values available for the predictions. Due to the context module, competence with respect to POMDP problems can be expected but has not been investigated explicitly.

In its current form, the systems are used on one time scale. No hierarchies in time are employed and only immediate next input is considered. However, hierarchies are certainly imaginable and the distinct goal selection via attention already somewhat comprises a hierarchy and a more distant expectation of state in itself.

The system has clear generalization capabilities to similar events and analogous structures and sequences. Further studies on the generalization mechanisms and the substitution of modules in the system by other, for example RNN or LSTM-based learning mechanisms, pose a promising challenge to the MindRACES project.

In its current form, there is a clear contextual processing module but the capabilities of the module are somewhat limited since context is processed directly as a separate input. Interactive preprocessing stages could be implemented that are used to detect the context that is most effective to distinguish different behavioral patterns or different modes of attention. Such processes pose yet another challenge to the MindRACES project but might be accomplished by an XCS-like clustering mechanism or also with a (R)NN-based mechanism. Currently, though, there is no merging of top-down predictive and bottom-up sensory information.

Actions may have a special status in the predictive architecture as part of the context or also completely separately. In this way, processing will depend on action decision making and motor programs. Discrete and continuous actions can be processed in this way.

One of the main aspects of the system is the capability of ignoring distracting inputs and incorporating attentional biases using the available context information. Seeing that these capabilities are strongly related to actual cognitive processes (Balkenius, 2003), these mechanisms deserve more intense considerations during the MindRACES project – especially also with respect to the effects of the mechanisms on action decision making and control.

Another currently lacking capability is the prediction of inner states – since attention and consequent action-selection is expected to be guided by a motivational module, such capabilities may be included in future versions of the general context-based system architecture.

### 3.12.2 Anticipatory capabilities

The anticipatory capabilities mainly lie in the predisposition and control of sensory processing (and possibly action selection and execution control). Thus, the model has direct influence on model as well as behavioral learning by putting attention on the currently considered most important sensory features and currently selected motor programs, ignoring irrelevant distracting stimuli and improving learning of the relevant stimuli or action control patterns. Additionally, co-occurring unexpected events, such as novel context information and novel sensory or motor behavior patterns can be and are correlated with the approach taken.

Interrelated with learning is the consequent effect on sensory processing. Attention is directly induced by the system using context information, consequently ignoring irrelevant attributes and improving processing of relevant attributes. The contextual clues facilitate target detection and identification. Further mechanisms concerning motivational influences or goal-oriented selection mechanisms were not investigated so far but are certainly imaginable in the given architectural framework.

Influences on action initiation, control, and decision making were not investigated so far but pose a very strong challenge to the MindRACES project. A big achievement of the project would be to further investigate the context-based processing pre-disposition processes and also apply them to action decision making, initiation, and control.

## 3.13 AMBR

The AMBR model has not been applied to action control or decision making tasks. However, its reasoning capabilities suggest also high potential in this direction. Nonetheless, currently, the predictive capabilities are the ones most noteworthy.

### 3.13.1 Predictive Capabilities

Although the existing model has only been used in problem solving tasks, prediction tasks seem to lie in close vicinity. The following are suggestions how the AMBR model of analogy-making can be used for predicting future events and specific plans for implementing them within the project.

The main idea is that a cognitive system may predict what will happen next based on an analogy with a past episode. This approach differs from other types of learning and predicting in a number of ways. (1) The prediction in the current approach is based on a single past experience, thus learning from single cases will be possible. (2) The prediction is based on structural similarity between the two cases – this allows the two cases to differ considerably at the surface level and therefore transfer from remote domains can be accomplished – this makes the cognitive system especially adaptive in new environments or facing novel tasks. The system may not have any experience in that environment or with that task and still cope with the situation because a remote analogy with another situation previously experienced. (3) The AMBR model of analogy-making is context-sensitive and therefore the predictions will be context-sensitive, i.e. faced with the same novel task and environment the systems will make different analogies in different contexts and therefore different predictions will be generated.

To further specify, predictions in AMBR may be about the relevance of a specific piece of knowledge or specific past experience, about unobserved properties of objects, about unobserved relations between objects, about the presence of unobserved objects, about unobserved actions of other agents, about the goals of other agents, about the emotional states of other agents, about the believes of other agents, or even about future events that will be caused by the current situation or by possible actions to be undertaken. Thus, the AMBR model has great predictive potentials but is strongly representation dependent. The creation of an emergent representation that is then used for activity propagation by AMBR seems to be a challenging endeavor for the MindRACES project. To achieve this, AMBR might be combined with several structuring mechanisms such as XCS, AIS, or also a pre-processed Rao-Ballard-like hierarchical network structure.

### 3.13.2 Anticipatory Capabilities

AMBR has never been used for anticipation so far. The combination with other approaches and the investigation of anticipatory capabilities of AMBR itself are another aim of the MindRACES project. Here are some examples of such mechanisms that will be developed within the project. Top-down attentional predispositions might be developed by the means of an extension of AMBR. The prediction of a new unobserved but, for example, functionally or feature-based related object based on analogy should result in visual search for such an object. Unobserved, but predicted (by analogical reasoning) aspects of an environment could lead to epistemic actions as well as the preparedness for actions matching with the environmental aspects. Similarly, new goals may be generated using analogically derived (but still unobserved) options in the environment. Even one step further would then be to execute (e.g. preventive) actions in response to predicted (but not yet occurred) events.

# 4 Discussion

The system classification by the means of the system taxonomy in the previous section points towards several immediate and longer term challenges highly relevant for the remainder of the project – and most probably for the general research endeavor of developing competent adaptive anticipatory embodied systems. In this discussion, we contrast the different systems with respect to their predictive and anticipatory capabilities and identify the most important challenges lying ahead. Hereby, the combination of several systems and system capabilities appears highly important and appealing, not only due to the fact that the MindRACES project is an interdisciplinary European project that funds several partners with differing but related expertise but even more importantly because the partners' expertise promise to add-up very nicely for the aimed creation of competent anticipatory behavior mechanisms and cognitive embodied systems in general.

