PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/s13230-010-0010-4 · JBR · 1(2) · 2010 · 116-129

robo-CAMAL: A BDI Motivational Robot

Darryl Davis[1]*, James Gwatkin[2]†

1 Department of Computer Science, University of Hull, Cottingham Road, Hull HU6 7RX, U.K.

2 Portaltech, Isis House, 67-69 Southwalk Street, London SE1 0HX, U.K.

Received 6 September 2009

Accepted 17 May 2010

Abstract

Motivation is a central concept in the development of autonomous agents and robots. This paper describes an architecture that uses a psychological BDI model of reasoning, combined with a distributed multi-level model of motivation. The robot controlling architecture makes use of a generic set of deliberative components plus an environment task-centred set of reactive components that reflect the architecture's embodiment. The architecture has been used in a number of simulated environments and here is used to control a mobile robot. A theoretical framework for motivation and affect is given, and related to the nature of autonomy and embodiment. A BDI model, based on a psychological model of reasoning in a 5 year old child, is described in terms of the nature of motivation and affect within the architecture. Finally, criteria for judging the nature of an agent's motivation are introduced, and used to validate the motivational constructs implemented within the architecture. Experimental results lead to a comparative discussion.

Keywords

cognitive robot · motivation · anchoring · BDI · learning

1. Introduction

Mobile robots provide an essential tool when investigating the interaction of cognitive architectures and the physical environment. Robots have been used to investigate many different aspects of artificial intelligence such as mapping and localization techniques [1, 2], robot perception [3] and robot learning [4, 5]. This research described here seeks to use a mobile robot to investigate a specific area of cognitive science known as the anchoring problem. The anchoring problem describes the problem of generating and maintaining links between symbols and perceptual data.

The high level aspects of the cognitive architecture have been developed over a number of related projects using simulation environments [6–9]. This paper describes how we anchor this cognitive architecture using an embodied agent (or robot). This research attempts to achieve two main goals. The first is to develop a robotic agent that can learn how to achieve its goals with no prior understanding of the effect of its actions. To do this the agent must be able to identify the focus of its goal, i.e. if the goal is to touch a specific object, it must recognise when that object is close. The agent must also be able to recognise when the objective of its goal has been achieved. This is akin to dynamically creating plans for behaviours to achieve successful goal completion [10].

To achieve these goals a hybrid reactive-deliberative architecture has been implemented upon a mobile robot [11]. Hybrid architectures seek to avoid the disadvantages of their component architectures, whilst retaining all their benefits. A common hybrid architecture is the reactive-deliberative architecture [12]. Here we present an architecture consisting of several different elements including a set of low-level robot actions; a reactive component built from many different reactive behaviours; a belief-desire-intention (BDI) schema; a distributed model of affect; an association construct; a domain model; and a motivational blackboard that links all these subsystems together. All these components implemented on a robotic platform combine to make robo-CAMAL.

Motivation can be considered as the driving force behind all the actions of an agent [13, 14]. Motivation cannot be observed directly, but it can be inferred from the observable behaviour of an agent [15]. If a robotic agent is to act of its own volition, then it requires some form of motivation to be incorporated within its architecture. Other researchers ([5, 16, 17]) have employed such concepts with promising results. This paper will discuss the nature of the motivational constructs used by the robo-CAMAL architecture, and show how motivation is grounded within its environment. Summary results from extensive testing and experimentation show how effective these mechanisms are for controlling an autonomous robot in a structured dynamic environment. A discussion of the relative merits of the current approach is given, before highlighting current and future directions in the concluding remarks.

2. Research issues

This section will consider some of the issues that need to be addressed in answering the questions under investigation. robo-CAMAL is one instantiation of the range of cognitive architectures developed within the CAMAL research, and the first to couple the cognitive architecture with a physical robot. CAMAL is a cognitive architecture that attempts to provide a positive answer to the two following questions.

The first is can the robotic agent developed here modify its goals and behaviour in response to changes, outside of its control, in a dynamic physical real-time environment.

The second question is can the agent learn which actions to use to achieve its goals. One research question that this and ongoing projects are addressing is how can architecture learn the optimal configuration of these various sub-systems through a combination of adapta-

*E-mail: D.N.Davis@dcs.hull.ac.uk
†E-mail: james.gwatkin@portaltech.co.uk

tion, learning and (meta-) reasoning. The agent has a set of beliefs it can hold, actions it can perform, and goals to achieve. To achieve its goal the agent needs to instigate the correct action based on its current belief. Given that the agent is given no explicit knowledge of the correct belief-goal-action combination (or association), can it determine the correct combination on its own? A related point is the case where the agent is provided with several possible belief-goal-action combinations that achieve a specific goal. Can the agent determine which is the best combination, and then modify its preferred selection as the environment changes? For a robotic agent to provide a positive answer to these questions its architecture needs to address several key issues within cognitive science and robotics.

2.1. The anchoring problem

The symbol grounding problem concerns the difficulties of generating symbols using perceptual systems, and the meaning of those symbols [18]. The anchoring problem is a subset of the grounding problem. It investigates how links are generated and maintained between symbols used within an agent's cognitive architecture, and the data obtained via the agent's perceptual system [19]. In the past within robotics, this linking of symbols to perceptual data has been buried in the code of the agent's architecture.

Recently there has been a push to formalise the problem in order to identify the difficulties in linking symbols to objects, and to either separate the anchoring process from the rest of the architecture, or to specify when and where anchoring occurs. This approach provides insight into the specific problems that surround the anchoring process. If an agent is to reason about symbolic representations of its environment, it must be able to perceive its environment. It must also be able to link those perceptions to the relevant symbols.

2.2. Situated and embodied cognition

Situated and embodied cognition refers to the role the environment plays in the development of cognitive processes within the agent [20–22]. The cognitive processes of an agent that is situated (i.e. is present within its environment) are determined to a large extent by its environment.

The cognitive processes of an agent that is embodied (has a physical body within its environment) are determined by its interactions with its environment. In other words, cognitive processes develop from real-time, goal-directed interactions between the agent and its environment [23]. From this viewpoint the agent can learn to achieve its goals by interacting with its environment. If this is the case then information about its environment and its physical body must be available to the agent. Furthermore for such an agent to be considered autonomous, it must have the means to select and achieve actions in motivated behaviour [17].