## 4.1 Contrasting Predictive System Capabilities

The system categorizations show that the partners in the project have access to a rather wide variety of predictive learning systems with diverse anticipatory processing potentials in them. Although it is hard to contrast these potentials directly, Table 1 shows the most important predictive learning mechanisms, listing their predictive capabilities according to our taxonomy. All systems exhibit great but in many aspects differing predictive capabilities. Often, we decided to point out the immediate potential of a system for certain aspects of additional predictive capabilities that might be incorporated in the system – certainly the first big challenge before us in the MindRACES project. The partners are expected to continue to improve and to further analyze their respective predictive systems. The taxonomy may serve as an indicator of which most important challenges are lying ahead for each system and which aspects are the most immediate challenges that point towards successful system enhancements and improvements.

Thus, one major aspect of the project remains to be the improvement of predictive capabilities of project-relevant systems. The table suggest that there is a current lack of system competencies in several seemingly highly project-relevant aspects of predictive capabilities: (1) the development of a predictive system that is able to predict at multiple levels of abstraction; (2) the development of a system that is able to predict at multiple time scales; (3) the effective incorporation of context information for prediction; (4) the capability of directing focus more effectively. The four points are discussed in the remainder of this section.

Although several of the predictive systems have the potential of predicting multiple aspects and provide accuracy or confidence estimates of their predictions, it seems to be difficult to provide multiple predictions in parallel such as the prediction of next sensory inputs plus the prediction of the position of an object in the input or the prediction of other, often pre-processed environmental features. The hierarchical networks starting from Rao and Ballard (1997) might be an approach to realize such multiple abstract capabilities. The hierarchically combined layers, structured appropriately, may each have a different (emerging) type of abstract representation and thus also abstract prediction. It seems that the integration of other mechanisms such as the clustering-for-prediction capabilities of the XCS system or the long-term dependencies detection of the LSTM system into these hierarchical network structures points towards a very interesting challenge in the MindRACES project.

Related but not identical to the capability of predicting at multiple levels of abstract representation lies the capability of predicting at multiple levels in time. Again, hierarchical networks seem to have the most potential in this respect. However, even more important than with respect to representational abstraction is the question of how to abstract in time. To generate flexible longer time-bridging capabilities during learning, it needs to be clarified, when predictive responsibility should be delegated to the next higher level. Schmidhuber's (1992, 1993) early work points towards interesting methodologies of how to distribute learning responsibilities and predictive hierarchies. The basic principle is to delegate learning to a higher level, if the current level is well-predicting on (moving-) average but currently encounters highly ill-predicted input. Interestingly, it was recently shown that a very similar principle can serve for the effective detection and generation of options, that is, higher level motor programs, in reinforcement learning (Butz, Swarup, Goldberg, 2004, Simsek, Barto, 2004). In general, the information content received from the sensory inputs must be significantly higher and persistently high in order to delegate predictive responsibility to the higher level prediction layer. Further research in this respect seems highly important.

Another approach for multiple levels of abstraction in time is the consideration of delay in sensory feedback. Hierarchical control structures partially take these feedback constraints into account, such as the seminal work of Kawato, Furukawa, & Suzuki (1987) in which a lowest-level PD controller serves as backup in case the higher level inverse model-based controllers and forward models are incorrect or inaccurate. The combination of these principles with more competent network structures, such as LSTM or Rao, Ballard's network, points towards another big challenge for our project.

The incorporation of actual context information for prediction is also only partially realized in most of the predictive systems available. Hereby, it can be expected that context information should not be simply included as an additional lower level input, but rather should be exploited as a different type of input that serves as a focusing and predisposition mechanism in the system applied to. Thus, in the rule-based XCS system, context may pre-select currently relevant rules, or, in the LSTM system, context information may be used to open and close certain input, forget, and output gates in order to stream information flow in a context-dependent way. The usage of context information from Balkenius' context dependent attention-processing and reinforcement learning systems may serve as an inspiration of how to incorporate such mechanisms in a more flexible, neural-network-based or rule-based learning mechanism such as LSTM, NN-type DYNA, XCS, XACS or the AISs.

In relation to the incorporation of context information stands the importance of directing predictive focus more effectively. Besides context information, other information such as task relevancies or motivation-dependent information may guide and direct current predictive foci in similar ways. The focus then may not only restrict predictive sensory processing but also should help to improve the restricted sensory processing or, similarly, the restricted action processing (including decision making and control). These considerations are discussed further in the anticipatory systems considerations below.

These considerations, however, are certainly only the precursor for the actual ambitions of the project, that is, the development of competent anticipatory cognitive embodied systems. With respect to workpackage four, the ambition lies in the development of competent anticipatory decision making and control systems. To achieve this endeavor, it will be necessary to combine several predictive system competencies for the problem structures at hand and exploit these competencies to generate more effective anticipatory processing mechanisms. How this might be achieved is outline in the following section.

| ***Predictive Capabilities*** | | **XCS** | **PFMs** | **LSTM** | **Rao Ballard** | **DYNA** | **AIS** | **XACS** | **NN-type DYNA** | **Hier Contr.** | **AMBR** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Types** | ***dis./cont*** | both | cont. | cont. | cont. | dis. | dis. | dis. | cont. | cont. | both |
| | ***rew/sens/both*** | either | sens. | both | both | both | sens | both | both | both | sensory |
| | ***levels of abstraction*** | one (+pot) | one (+pot) | one (+pot) | pot. multiple | one | one (+pot.) | one | one | pot. multiple | one |
| | ***pred. of parts*** | no (+pot) | no | no (+pot) | yes | no | yes | yes | no | no | yes |
| **Quality** | ***conc/fuzzy*** | fuzzy | fuzzy | fuzzy | fuzzy | conc. | conc. | conc. | fuzzy | fuzzy | conc. |
| | ***det./noisy*** | noisy | det | noisy | noisy | det | noisy | det. | noisy | noisy | noisy |
| | ***MDP/POMDP*** | MDP | POMDP | POMDP | POMDP (pot) | MDP | MDP | MDP | MDP | POMDP (pot) | MDP |
| | ***confidence measure*** | yes | yes | potentially | yes | no | yes | yes | potentially | potentially | yes |
| **Time-scale** | ***flat / multiple levels*** | flat (+pot) | flat (+pot) | flat (+pot) | multiple | flat | flat (+pot) | flat (+pot) | flat | multiple | flat (rep. dep.) |
| | ***immediate / long term*** | immediate | immediate | both | both | immediate | immediate | immediate | immediate | both | immediate |
| **Generalization** | ***similar events*** | yes | yes | yes | yes | no | yes | yes | yes | yes | rep. limited |
| | ***similar sequences*** | no (+pot) | yes | yes | potentially | no | yes | no (+pot) | no (+pot) | yes | no |
| | ***object oriented gen.*** | potentially | no | potentially | potentially | no | potentially | potentially | no | potentially | no |
| **Context info** | ***action influences*** | concrete | concrete | extra input | potentially | concrete | concrete | concrete | concrete | yes | no (+pot) |
| | ***context info.*** | no | no | no | yes | no | no | no | no | yes | yes |
| | ***merging feedback*** | no | no | yes | yes | no | no | no | no | potentially | no (+pot) |
| | ***using reward info*** | yes | no | no | no | no | no | potentially | yes | potentially | no |
| **Focusing** | ***ignore irrelevant or distracting inputs*** | yes | no | yes | yes | no | yes | yes | yes | yes | partially |
| | ***focus learning*** | no (+pot.) | no | no (+pot.) | no (+pot.) | no | no (+pot.) | no (+pot.) | no | potentially | no |
| | ***object-oriented focus*** | no (+pot.) | no | no (+pot.) | potentially | no | no (+pot.) | no (+pot.) | no | potentially | no |
| | ***spatial focus*** | no (+pot.) | no | no (+pot.) | potentially | no | no (+pot.) | no (+pot.) | no | potentially | no |
| **Internal pred.** | ***motiv. & emotions*** | potentially | potentially | potentially | potentially | yes | potentially | yes | yes | potentially | no |
| | ***future behavior*** | potentially | yes | potentially | potentially | yes | yes | yes | yes | yes | no |

**Table 1: Relevant predictive systems classified according to various aspects of predictive capabilities.**

| ***Anticipatory Capabilities*** | | **ALVINN** | **AIS** | **XACS** | **NN-DYNA** | **Hier. Contr.** | **Inverse Gradient** | **Context Systems** |
|---|---|---|---|---|---|---|---|---|
| **Learning Influences** | *Shaping Internal Structures* | yes | no | no | no | yes | yes | no |
| | **On Model Learning** | | | | | | | |
| | Focus Learning | no | no | no | no | yes | no | yes |
| | Allocate and Delegate Learning Resources | no | no | no | no | potential | no | yes |
| | Estimate Feedback | | | | | | | |
| | Reliability | no | no | no | potential | yes | no | no |
| | Correlate co-occurring unexpected events | no | no | yes | no | potential | no | no |
| | **On Behavior** | | | | | | | |
| | Improve Behavior Policy | yes | no | yes | yes | potential | yes | yes |
| | Offline DYNA-like Rehearsal | no | potential | yes | yes | no | potential | no |
| | Planning | no | possible | yes | yes | no | potential | no |
| **Attention Influences** | **Sensory Processing Influence** | | | | | | | |
| | Ignore irrelevant information | no | yes | yes | yes | yes | no | yes |
| | Selectively Improve Processing | no | yes | no | no | potential | no | yes |
| | Process Unexpected Inputs in Further Detail | no | yes | yes | no | yes | no | no |
| | **Context Information Exploitation** | | | | | | | |
| | Target Detection | no | yes | no | no | potential | no | potential |
| | Input Comprehension | no | yes | no | no | potential | no | potential |
| | ***Motivation Dependence*** | no | potential | potential | no | potential | no | no |
| | ***Modal Sensory Selection*** | no | no | no | no | potential | no | no |
| **Action Initiation and Control** | ***Inverse Modeling*** | no | no | no | no | yes | potential | no |
| | ***Action Control*** | yes | potential | potential | yes | yes | yes | yes |
| | *Faster Behavioral Adjustments* | no | potential | no | no | potential | no | no |
| **Action Decision Influences** | **Direct Influence** | | | | | | | |
| | predictive adjustments | no | potential | yes | yes | potential | no | no |
| | explicit planning | no | potential | yes | yes | potential | no | no |
| | **Via Motivations** | | | | | | | |
| | opportunistic behavior | no | potential | potential | potential | potential | no | no |
| | shape action preparation | no | potential | potential | no | potential | no | yes |
| | predispose action decision making | no | potential | potential | no | potential | no | yes |
| | **Unusual Information Content** | | | | | | | |
| | Curious behavior | no | potential | yes | no | potential | no | no |
| | Epistemic actions | no | potential | potential | no | potential | no | no |

**Table 2: Relevant systems classified according to their anticipatory capabilities and potentials.**

## 4.2 Contrasting Anticipatory System Capabilities

Table 2 shows the current anticipatory capabilities of the relevant anticipatory learning systems at hand. Hereby, we only list those systems that were directly targeted towards anticipatory capabilities. Clearly, the DYNA architecture is reflected in several other systems including XACS and the NN-based DYNA models. The artificial immune system architecture (AIS) has similar potentials. Inverse model-based systems as well as the inverse gradient method are additional mechanisms that may shape behavioral learning combining predictive learning capabilities and predictive structures with reinforcement learning capabilities. The context-based systems, then, provide a useful tool of how to include context information to guide and direct anticipatory mechanisms not only for attention processes but also for consequent action decision making and action control mechanisms.

Due to the focus of this workpackage on action decision making and control, we focus on these concerns in the remainder of this discussion. Before doing so, though, we want to point out that the model learning components themselves are not as much influenced by their own predictive capabilities as might be advantageous. Although most considered systems use error-based information, directing learning focus and delegating learning resources more effectively – potentially task-, motivation-, and emotion-dependent – appears to pose an interesting challenge to the project partners. Other predictive systems including LSTM and XCS may be able to support such capabilities more effectively. Research in this direction seems mandatory.

Besides the potential model learning improvements by the means of anticipatory mechanisms, the table shows that several other capabilities require future research. First, inverse modeling capabilities have not been investigated in considerable detail by any of the partners. The targeted adaptive robotic applications most probably require such control models so that more research effort in the MindRACES project needs to be directed in this direction. The inverse gradient approach converting reinforcement gradient into action gradients may help to improve learning of such inverse models.