2.3. Machine learning

One of the questions posed earlier, is whether the agent can learn the correct behaviour required to achieve its goal. It is therefore clear that the agent needs some form of learning mechanism [10, 24]. There are various different mechanisms possible but the one implemented here uses a simplified reinforcement learning technique [25]. This is where an agent learns by interacting with, and receiving feedback from, its environment.

3. CAMAL: A Cognitive Architecture for Motivation, Affect and Learning

This section will briefly introduce some of the main components of the generic CAMAL architecture, of which robo-CAMAL is a variant.

3.1. Motivation in CAMAL

Mind can be viewed as an organised collection of cognitive processes. These processes are integrated in a way that enables an agent to decide its next action. One approach that takes this view is the use of mind as a control system [4]. This takes the approach that mind is a collection of many different control processes passing data between them asynchronously.
The use of control states within the CAMAL architectures leads to the use of motivational control states. Figure 1 shows the five (top-level) motivational control states that have been used in implementations to date [7, 8]. Further sub-types have been investigated in other research [9]. To a certain extent, motivational states to the left of this figure can be subsumed within those to the right. For example, the intention to combine certain behaviours and plans may subsume the desire to avoid-collisions, which subsumes the qualitative goal no(collision), which makes use of adaptive thresholds (managed as quantitative goals) related to instincts and reflexes to perform specific micro-behaviours (such as stop, turn-left etc.). Drives are low level mechanisms and refer to the same types of systems as seen from the behaviourist or reactive perspective of motivation. The onset of a drive is dependent on variables that fluctuate in response to internal and external processes. If the variable crosses a specific threshold then the drive activates a pre-set behaviour or response. Instincts and reflexes are highly constrained drives. Instincts and drives are present in CAMAL at the reactive level. As the reactive aspects of CAMAL architectures are dependent upon the application, these are typically related to specific embodiments and environments. For example, robo-CAMAL has a number of low-level micro-behaviours (such as "turn right", "stop", "turn left" etc.). Each micro-behaviour can be considered a reflex. For example the reactive control rule

$$\text{IF left sonar value} < \text{threshold THEN turn right}$$

is a pre-set response to a sensor variable. The (reactive) threshold can be varied according to deliberative goals, or environment. The forming of optimal combinations of micro-behaviours to create a macro-behaviour can be also considered an agent's drive.
Goals come in two types, quantitative and qualitative. Quantitative goals are the same as goals used within control theory [26]. Within control theory the system has a specific output that it needs to achieve or maintain. The system uses feedback from its environment to modify its actions to achieve or maintain that state. Quantitative goals are present within CAMAL at the reactive level and the deliberative-reactive interface. For example, in some simulation experiments we have investigated artificial physiologies and metabolisms [9]. We could design robo-CAMAL to track other (dynamic) objects using goals that state the preferred (minimum and maximum) range and tie suitable reactive behaviours direct to the perceptual systems. Such a behaviour would then return control to the deliberative system only on failing to meet this quantitative goal or after performing the goal for a set time limit.
Qualitative goals describe some desired end state for an agent; for example, when an agent searches through problem space to find a solution or goal state. Qualitative goals are present in CAMAL at the deliberative level and take the form hit a ball (in five-aside football) or avoid an object (in general navigation).However with the adoption of

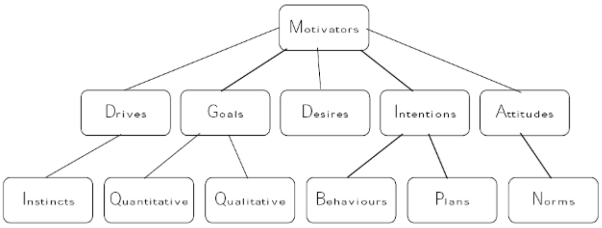

Figure 1. Five major Motivational Control States with (non-exhaustive set of) subtypes.

BDI reasoning model, it is more appropriate to think of these goals as propositions describing the end states associated with desires.

Desires are symbolic statements that define a specific preferred environmental state. Desires here are the same as those used within a BDI schema. Desires within CAMAL describe the specific goal, the belief required for the goals success, and the desires importance. Desires are present at the deliberative level within CAMAL. These take the form

goal(Desire, SuccessCondition, GoalImportance, ThreatValue
goal(Desire, SuccessCondition, GoalImportance, ThreatValue)

Intentions are also the same as those used within a BDI schema. They are strategies, plans and behaviours that are used to achieve desires. Intentions are found within CAMAL at the deliberative level. They take the form of predicates detailing the various possible reactive architectures, or calls to planners or other reasoning sub-systems.

Finally attitudes are pre-dispositions to respond in certain ways to certain perceptual or internal triggers. For example consider an agent developed to play five-a-side football. The agent may choose to attack or defend. This can depend on team orders or the specific environmental situation [27]. These attitudes affect which goals are chosen. If the attitude is to attack then the goal may be to hit the ball. If the attitude is to defend then the goal may be to get between the ball and the scoring zone. Attitudes are present within robo-CAMAL, but are pre-programmed prior to run time. An attitude in robo-CAMAL refers to the pre-defined goal set. Different attitudes or goal sets must be changed by the user off-line. Other research has looked at the use of meta-cognition to dynamically change attitudes using collections of Norms [9].

In summary motivational control states are distributed throughout the CAMAL architecture. Instincts, desires, and quantitative goals are present at the reactive level. Qualitative goals, desires, intentions, and attitudes are all managed at the deliberative level using a motivational blackboard system.

3.2. Reasoning using a Cognitive BDI Model

The belief-desire-intention (BDI) model [28] is a schema that calculates the actions of an agent based on its beliefs and its desires. A belief is a statement about the confidence of a proposition. The confidence the agent can have in a belief can vary. In the BDI model beliefs are based on input from the agent's perceptual system, and its previously held beliefs. The agent's desires are a set of goals which the agent wishes to achieve. The agent's current desires are based on its internal state, possibly its emotional state, and its previously held desires. Coupling the agent's beliefs and its desires generate a set of intentions or plans to achieve its goals. For example the agent has a goal to hit a ball. Its perceptual system generates the belief that there is a ball to the right. The agent can implement a set of plans to turn the agent right and move forward.