Second, faster behavioral adjustments due to unexpected sensory inputs have not been investigated further so far. Kalman filtering-based updates and other error and information gain estimate methods will help to improve such control and stabilization capabilities.

Third, planning capabilities are especially directed, task-dependent planning mechanisms may be investigated further. The combination of differing predictive methods in order to enable prediction for action decision on multiple levels of abstraction seems inevitable. It also remains an interesting question how detailed planning has to be in order to be effective. Davidsson (1997, 2003) showed that linear predictions are often sufficient to improve behavior by inducing preventive mechanisms if the linear prediction (considering only what the agent would do usually) leads to undesired states.

Fourth, the coupling of motivational mechanisms and potentially even emotional mechanisms with the behavioral decision and control module poses additional challenges. Seeing that currently only the context systems architecture is able to direct attention and consequent action preparation and decision making mechanisms, the partners are encouraged to compare their system with the context processing mechanisms in Balkenius' work and evaluate how they can incorporate similar mechanisms into their own learning systems. Context as a special input may be coupled with motivations in that motivations may trigger certain motivation-relevant contexts. The activated contexts – activated, for example, in the form of an antibody in the AIS framework or a neural activity pattern in the hierarchical neural architecture – then should activate corresponding motor programs and action decisions that usually lead to the activated context.

Fifth, while curious behavior has been implemented in a few architectures, epistemic actions were not successfully shown in any architecture gathered in this deliverable. Epistemic actions may be realized in several systems, however, and pose an important challenge to the MindRACES project and workpackage four. The AIS system may exhibit curious behavior if antibodies in question can trigger additional information gathering by incurring action processing. The XACS architecture may be similarly enhanced if inputs are not available or have not enough information support – several enhancements and combinations with a more noise-robust learning mechanism will be necessary, though. Hierarchical NN-based system architectures may offer another solution for the incurrence of epistemic actions: Once higher levels are able to pre-activate lower level neurons, these pre-activations may not only lead to the faster detection of such consequent inputs but also to action activation mechanisms triggering curious actions. In general, while systems might have a general curious action selection mechanism, for example, for improving predictive model learning, epistemic actions might be based on the same principle of predicted information gain, only that in this case, the plasticity needs to be more dynamic in that the entropy of current important available information needs to be considered and selectively improved. This then will lead to truly current curious behavior and pro-activity.

### 4.3 Most Important Challenges Ahead

The contrasting factors show that the challenges related to workpackage four comprise improvements of the predictive capabilities of the systems as well of the anticipatory capabilities of the system.

The predictive capabilities are expected to be continued to be investigated and improved by the partners, separately for each of their systems but, more importantly, also in cooperation combining the strengths of several of the systems. The most important anticipated enhancements are (1) the development of predictive systems that process and combine different sources of information (such as context information, sensory information, and action information) maximally effectively, (2) the implementation of predictive hierarchies that can generate predictions at different levels of abstraction in time and space, and (3) to couple the predictive representations with an action control model. Especially the last point poses a great challenge to the MindRACES project but might be the key to the generation of actual cognitive systems in which perceptions are immediately linked with appropriate action codes (causing affordances and bottom-up action predispositions) and action codes, vice versa, are linked to corresponding sensory effect codes that are expected to change after action execution.

With such a cognitive model structure at hand, many anticipatory capabilities might even emerge naturally from the structure itself. However, even with a less sophisticated representation, several advanced anticipatory capabilities will need to be investigated. (1) Anticipatory shaping will be continued to be investigated by the partners and utilized to improve (the learning of) behavioral decision making and control. (2) The further development of curious behavior capabilities and epistemic action capabilities will be pursued further. Hereby, it is important that such behavior needs to be driven by anticipated information gain. Epistemic actions might be implemented as task-dependent curious behavior in that tasks require information that, if currently not available, triggers the epistemic actions that promise to lead (or rather previously led) to this information. (3) Anticipatory top-down mechanisms need to be further developed, that influence bottom-up sensory processing. This includes attentional mechanisms (investigated in further detail in workpackage three) but also action decision making and control mechanisms since action decision making can be considered as yet another attention process. (4) Finally, a motivational and potentially emotional module may be coupled with the predictive system in order to induce even better action decision making capabilities enabling learning and execution of opportunistic actions as well as anticipatory actions that satisfy expected motivations (such as taking food and water on a hike).

The anticipatory enhancements are certainly not stand-alone but are very interdependent on each other as well as on the underlying predictive representation used. Certainly, further investigations in the relevant anticipatory capabilities will again point out advantageous predictive structures, such as the hierarchical structures that we expect to be very suitable for the generation of effective anticipatory behavior. Thus, the discussed enhancements of the predictive capabilities of the system should not (only) be pursued in isolation but rather should be targeted from the beginning on the anticipatory mechanisms that we intend to implement using the developed predictive representation. Certainly, interactive, emergent, unexpected properties might develop along the way of this research endeavor and might as well lead to novel insights in information processing, adaptive behavior, embodiment, and cognition as a whole.

## 5 Conclusions

This document has shown that the challenges ahead are broad and not always straight forward. In order to create competent anticipatory, cognitive embodied systems, the systems do not only need to be competent in learning an effective predictive model of their environment but also need to be able to effectively exploit the learned model for adaptive behavior. This process is expected to be interactive rather than iterative in that the developing predictive capabilities should immediately cause anticipatory mechanisms that, vice versa, immediately influence the further development of the predictive capabilities. The expertise of the partners is sufficiently broad to cover all areas of interest but further effort is necessary to actually combine the different bits of knowledge of each partner effectively to create the desired cognitive systems. It is hoped that this document does not only show that the efforts in workpackage four and the MindRACES project as a whole proceed in the right direction, but also that the partners are encouraged to further consider cooperation amongst them (1) to assess the potentials of each relevant learning systems with respect to the predictive and anticipatory challenges ahead and (2) to create combinations of their systems to be able to tackle the challenges most effectively. The challenges and potentially highly rewarding tasks ahead in the MindRACES promise to lead to new inventions of competent anticipatory behavior-based robotic systems as well as to a better understanding of adaptive behavior and even cognitive processes in general. The categorizations and contrasting discussions in this document can serve as the guidelines to develop such competent anticipatory cognitive embodied mechanisms and systems.