The CRIBB (Children's Reasoning about Intentions, Beliefs and Behaviour) model was developed to investigate reasoning in young children [29]. This schema was implemented as a computer model to simulate knowledge and the inference processes of a child solving problems [30]. For the current work, a major difference to the standard BDI model is that different degrees of belief are ascribed to belief statements, according to the source of the belief. A preference operator allows discrimination between beliefs that are based on assumption, perception and deduction. Perception can be further sub-divided to include direct perception and indirect perception (i.e. from another agent that has some degree of trust associated with it). Hence beliefs can be ordered according to the degree of trust in them.

The CRIBB computer model did not incorporate emotions present in the original cognitive schema. The a-CRIBB model [31] was developed to investigate the use of affective computing within the CRIBB schema. a-CRIBB added several new elements to the original CRIBB computer model. In moving from the first to later CAMAL models (i.e. [6–8, 32]), these novel aspects of a-CRIBB were further developed, and integrated into the existing CAMAL model (Figure 2). In terms of the BDI model, degrees of belief could be added to every component of the a-CRIBB BDI schema based on a distributed model of affect. The interpretation of these real-number values is consistent with the semantic interpretation of motivational states, as proposed by Sloman [4] and Davis [7, 8]. Since with situated agents, many forms of sensor may be available, degrees of trust can be ascribed to them (and modified over the life-time of any experiment). This gives rise to Belief statements based on these sensors (direct perception), as well as assumption and deduction, taking real-number values which can propagate across the BDI schema.

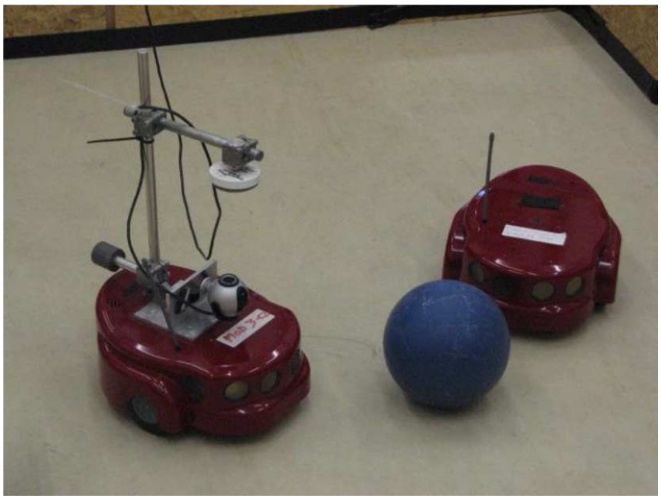

Figure 2. robo-CAMAL (left) in an experiment environment with ball and a further robot within a bounded maze.

Affective computing refers to the use of computers to explore emotion within cognitive architectures [33]. There are many different models of affect, and theories of emotion can be typified as belonging in one of several types, for example physiological [34], evolutionary [35], expressio [36], appraisal [37] or goal based [38]. The affect model (extended from that in a-CRIBB) distributes affect values across the entire architecture rather than have a centralised emotion module.

Rather than use a centralised model of affect, CAMAL uses a distributed model of affect. We do not associate emotional tags to these processes, such as fear of goal failure or happiness of association success, or indeed embody emotions in the architecture for reasons outlined elsewhere [32]. In short, work on emotion is a morass of definitions and competing theories. We suggest that we should not further this confused framework with further models of emotion for artificial systems. The thesis is that overall the theory of emotion is too disorganized to be of much use in the design of synthetic intelligence and that more pointedly, emotion is not really a requirement for synthetic intelligence. Obvious exceptions are intelligent interface systems that require emotional recognition [39] and deep language understanding systems [40]. It is suggested that a direction given by the less semantically overloaded term affect is a more appropriate.

This model of affect means that various elements within the architecture have an associated magnitude that can fluctuate according to success or failure associated with that element as mapped onto other processes and eventually actions in the environment (see Table 1). The underlying mechanism is the same affective range (real-number valued minus to plus 1), but used and modified in different parts of the architecture according to need. For example, each belief has a confidence value which reflects the reliability of that belief. Furthermore each goal has an importance value that determines the level of relevance of that goal to the agent at that time. Also the association value indicates the likelihood of success of a specific plan given a specific belief-desire combination. All these values fluctuate and are often highly dependent on other systems within the architecture.

Associations are a construct that consist of a belief, a desire, an intention, and an association value (insistence) [8]. The associations provide an indication of the past success of a specific set of plans given the agent's current beliefs and desires. This allows the agent to consistently determine the most appropriate set of plans based on its beliefs and desires; and to modify them where they fail. Associations take the general form:

$\mathrm{assocation(\ BeliefSet,\ Goal,\ BehaviourSpecification,\ Insistence)}$

Associations can be pre-defined (typically a small number related to high priority tasks for specific environment configurations), or formed when the architecture is initialised, or dynamically created when existing associations fail. Meta-level operators define which of these modes (or combinations) is to be preferred. Where meta-level functioning is minimised (for example in the first implementations of robo-CAMAL), the architecture is simply configured to run in a set configuration, and user intervention is required if an alternative mode is required (through changing what in effect becomes a architecture spanning global parameter).

The associations work in the following way. From a large list of associations the agent extracts only those that have a belief-desire combination that correspond to the agent's current belief and desire set. Of the remaining associations the one with the highest association value is chosen. If multiple associations have identical insistence values, then the association with the most important goal is chosen. If multiple associations still remain, belief preference is used then list order to resolve the conflict set. The resulting association represents the set of plans that are the most likely to achieve the given goal. The association value is modified depending on the outcome of the agent's actions. If it fails to achieve its goal, the value is reduced. If the plans succeed in achieving the goal then the value is increased. Goals too have an affect value (importance), which depending upon the architecture's mode of operation, remains static over an experiment or can be varied. When in dynamic goal importance mode, goal importance is decreased where no available association (i.e. plan) proves to be successful, and temporarily reduced when achieved.

The use of motivation is pervasive throughout the architecture [7, 8]. The most important aspect here is the use of a motivational blackboard, through which the deliberative systems are coordinated. A blackboard system [41] uses three components. The first is the blackboard, which is a global structure and holds all the information relevant to current and past motivators (whether adopted or not) such as the agent's beliefs, goals, association, feedback etc. This structure is accessible to the whole deliberative agent. The second component consists of various knowledge sources which access the blackboard. These extract the relevant information, manipulate it in some way, and then post the result back to the blackboard. This could be a belief or goal update mechanism for example. The final element is a control component; in this instance a motivational construct and its management. The construct and management act as a top-level scheduler for the architecture, causing perceptual update, behaviour feedback, belief revision, goal update etc. To be called in turn. This high level reasoning cycle can fork into alternative processing cycles according to the effect each step has.

4. robo-CAMAL: A Cognitive robot

The cognitive architecture developed here (as robo-CAMAL) is a combination of the deliberative components described in the previous section and an asynchronous reactive sub-architecture, implemented on a mobile robot (left in Figure 2) and a desktop computer [11]. Perceptual data (from sonar and omni-directional camera) and control commands are passed between the robot and the computer via a radio modem and a USB cable. This section will highlight some of the main components of the architecture, highlighting aspects that differ from the generic CAMAL architecture.

4.1. robo-CAMAL Architecture

robo-CAMAL is a reactive-deliberative hybrid architecture used to control a mobile robot. It makes use of simplified CAMAL architecture with reactive sub-architectures tailored to the robot and its sensors, and no meta-deliberative layer. A schematic of the architecture can be seen in Figure 3.

The deliberative component works as a blackboard system. A motivational blackboard contains information related to current and past motivations (and their parts). The reasoning module acts as a co-ordinating control component. The various update modules (for example Belief Revision, and Goal Selection as discussed in section 3.2), and the affect model related to motivational states, are the knowledge sources. Once instantiated (i.e. domain model loaded or indeed updated), the run-time control cycle is, at its simplest, attend to feedback on blackboard, then call (in order) belief revision, goal update, association update, motivator update and motivator activation. Motivator activation calls on the reactive processing (or some other module, e.g. association generation). This relatively simple high level reasoning cycle allows the different knowledge sources to access and update the motivational blackboard, and where appropriate re-enter the processing cycle earlier in the chain. At the reactive level sensor data is passed to the perception module. The perceptual module uses a set of pre-defined rules to map the sensor data and reactive-goal feedback to deliberative messages (in effect a list of new belief statements). This perceptual message is posted to the motivational blackboard. The reasoning module then allows the various knowledge sources access to the blackboard in a specific order. First the belief

Table 1. Affect metrics used in Motivator Constructs and constituents.

Affect metric	Aspect	Process and dimension category	Affect magnitude
Belief Indicator	Motivator	Truth values for Semantic Content and Motivator Attitude; with following preference: Perception $\gg$ Deduction $\gg$ Assumption	[0, 1]
Commitment	Motivator	Motivator Acceptance (ignored to first priority)	[0, 1]
Dynamic State	Motivator	Motivator Process (uninstantiated to complete)	[0, 1]
Importance	Goal	Goal Importance (low to high)	[0, 1]
Insistence	Association	BDI Association Strength (low to high)	[0, 1]
Intensity	Motivator	Motivator Strength (low to high)	[0, 1]
Urgency	Motivator	Urgency (low to high) or time cost function	[0, 1]
Decay	Motivator	Motivator Decay (low to high) or time cost function	[0, 1]
Reinforcer	Affect	Goal and Association Feedback (negative to positive)	[-1, 1]

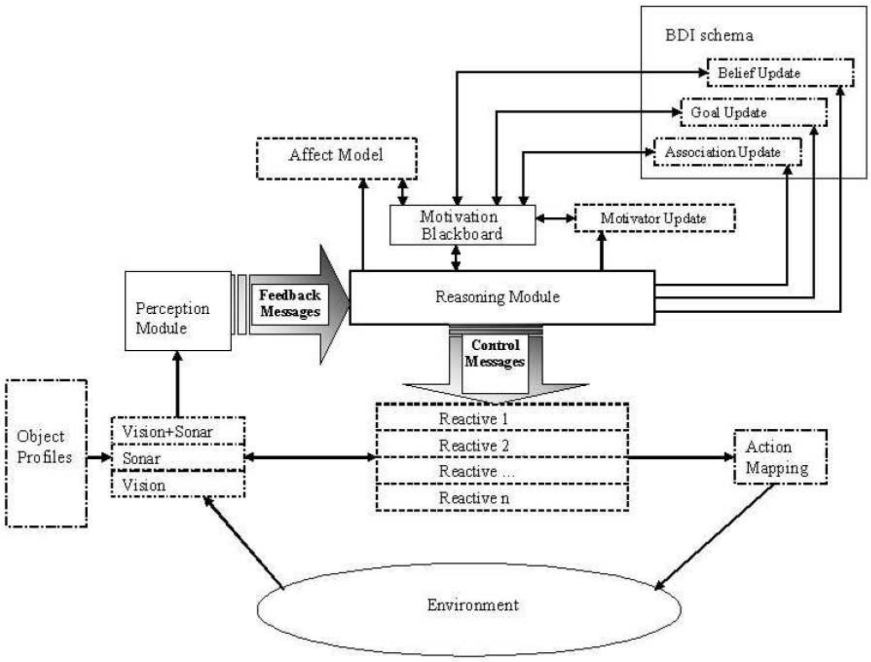

Figure 3. The robo-CAMAL architecture.

update uses the new information to modify its belief set. The goal update then uses the updated belief set to determine if the current goal has been achieved, and what the new goal is. The association update then uses the new belief and goal set to determine the relevant action or intention. The BDI schema is implemented through the use of associations. An association is a coupling of a belief-desire-intention triplet, together with an affect value. This affect value details the likelihood that the intention (behaviour or reactive sub-architecture) of a given association will achieve a goal given a specific belief. For example, an association with the form
$\mathrm{association(found(ball),\ hit(ball),\ moveTowards(ball),\ 0.75))}$
is more likely to hit the ball than the association
$\mathrm{association(found(ball),\ hit(ball),\ moveAwayFrom(ball),\ 0.25))}$.