**References:**

Arbib, M. A. (1995) Schema theory, in M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*. MIT Press.

Arkin, R. C. (1998). *Behavior-Based Robotics*, MIT Press.

Baldassarre G. (2003). Forward and bidirectional planning based on reinforcement learning and neural networks in a simulated robot. In Butz M., Sigaud O., Gérard P. (eds.), *Anticipatory Behavior in Adaptive Learning Systems*, pp. 179-200. Berlin: Springer Verlag.

Baldassarre G. (2001). Coarse planning for landmark navigation in a neural-network reinforcement learning robot. *Proceedings of the International Conference on Intelligent Robots and Systems* (IROS-2001). IEEE.

Balkenius, C. (1998). Spatial learning with perceptually grounded representations. *Robotics and Autonomous Systems*, 25, 165-175

Balkenius, C. (2000). Attention, habituation and conditioning: toward a computational model. *Cognitive Science Quarterly*, 1, 2, 171-214.

Balkenius, C. (2003). Cognitive processes in contextual cueing. In Schmalhofer, F., Young, R. M., and Katz, G. (Eds.), *Proceedings of the European Cognitive Science Conference 2003* (pp. 43-47). Mahwah, NJ: Lawrence Erlbaum Associates.

Balkenius, C., & Winberg, S. (2004). Cognitive modeling with context sensitive reinforcement learning. In *Proceedings of AILS '04*. Lund: Dept. of Computer Science.

Balkenius, C., & Björne, P. (2001). Toward a robot model of attention-deficit hyperactivity disorder (ADHD). In Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., and Breazeal, C. (Eds.), *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems.* Lund University Cognitive Studies, 85.

Balkenius, C., & Morén, J. (2000). A computational model of context processing. In Meyer, J-A., Berthoz, A., Floreano, D., Roitblat, H. L., Wilson, S. W. (Eds.), *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior.* Cambridge, MA: MIT Press.

Balkenius, C., & Morén, J. (2000). Emotional learning: a computational model of the amygdala. *Cybernetics and Systems*, 32, 6, 611-636.

Balkenius, C., & Hulth, N. (1999). Attention as selection-for-action: a scheme for active perception. In Schweitzer, G., Burgard, W., Nehmzow, U., and Vestli, S. J. (Eds.), *Proceedings of EUROBOT '99* (pp. 113-119). IEEE.

Balkenius, C., & Johansson, B. (2004). The Coding of Context in Visual Scenes. unpublished manuscript.Bandura, A. (1999). A social cognitive theory of personality. In L. Pervin & O. John (Ed.), Handbook of personality (2nd ed., pp. 154-196). New York: Guilford Publications. (Reprinted in D. Cervone & Y. Shoda [Eds.], *The coherence of personality*. New York: Guilford Press.)

Balkenius, C., & Morén, J. (2000). A computational model of context processing. In Meyer, J-A., Berthoz, A., Floreano, D., Roitblat, H. L., Wilson, S. W. (Eds.), *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.

Bandura, A. (1999). A social cognitive theory of personality. In L. Pervin & O. John (Ed.), *Handbook of personality* (2nd ed., pp. 154-196). New York: Guilford Publications. (Reprinted in D. Cervone & Y. Shoda [Eds.], *The coherence of personality*. New York: Guilford Press.)

Barnsley M. F. (1998). *Fractals everywhere*. Academic Press, New York.

Barsalou, L. W. (1999). *Perceptual symbol systems*. Behavioral and Brain Sciences, 22, 577-600.

Bernado-Mansilla, E., & Garrell-Guiu, J. M. (2003). Accuracy-based learning classifier systems: Models, analysis, and applications to classification tasks. *Evolutionary Computation*, 11, 209-238.

Berthier, N. E., Singh, S. P., & Barto, A. G. (1992). A cortico-cerebellar model that learns to generate distributed motor commands to control a kinematic arm. *Advances in Neural Information Processing Systems*, 4, 611-618.

Björne, P., & Balkenius, C. (2005). The role of context and extinction in ADHD. *Behavioral and Brain Sciences,* in press.

Boden, M. (2004). Generalization by symbolic abstraction in cascaded recurrent networks. Neurocomputing, 2004.

Brooks, R.A., "How to build complete creatures rather than isolated cognitive simulators," in K. VanLehn (ed.), *Architectures for Intelligence*, pp. 225-239, Lawrence Erlbaum Assosiates, Hillsdale, NJ, 1991.

Bühlmann P. (1988). Extreme events from return-volume process: A discretization approach for complexity reduction. *Applied Financial Economics*, 8, 267-278.

Butz, M. V. (2002). *Anticipatory learning classifier systems*. Boston, MA: Kluwer Academic Publishers.

Butz, M. V., & Goldberg, D. E. (2003). Generalized State Values in an Anticipatory Learning Classifier System In Butz, M. V., Sigaud, O.,& Gérard, P. (Eds.), *Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems (LNCS 2684)*, 282-301, Springer Verlag, Berlin.

Butz, M. V. (2004a). *Rule-based Evolutionary Online Learning Systems: Learning Bounds, Classification, and Prediction*. PhD thesis. University of Illinois at Urbana-Champaign, Department of Computer Science, IL, USA.

Butz, M. V. (2004b). COSEL: A cognitive sequence learning architecture (*IlliGAL report 2004021*). University of Illinois at Urbana-Champaign: Illinois Genetic Algorithms Laboratory.

Butz, M. V. (in press). Kernel-based, ellipsoidal conditions in the real-valued XCS classifier system. *Proceedings of the Seventh Genetic and Evolutionary Computation Conference (GECCO-2005).*

Butz, M. V., Goldberg, D. E., & Lanzi, P. L. (in press). Gradient Descent Methods in Learning Classifier Systems: Improving XCS Performance in Multistep Problems. *IEEE Transactions on Evolutionary Computation.*

Butz, M. V., Goldberg, D. E., & Tharakunnel, K. (2003). Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation*, 11, 239-277.

Butz, M. V., & Hoffmann, J. (2002). Anticipations control behavior: Animal behavior in an anticipatory learning classifier system. *Adaptive Behavior*, 10, 75-96.