Associations are chosen based on the agent's current beliefs and desires. The agent's intention is then chosen based on the remaining association's insistence values as described above for CAMAL.

If an agent is to control its motivations, it must be able to set its own goals. If an agent is to choose its own goals, then it must understand the context of those goals in relation to its environment. In essence this means that a situated and embodied agent's motives must be grounded in its environment. This problem can be viewed as part of the wider grounding problem on how to integrate an agent with its environment [42]. One way of grounding motivation in an agent is through the use of reactive systems. For example, a motor powered by a photosensitive plate may be motivated to seek light. There is an argument that evidence of motivation within reactive systems lies with the observer and is not objective [43]. However, if this argument is left to one side then the systems motivation is embodied in the control architecture of the sensing and acting mechanisms [44].

If an agent's motivation is based on internal representations, the issue of grounding is no longer as straight forward as for a reactive system. These representations, and therefore the relevant motives, must be connected to the appropriate events and activities in the agent's environment [42]. In essence an agent's motivation must be grounded to its environment through its actions. The use of associations that include reactive behaviours as their intention ensure this. Savage suggests that, motivational grounding is derived from the interaction of an agent with the appropriate aspects of its environment [42]. The feedback from the reactive subsystems, as described above, ensures this.

This section has discussed how motivation can be integrated with the agent's environment using a reactive system, or an interactive process in the case of symbolic motivational states. It has not, however, provided a way of determining whether a motivation is a simple reactive behaviour, or a symbolic representation. The indices of motivation models look to provide a way of making such a distinction [42, 45].

This model provides three features that may distinguish between reactive and deliberative motivations. The first is individuation which describes an agent's ability to achieve the relevant goal using a number of different strategies. That is, the agent's ability to achieve a goal using an alternative method if its preferred response is blocked.

The second characteristic is the formation of expectancies relating to its goal object. This relates to cognitive representations reflecting aspects of the goal object such as how it will react under certain conditions.

The third characteristic relates to the presence of an affective response towards the goal object. For example, an agent should an affect value associated with an object that changes depending on the agents' interaction with that object. The Perceptual Feedback-BDI-Reactive cycle in robo-CAMAL does just that, with belief, goal, intention, association, and motivational affective values varying over time.

4.2. Reactive component

A reactive robot is one where the perceptual input is directly connected to the motor output [46, 47]. There can be various different definitions of what constitutes a reactive component. For example the system may have no changeable internal state so that the current input determines the current output. In our broad research, we term these reflexes [7, 8]. In this case the output is always the same given the same input. The definition of a reactive system taken here is that the systems output is determined not only by its input, but also by its internal state. This is akin to a finite state machine. The system's output and new state is based on its input and its current state.

The reactive component consists of a number of several different reactive behaviours. These behaviours are modelled using software written on the desktop computer, and the robot circuit board, as opposed to being hard-wired into the robot. The lowest level consists of simple micro-behaviours that turn the robot left, or move it forward etc. These micro-behaviours are programmed directly on the robot. The micro-behaviours are combined to generate task specific macro-behaviours e.g. find or hit a specific object, or avoid objects etc.

The micro-behaviours can be grouped in specific ways to produce multiple macro-behaviours capable of (potentially) achieving specific goals such as hit(ball), track(redrobot), find(blackrobot) or avoid(objects). For example, one reactive behaviour uses sonar to avoid objects on the right, where as a second uses the vision system to achieve the same goal; a third instantiation, uses both sonar and the vision system. Furthermore, the micro-behaviours can be combined using four different arbitration methods: Priority Method where micro-behaviour activation preference is stated through design; Aggregate Method where the aggregated micro-behaviour is selected; Winner Method where weights are used to determine the preferred single micro-behaviour, the weights can be changed at run-time; and Behaviour Suppression, where micro-behaviours when activated deselect other micro-behaviours. This provides twelve different methods of performing any specific task based behaviour (the four arbitration methods, plus the three possible perceptual modes (sonar, vision, sonar and vision).

The specific behaviours that make up a task dependent behaviour group are determined prior to runtime, and defined in the Domain Model. The specific behaviour grouping, combination method, and sensor mode, is chosen at runtime by the deliberative component.

4.3. The Domain Model

robo-CAMAL operates within a structured, dynamic environment consisting of various objects (walls, maze, balls and other robots (both static and dynamic)). Its actions are also confined by what its physical body can perceive and do. It is therefore vital that information about the agent's environment and its physical body be available to the agent. This encoding is achieved with the use of a domain model. Together with the three aspects described below, the domain model includes variables that define values used in updating goal and association value, and other parameters that define thresholds for the various components of the architecture. In a fuller implementation of robo-CAMAL, a meta-deliberative layer could control these parameters and hence refine and optimise the various modules to suit the current task and environments. Indeed, this has been investigated using CAMAL with simulated environments [48].

The domain model first defines the type of objects to be found within the agent's environment. There is also an abstract belief schema that details the structure and constituents of all possible beliefs the agent can have about its environment. The domain model therefore defines constraints on the possible beliefs that can be generated by the agent. It also defines the relationships between possible beliefs, for example synonyms and antonyms. For example, for many experiments, the domain model specifies a number for found objects that causes the belief environment(cluttered) to become true, and the default belief environment(sparse) to become false (see Figure 4 for first experiment). Such beliefs act as constraints on the belief revision system, and reflect tasks and environments. These belief definitions and relationships incorporate the situated nature of the agent into the more abstract BDI schema used in the architecture.

The domain model also defines the goals the agent can have. These goals are designed to reflect the possible objects and beliefs defined by the model. They are also constrained by the possible actions the robot can perform. The domain model provides a list of all the possible actions the agent can undertake. In relation to robo-CAMAL, this refers to the macro-behaviours. These two elements incorporate some of the embodied nature of the agent into the architecture.

Finally the domain model provides the object perceptual profiles, i.e. the information required to recognise an object. This too incorporates both the situated and embodied nature of the agent into the architecture. It is situated in that it provides some of the physical attributes of the environments objects. It is embodied in that it provides information on how the agent's sensors should process the perceptual data.