Butz, M. V., Kovacs, T., Lanzi, P. L., & Wilson, S. W. (2004). Toward a theory of generalization and learning in XCS. *IEEE Transactions on Evolutionary Computation*, 8, 28-46.

Butz, M. V., Goldberg, D. E., & Lanzi, P. L. (in press). Gradient Descent Methods in Learning Classifier Systems: Improving XCS Performance in Multistep Problems. *IEEE Transactions on Evolutionary Computation.*

Butz, M. V., Swarup, S., & Goldberg, D. E. (2004) Effective Landmark Detection. ICML 2004, *Workshop on Predictive Representations of World Knowledge*. Online proceedings.

Brillinger D.R. (1994). Examples of scientific problems and data analysis in demography, neurophysiology and seismology. *J. Comp. and Graph. Statistics*, 3, 1-22.

Castelfranchi, C., Falcone, R., (1998a) Towards a Theory of Delegation for Agent-based Systems, *Robotics and Autonomous Systems*, Special issue on Multi-Agent Rationality, Elsevier Editor, Vol 24, Nos 3-4, pp.141-157.

Castelfranchi C., Falcone R., (1998b) Principles of trust for MAS: cognitive anatomy, social importance, and quantification, *Proceedings of the International Conference on Multi-Agent Systems (ICMAS'98)*, Paris, July, pp.72-79.

Castelfranchi, C., Lorini, E. (2003). Cognitive Anatomy and Functions of Expectations. In *Proceedings of IJCAI'03 Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, Acapulco, Mexico, August 9-11.

Chun, M.M., & Jiang, Y. (1998) Contextual cueing: implicit learning and memory of visual context guides spatial attention. *Cognitive Psychology* 36, 28–71

Chun, M. M. (2000). Contextual cueing of visual attention. *Trends in Cognitive Science*, 4 (5), 170-178.

Cohen, P. R. , Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42, pp. 213-261.

Corrêa, M., Coelho, E. (1998). Agent´s programming from a mental states framework. In *Proceedings of the 14th Brazilian Symposium on Artificial Intelligence (SBIA98),* pp. 31-39.

Coulson S., King J.W., & Kutas M. (1998). Expect the unexpected: Responses to morphosyntactic violations. *Language and Cognitive Processes*, 13(1).

Cruse, H., Steinkühler, U., & Burkamp, C. (1998). MMC – a recurrent neural network which can be used as manipulable body model. *From Animals to Animats 5 (SAB 1998),* 381-389.

Davidsson, P. (1997). Learning by linear anticipation in multi-agent systems. In Weiss, G. (Ed.), *Distributed artificial intelligence meets machine learning* (pp. 62-72). Berlin Heidelberg: Springer-Verlag.

Davidsson, P. (2003). A framework for preventive state anticipation. in Butz, M. V., Sigaud, O., & Gérard, P. (Eds.) *Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems*. Springer-Verlag, Berlin-Heidelberg, 151-166.

Drescher, G. (1991). *Made-up minds: A constructivist approach to artificial intelligence*. MIT Press.

Falcone R., Castelfranchi C., (2001). Social Trust: A Cognitive Approach, in *Trust and Deception in Virtual Societies* by Castelfranchi C. and Yao-Hua Tan (eds), Kluwer Academic Publishers, pp. 55-90.

Farmer, J. D, Packard H. N, Perelson A. H. The Immune system, adaption and machine learning. *Physica D,* 22:187-204, 1986.

Gers, F. A. and Schmidhuber, J. and Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10): 2451—2471.

Gallese, V. Fadiga, L., Fogassi, L. & Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119: 593-609.

Gallese, V. & Goldman, A. (1998). Mirror neurons and the simulation theory of mind reading. *Trends in Cognitive Psychology*, 2, 493-501.

Gibson, J. J. (1979). *The ecological approach to visual perception*.Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Giese, M. A., & Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4, 179-192.

Gloye, A., Goektekin, C., Egorova, A., Tenchio, O., & Rojas, R. (2004).FU-Fighters small size 2004. *RoboCup-2004: Robot Soccer World Cup VIII*, Springer-Verlag.

Graves, A., Eck, D., Beringer, N., & Schmidhuber, J. (2004). Biologically Plausible Speech Recognition with LSTM Neural Nets. In J. Ijspeert (Ed.), *First Intl. Workshop on Biologically Inspired Approaches to Advanced Information Technology, Bio-ADIT 2004*, Lausanne, Switzerland, p. 175-184, 2004.

Grimes, D. B., & Rao, R. P. N. (2005). Bilinear sparse coding for invariant vision. *Neural Computation*, 17, 47-73.

Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27 (3): 377-396.

Hart E., Ross P., Webb A., & Lawson A. (2003). A Role for Immunology in Next Generation Robot Controllers. *In Proceedings of ICARIS 2003*. Springer Verlag Heidelberg.

Haruno, M., Wolpert, D. M., & Kawato, M. (2001). MOSAIC model for sensorimotor learning and control. *Neural Computation*, 13, 2201-2220.

Hirayama, M., Kawato, M., & Jordan, M. I. (1993). The cascade neural network model and a speed accuracy trade-off of arm movement. *Journal of Motor Behavior, 25*(3), 162-174.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8):1735-1780.

Hoffmann, J. (1993). Vorhersage und Erkenntnis, Göttingen, Germany: Hogrefe.

Hoffmann, J. (2003). Anticipatory behavioral control. In Butz, M. V., Sigaud, O., & Gérard, P. (Eds.), *Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems* (pp. 44-65). Berlin Heidelberg: Springer-Verlag.

Holmes, M., & Isbell, C. (2005). Schema Learning: Experience-Based Construction of Predictive Action Models *NIPS* 17.

Jerne N.K. (1974). Towards a network theory of the immune system. *Ann. Immunol (Inst. Pasteur)* 125C. (p.373-389).

Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16, 307-354.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering,* 82, 35 -45.

Kalveram, K. T. (2004). The inverse problem in cognitve, perceptual and proprioceptive control of sensorimotor behavior: Towards a biologically plausible model of the control of aiming movements. *International Journal of Sport and Exercise*, 2, 255-273.