The use of the domain model provides a number of key advantages. The first is that the model allows the situated and embodied nature of the agent to be separated from the deliberative component. This means that, as is the case with robo-CAMAL, the deliberative component can be generic and not task domain and environment specific. The second advantage is that the model makes it easy to pinpoint when and where within the architecture the anchoring of symbols occurs. The

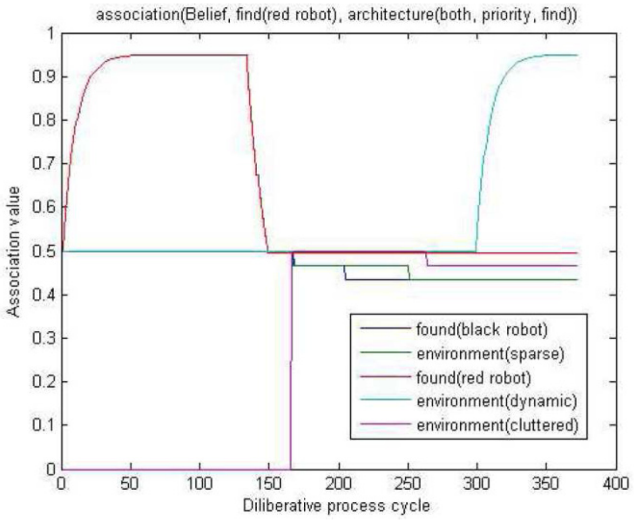

Figure 4. *Association generation process for the find(redrobot) experiment.*

domain model ensures that the generic architecture can be tuned to robot or agent capabilities, environments and tasks within those environments. A small change in task or environment, with the same robot (for example moving from find(ball) and avoid(robot) to avoid(ball) and find(robot)), involves a small change to the domain model and not to the robot architecture. This enables a number of small task-centric domain models to be built independently, and then merged to form more general purpose domain models. As the domain model includes goal importance and intention insistence values, an architecture that included task optimisation (via adaptation or learning) could optimise a given domain model for specific tasks and environments. This can be saved and used in subsequent trials with the robot architecture.

4.4. Learning in robo-CAMAL

robo-CAMAL makes use of associations in order to learn about the effect of its actions on its environment. Machine learning involves building systems that can use example data or past experience to optimise their performance [24]. There are many possible methods from supervised learning, where the system is provided with controlled training data, to unsupervised learning, where the system is given no labelled data and learns by its own means.
Standard BDI schemas [28, 49] do not provide mechanisms for learning. Given that robo-CAMAL uses a variant on the BDI schema, to model the reasoning of a five-year old child, the research into developmental robotics [5] seems apt, but beyond the current project. Phung et al. use an Inductive Logic learning paradigm with a BDI agent [50], while Subagdja et al. use meta-level operators to learn plans that fit into a BDI schema [10], following on from Subagdja and Sonenberg who found standard Q-learning with a BDI schema not to be a productive mechanism [51].
The CAMAL architecture can build new associations on initialisation or through a goal failure mechanism. At its simplest, this mechanism generates all possible belief-goal-intention combinations; each given a default association value. Whether using a given set of associations (defined in the selected Domain Model), or using a generated set, robo-CAMAL can adapt these associations through the modification of the affect value. Here a simple learning algorithm, based on reinforcement learning with an immediate reward, is utilised. Reinforcement learning involves an agent performing an action within its environment. It then receives a reward or penalty based on the result of that action. By trying several different actions and using the feedback provided, the agent attempts to learn how to maximise the total reward [25].

The method with which robo-CAMAL learns is a simplified Q learning approach. The association value can be written as A(s; g; a), where s is the state, g is the goal, and a is the action. Given a constant goal g, the algorithm for determining the association values is as follows.

```
robo-CAMAL learning algorithm
For each s,a pair initialise the table entry A(s; g; a)
Observe the current state s
Repeat
Select an action a and execute
Observe new state s'
Use s' to determine if r is positive or negative
Use r to update the table entry for A(s; g; a)
s ← s'
```

There are however some important differences between the robo-CAMAL and Q learning algorithms. The first is that the reward function is known to the agent. This means the agent can calculate its reward based on the observed new state. The second important difference is that robo-CAMAL is opportunistic. During the training phase, the learning attitude keeps the goal constant when looking for new association mappings. As the goal is constant, the action chosen is the relevant action-state pair with the highest association value. This means if the agent finds an action that provides a positive reward, it continues to execute that action. This is because only the immediate reward is considered. In the robo-CAMAL learning algorithm, the association update equation is the same as the Q update equation with γ equal to 0.

There are several different ways of initialising robo-CAMAL to achieve different levels of supervision when learning. The maximum level of supervision involves pre-defining the associations for a specific goal. For each macro-behaviour there are twelve possible architectures. For example the following set of four associations:

association(environment(cluttered), avoid(collisions), architecture(vision, priority, avoid), 0.5)
association(environment(cluttered), avoid(collisions), architecture(vision, aggregate, avoid), 0.5)
association(environment(cluttered), avoid(collisions), architecture(vision, winner, avoid), 0.5)
association(environment(cluttered), avoid(collisions), architecture(vision, suppression, avoid), 0.5)

define the four reactive sub-architectures for avoid(collisions), with differing arbitration methods. Another eight exist for the sonar only and vision+sonar variations. Given all the associations are pre-defined and the goal is constant, the agent only has to learn which architecture is the most successful. This is in essence telling robo-CAMAL which set of alternative reactive sub-architectures should achieve its goal, and asking it to determine the best.

A relaxation in the level of supervision is to allow robo-CAMAL to generate its own associations, but control the environment. In this scenario the agent has no indication of what effect each action will have on its environment. However, as the environment contains the correct object with which it can achieve its goal, it should learn the most appropriate action. This method provides robo-CAMAL with training data in order to learn the best policy. The minimum level of supervision allows robo-CAMAL to generate its own associations in an uncontrolled environment.

5. Experiments

A considerable range of experiments with the robot have been performed [11]. Here those most relevant to motivation and the issues described in the paper are highlighted.