Kalveram, K. T., Schinauer, T., Beirle, S., Richter, S., & Jansen Osmann, P. (2005). Threading neural feedforward into a mechanical spring - how biology exploits physics in limb control. *Biological Cybernetics*, 92, 229-240.

Kant, Immanuel (1781/1998). Critique of Pure Reason. Cambridge University Press.

Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57, 169-185.

Kersting, K., Van Otterlo, M., & De Raedt, L. (2004). Bellman goes relational. *Machine Learning, Proceedings of the Twenty-first International Conference* (ICML 2004), ACM.

Kinny, D. N., Georgeff, M. P. (1991). Commitment and Effectiveness of Situated Agents. In *Proceeding of IJCAI'91*, p82-88.

Kokinov, B. (1988). Associative Memory-Based Reasoning: How to Represent and Retrieve Cases. In: T.O'Shea & V.Sgurev (eds.) *Artificial Intelligence III: Methodology, Systems, Applications*, Elsevier Science Publ.

Kokinov, B. (1994). A Hybrid Model of Reasoning by Analogy. Chapter 5. in: K. Holyoak & J. Barnden (eds.) *Analogical Connections, Advances in Connectionist and Neural Computation Theory*, vol.2, Ablex Publ. Corp.

Kokinov, B. (1994). The DUAL Cognitive Architecture: A Hybrid Multi-Agent Approach. In: A. Cohn (ed.) *Proceedings of ECAI'94*. John Wiley & Sons, Ltd., London.

Kokinov, B. (1994). The Context-Sensitive Cognitive Architecture DUAL. In: *Proceedings of the 16th Annual Conference of the Cognitive Science Society*. Erlbaum, Hillsdale, NJ.

Kokinov, B. (1997). Micro-Level Hybridization in the Cognitive Architecture DUAL. In: R. Sun & F. Alexander (eds.) *Connectionist-Symbolic Integration: From Unified to Hybrid Architectures,* Lawrence Erlbaum Associates, Hilsdale, NJ, (First published in: Sun, R. & Alexander, F. (Eds.) *Proceedings of the IJCAI-95 Workshop on Connectionist-Symbolic Integration,* 1995.)

Kokinov, B. (1998). Analogy is like Cognition: Dynamic, Emergent, and Context-Sensitive. In: Holyoak, K., Gentner, D., Kokinov, B. (Eds.) – Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences. Sofia: NBU Press.

Kokinov, B., Petrov, A. (2001). Integration of Memory and Reasoning in Analogy-Making: The AMBR Model. In: Gentner, D., Holyoak, K., Kokinov, B. (Eds.) *The Analogical Mind: Perspectives from Cognitive Science*, Cambridge, MA: MIT Press.

Kokinov, B., Grinberg, M. (2001). Simulating Context Effects in Problem Solving with AMBR. In: Akman, V., Thomason, R., Bouquet, P. (Eds.) Modeling and Using Context. *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, vol. 1775, Springer Verlag.

Lin L.J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8, 293-391.

Littman, M. L., Sutton, R. S., & Singh, S. (2002). Predictive representations of state. In *Advances in Neural Information Processing Systems*, pages 1555–1561.

Loeb, G. E., Brown, I. & Cheng, E. (1999). A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research*, 126, 1-18.

Maes, P. (1990), Situated Agents Can Have Goals, *Robotics and Autonomous Systems*, 6:49-70.

McCallum, K. (1996). *Reinforcement learning with selective perception and hidden state*. Ph.D. Thesis.

McFarland D., & Bosser, T. (1994). Intelligent Behavior in Animals and Robots. MIT Press, Cambridge MA.

Meyer, J.-J. Ch., van der Hoek, W., van Linder, B. (1999). A Logical Approach to the Dynamics of Commitments. *Artificial Intelligence*, 113(1-2), pp. 1-40.

Miall, R. C., Weir, D. J., Wolpert, D. M., & Stein, J. F. (1993). Is the cerebellum a smith predictor? *Journal of Motor Behavior*, 25(3), 203-216.

Miceli, M, Castelfranchi, C. (2002). The Mind and the Future. The (Negative) Power of Expectations. *Theory & Psychology*, 12(3), pp. 335-366.

Michelan, R., & von Zuben, F., J. (2002). Decentralized control system for autonomous navigation based on an evolved artificial immune system. In C.L. Giles et al, editor, *Proceedings of CEC-02* (p.1021-1026). IEEE Press.

Minsky, M. (1975). A framework for representing knowledge. In P. Winston, editor, The Psychology of Computer Vision, pages 211–277. McGraw-Hill, New York.

Moore, A. W., & Atkeson, C. (1993). Prioritized sweeping: reinforcement learning with less data and less real time. *Machine Learning*, 13, 103-130.

Morén, J. (2003). Emotion and Learning. *PhD Thesis*. Lund University Cognitive Studies 93.

Mowrer, O. H. (1960/1973). Learning theory and behavior. New York: Wiley.

Parfitt S., Tino P., & Dorffner G. (1999). Graded grammaticality in prediction fractal machines. *NIPS* 1999: 52-58.

Petrov, A., Kokinov, B. (1999). Processing Symbols at Variable Speed in DUAL: Connectionist Activation as Power Supply. In: Dean, T. (ed.) Proceedings of the 16th International Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufman, p. 846-851

Piaget, J. (1954). *The Construction of Reality in the Child*. Ballentine.

Piaget, J. (1970). *La Formation du Symbole Chez L'Enfant*. Delachaut et Nestlie.

Peirce, C. S. (1897/1949). Logic as semiotic: The theory of signs. In Philosophical writings of Peirce. Dover.

Poggio, T., & Bizzi, E. (2004). Generalization in vivion and motor control. *Nature*, 431, 768-774.

Pomerleau, D. (1989). "ALVINN: An Autonomous Land Vehicle In a Neural Network," Advances in Neural Information Processing Systems 1, Morgan Kaufmann.

Powers, W. T. (1973). *Behavior: The Control of Perception*. Hawthorne, NY: Aldine de Gruyter.

Prum, B., Rodolphe, F., & de Turckheim, E. (1995). Finding words with unexpected frequencies in deoxyribonucleic acid sequences. *Journal of Royal Statistical Society*, B 57, 205-220.

Rao, A. S., Georgeff, M. P.(1991a). Deliberation and Intentions. In *Proceedings of 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles.