5.1. Association Creation

Associations can be pre-defined prior to run time. However, this has the effect of controlling which actions are to be used for each belief-goal combination. For example, the association:
$\mathrm{association(found(blueball),\ hit(blueball),\ architecture(vision{+}sonar,\ priority,\ hit),\ 0.8\ )}$
links the goal hit(blueball) to the specific architecture architecture(vision+sonar, priority, hit), with an affect intensity of 0.8.
If the associations are not pre-defined then robo-CAMAL has a set of possible actions with no indication of their purpose (although they are named to be meaningful to the human user). In this case robo-CAMAL needs to generate associations. It then needs to test the new associations to determine which is the most appropriate for each situation.
New associations are created as follows. First, all the agent's goals are placed in a list. Each goal is paired with each belief the agent has at that time. Each belief-goal pair is combined with every possible reactive sub-architecture. Each new association has its association value set to 0.5. This means that for each belief-goal pair 48 new associations are created. For example if the agent has the goal hit(ball), the two beliefs environment(sparse) and found(ball), the associations created are:
$\mathrm{association(environment(sparse),\ hit(blueball),\ architecture(1.48),\ 0.5)}$
$\mathrm{association(found(blueball),\ hit(blueball),\ architecture(1.48),\ 0.5)}$
giving a total of 96 new associations.
The following experiment gives a practical example of association creation. The system was set up as follows. For this specific experiment, all the micro-behaviours were deactivated; in effect, the reactive component of robo-CAMAL was reduced to nothing more than a vision system. The deliberative component was initialised with the goal find(redrobot), and the belief environment(sparse). The reactive cycle number (defining how many clock cycles the reactive sub-system is to run for) was set to 20. The vision system was initialised to detect an object corresponding to the object profile produced by the redrobot. A stationary redrobot was placed in front of robo-CAMAL within its lower proximity threshold (i.e. the redrobot was within a distance detectable to the vision system).
Once the experiment was started, the reactive component provided feedback that the redrobot had been found. After one minute the redrobot was replaced by the blackrobot. At this point robo-CAMAL created associations involving the beliefs found(blackrobot) and environment(cluttered). After a minute the blackrobot was replaced with the redrobot.
Figure 4 shows several associations with the same goal and intention. Each line represents an alternative belief basis. Initially associations with the beliefs environment(sparse), environment(dynamic), and found(redrobot) are created. The belief environment(dynamic) comes from the domain model assumption that if a robot is present, then the environment is dynamic. Initially the association with the belief found(redrobot) increases as it achieves its goal. Once the redrobot is removed then the association fails and its association value is reduced. Once the blackrobot is introduced then new associations involving the beliefs found(blackrobot) and environment(cluttered) are created. The belief environment(cluttered) is generated due to the domain model assumption that more than two objects means a cluttered environment. This appears in the figure as the association values that jump from 0 to 0.5 (it is shown as 0 as it was not created initially as the belief basis is antonym to the default belief environment(sparse)). The gap between the reduction of the found(redrobot) association and the creation of the new association is due to the finite time required to swap the two robots. As the reactive cycle number was set at a low value, the time taken to change the robots over is significant. Once the redrobot was reintroduced the association with the belief environment(dynamic) increases. This experiment appears to show that robo-CAMAL is successfully learning the appropriate behaviour to achieve its goal. However no hard conclusions can be drawn from this as the experimental set up was so contrived. These results are given to demonstrate association generation in a limited but changing environment where the perceptual input and therefore the belief set could be controlled.

5.2. Adaptation Using Goals and Associations

One important requirement for any agent is that it has the ability to adapt to a changing environment. There are several ways in which an agent can adapt, evolutionary adaptation, physiological adaptation, sensory adaptation and adaptation by learning [14].
Evolutionary adaptation occurs when agents adapt to their environment over many generations via natural selection. Physiological adaptation refers to the physiological changes that occur in response to changes in the environment. For example, sweating is a response to an increase in temperature in the environment. Sensory adaptation is when the perceptual systems adjust to the strength of the stimulus that they are sensitive to; for example, when the pupil dilates due to a change in light intensity. Adaptation by learning is a very general adaptation. It can refer to many kinds of things such as learning the quickest way to a specific location, or how best to avoid a predator. The way in which robo-CAMAL adapts is through the use of associations.
As described, each association contains four elements, a belief, a goal, an action, and a measure of the likelihood of success of the action given the belief and goal. For robo-CAMAL to adapt, it needs to choose the appropriate association. That is, the association that corresponds to its current environment and internal state. This is done using a two stage process.
The first stage (training phase) involves the agent learning which associations are apt for a given belief-goal combination. For example the association:
$\mathrm{association(found(blueball),\ hit(blueball),\ avoid(blueball),\ value)}$
will tend to fail to achieve its goal, while the association (for the same belief-goal pair)
$\mathrm{association(found(blueball),\ hit(blueball),\ hit(blueball),\ value)}$
should achieve its goal most of the time. It is the job of the training phase to determine the relevant associations.
In the second stage of adaptation after the training phase, robo-CAMAL has multiple goals in a variable environment. The task then becomes to choose the appropriate association with which to achieve one of its goals. This choice is based on robo-CAMAL's internal state, and its environment (or more specifically its beliefs about its environment). In order for robo-CAMAL to choose an association, it needs to rank associations. This ranking is calculated in terms of the agent's belief, goal, and association values. The value of each associations rank is calculated using Equation 1. Here a_v is the association's insistence value, g_v is the association's goal importance value, b_a is the age of the association's belief.

$$Rank = \sqrt{a_v g_v}\frac{1}{b_a + 1} \qquad (1)$$

Figure 5 shows how the association rank value varies with each of the three affect values av, g_v, and b_a. For each line on Figure 5, two of the terms in Equation 1 were kept constant, whilst the stated parameter was varied. It is clear from Figure 5 that the association rank value increases as a_v and g_v increase. It is also clear that the association rank decreases as the belief gets older. This means that for each association the higher its goal importance, the more recent its belief was formed, and the more likely the action is to achieve its goal, then the higher its rank value.