Rao, R. P. N., & Ballard, D. H. (1997). Dynamic Model of Visual Recognition Predicts Neural Response Properties in the Visual Cortex. *Neural Computation*. 9(2).

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2, 79-87.

Rattenberger J., Poelz P.M. and Prem E. (2004). Artificial Immune Networks for Robot Control. In *Proceedings of Epigenetic Robotics 2004*.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortes. *Nature Neuroscience, 2*(11), 1019-1025.

Rizzolatti, G., Fadiga, L, Gallese, V., & Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. Cognitive Brain Research, 3: 131-141.

Rosenblueth, A., Wiener, N., Bigelow, J. (1960). Behavior, Purpose, and Teleology. In W. Buckley (Ed.), *Modern Systems Research for the Behavioral Scientist*, Aldine, Chicago.

Roy, D., Hsiao, K.-Y., & Mavridis, N. (2004). Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34: 1374-1383.

Roy, D. (submitted). Grounding Language in the World: Schema Theory Meets Semiotics. Submitted to Artificial Intelligence.

Salinas, E., & Abbott, L. F. (1995). Transfer of coded information from sensory to motor networks. *Journal of Neuroscience, 15*(10), 6461-6474.

Schank, R., & Abelson, R. (1977). *Scripts, Plans, Goals and Understanding*. Erlbaum, 1977.

Scherl, R. B., Levesque, H. (2003). Knowledge, action, and the frame problem. *Artificial Intelligence*, 144, pp. 1-39.

Schmidhuber, J., & Huber, R. (1991a). Learning to generate artificial fovea trajectories for target detection. *International Journal of Neural Systems*, 2(1 & 2):135-141.

Schmidhuber, J., & Huber, R. (1991b). Using sequential adaptive neuro-control for efficient learning of rotation and translation invariance. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, Artificial Neural Networks, pages 315-320. Elsevier Science Publishers B.V., North-Holland.

Schmidhuber, J. (1990a). Learning algorithms for networks with internal and external feedback. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, Proc. of the 1990 Connectionist Models Summer School, pages 52-61. San Mateo, CA: Morgan Kaufmann.

Schmidhuber, J. (1990b). An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In Proc. IEEE/INNS International Joint Conference on Neural Networks, San Diego, volume 2, pages 253-258.

Schmidhuber, J. (1990c). Reinforcement learning with interacting continually running fully recurrent networks. In Proc. INNC International Neural Network Conference, Paris, volume 2, pages 817-820.

Schmidhuber, J. (1990d). Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical Report FKI-126-90, Institut fuer Informatik, Technische Universitaet Muenchen, February (revised in November).

Schmidhuber, J. (1991a). Curious model-building control systems. In *Proc. International Joint Conference on Neural Networks*, Singapore, volume 2, pages 1458-1463. IEEE.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats, pages 222-227. MIT Press/Bradford Books.

Schmidhuber, J. (1991c). Reinforcement learning in Markovian and non-Markovian environments. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, Advances in Neural Information Processing Systems 3, NIPS'3, pages 500-506. San Mateo, CA: Morgan Kaufmann.

Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2): 234-242.

Schut, M. C., Wooldridge, M. J., Parsons S. D. (2004). The Theory and Practice of Intention Reconsideration. *Journal of Experimental and Theoretical Artificial Intelligence*, 16(4), pp. 261-293.

Shanahan, M. P., Randell, D. (2004). A Logic-Based Formulation of Active Visual Perception. In *Proceedings of KR 2004*, pp. 64-72.

Shanahan, M. P. (2005). Perception as Abduction: Turning Sensor Data into Meaningful Representation. *Cognitive Science*, 29, pp. 103-134.

Singh, M. P., Asher, N. M. (1993). A Logic of Intentions and Beliefs. *Journal of Philosophical Logic*, 22, pp. 513-544.

Singh, S., Barto, A. G., & Chentanez, N. (2005). Intrinsically Motivated Reinforcement Learning. *Proceedings of Advances in Neural Information Processing Systems (NIPS)* 17.

Simon, H. A.(1969). *Sciences of the Artificial*. Cambridge, MA: MIT Press.

Simsek, Ö., & Barto, Andrew G. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. *Proceedings of the Twenty-First International Conference on Machine Learning*, 751-758.

Stroeve, S. (1996). Linear combined feedback and feedforward control of a musculoskeletal system. *Biological Cybernetics*, 75, 73-83.

Stroeve, S. (1997). A learning feedback and feedforward neuromuscular control model for two degrees of freedom human arm movements. *Human Movement Science*, 16, 621-651.

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

Sutton, R. S. (1991). DYNA, an integrated architecture for learning, planning, and reacting. In Working Notes of the 1991 AAAI Symposium, pp. 151- 155. AAAI Press.

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

Tino P., & Doffner G. (1998). Recurrent neural networks with iterated function systems dynamics. In *International ICSC/IFAC Symposium on Neural Computation* (pp. 526-532). Canada/Switzerland: ISCSC Academic Press.

Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, *5(*11), 1226-1235.

Urban, J.-P., Bruessler, J. L., & Gresser J. (1998). Neural Networks for Visual Servoing in Robotics,. In L. C. Jain & T. Fukuda (Eds.): *Soft computing for intelligent robotic systems*. Berlin, Germany: Springer-Verlag.

Vu Trân, B., Harland, J., Hamilton, M. (2004). Observation Expectation Reasoning in Agent Systems. In *Proceedings of KR 2004*, pp. 246-254.

Watanabe, Y, Ishiguro, A., Shiari, Y., & Uchikawa, Y. (1998). Emergent construction of behavior-arbitration mechanism based on the immune system. In *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, (pp. 481-486). Los Alamitos, CA: IEEE Computer Society Press.

Webb A., Hart E., Ross P., and Lawson A. (2003). Controlling a simulated Khepera with an XCS classifier system. In Proceedings of 7th European Conference on A-Life. Springer-Verlag.

Wilson, S. W. (1995). Classifier fitness based on accuracy. Evolutionary Computation, 3 (2), 149-175.

Wilson, S. W. (2002). Classifiers that approximate functions. Natural Computing, 1, 211-234.

Wolpert, D. M. & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks, 11*(7-8), 1317–1329.