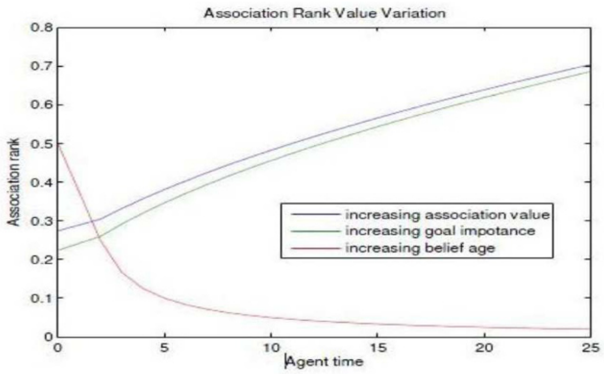

Figure 5. *Effect on association rank value by major contributing factors.*

Over time, the association value varies according to Equations 2 and 3. Here vi is the current association insistence value, and v_{i+1} is the new association insistence value. If the agent achieves its goal, the relevant association insistence value is increased using Equation 2. If the agent fails to achieve its goal the relevant association value is decreased using Equation 3. While numbers are given in these equations, the numbers are taken from variables defined in the domain model.

$$v_{i+1} = v_i + ((0.95 - v_i) \cdot 0.1) \tag{2}$$

$$v_{i+1} = v_i - (v_i - 0.15) \cdot 0.1) \tag{3}$$

The default initial goal importance value is 0.5 (or some other number defined using a variable in the domain model). It is possible to specify other values through configuring the domain model appropriately prior to initialisation. At each time step the importance value is increased by 0.02, up to a maximum value of 0.95 (again variables in the domain model are used to define these values). The only occasion in which the goal importance value is not increased is if the agent has failed to achieve that goal, or if the agent attempted to achieve the goal in the previous deliberative cycle. If a goal is achieved its importance value is reset to 0.5 (the default goal value). If a goal consecutively fails X times, where X is determined by the domain model threshold goal threat max, its importance is then set to 0.1. The goal importance variation can be seen in Figure 6.

Figure 6 shows a goal with its importance value set at 0.1. Its value increases over time. Just before cycle 30 the goal is achieved, and its importance value is set to 0.5. The importance value then increases again, until the goal consecutively fails X times. At this point (cycle 45) the importance value is reduced to 0.1; it subsequently rises.

The goal importance function has several effects. The first effect to notice is the goal failure strategy. If a goal fails once its value is not

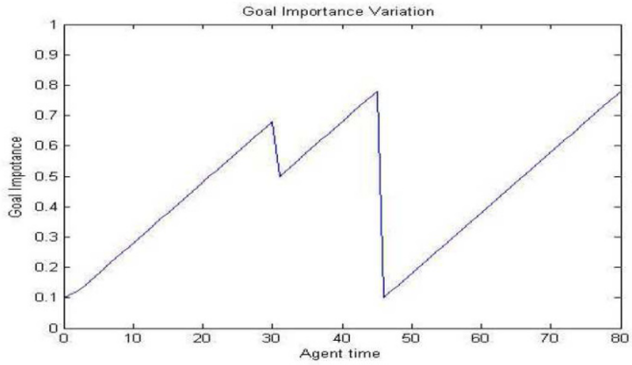

Figure 6. *Goal importance value fluctuation over an experiment.*

automatically reduced. The reasoning behind this design decision is that the failure may be due to something simple beyond the agent's control. For example, if the goal is hit(blueball), the agent may simply miss, or a second robot may get in the way. However if robo-CAMAL consecutively fails on a number of occasions, then it is likely that there is something more substantial preventing it from achieving that goal. For example the blueball may no longer be present or have been moved by further robots in the environment. If this is the case then the goal importance is greatly reduced to prevent its selection.

The goal's success has two effects on the goal importance depending on its previous value. If the goal importance value was initially low due to a previous failure, its value is increased to 0.5. This reflects that the goal is now achievable. If the goal importance value was initially high, the value is reduced to 0.5. This reflects that the goal has been achieved and is no longer as important. This prevents goals with a high importance value being selected repeatedly.

The increase in the importance value over time reflects the variable nature of the environment. A goal that is unachievable at time t may become achievable at time t+x. For example, if there is no blueball in the environment, the goal hit(blueball) will fail. Its goal importance value will be reduced to 0.1. If a blueball is then introduced to the environment, the goal then becomes achievable. Unless the importance value increases then the goal hit(blueball) will not be selected.

The time the belief was formed also affects the association's rank value, as the older the belief, the lower the association's rank. This is to reflect the fact that in dynamic environments older beliefs may not be as reliable or accurate as more recent beliefs. Belief revision is described in earlier sections and again the domain model has a role in defining maximum length of time a perception based belief can exist, and other reasons for updating or negating a belief.

These experiments have shown that agent adaptation within robo-CAMAL can be divided into two components. The first is a training phase. First robo-CAMAL uses the method described to learn the most appropriate associations for each goal. The most successful associations can then be retained by the agent. Once the training phase is over, the agent's environment becomes variable, and it is given multiple goals. At this point robo-CAMAL uses its current beliefs, and the various affect values, to choose the most relevant association. The changing beliefs and affect values should mirror changes in the agent's environment. The modification of goal importance values over time ensures that the robot systematically tries various tasks as represented by the goals in its domain model.

5.3. Association Learning

This experiment was designed to test robo-CAMAL's ability to learn the correct action to achieve a specific goal. To do this robo-CAMAL needs to be able to generate a list of associations, and select the correct association to achieve that goal.
For this experiment no reactive behaviours were disallowed but robo-CAMAL was instantiated with a single object based goal, and with the correct initial set of beliefs. No associations were pre-defined. robo-CAMAL was run in one of a number of possible environments for five minutes. The experiment was repeated three times for each environment. The experiment was run for every object based goal (i.e. find, track, and hit), with every possible object as the focus of that goal. Given three possible goals, three possible objects, with three experiments in six environments, the total number of experiments was 162. Each experiment produced a number of associations. The value of each association was recorded at every deliberative processing cycle. Abridged results (for the redrobot experiments) are presented here.
Figure 7 shows eight key associations (those that attain an insistence value greater than 0.5) for the hit(redrobot) experiment. Initially robo-CAMAL has no associations. Given the goal hit(redrobot), a set of pre-defined beliefs, and a list of the possible actions, robo-CAMAL produced 144 different associations for this experiment; some of these associations have default or lower values, and are never chosen. Eight associations are shown, with two highlighting what happens to initially promising but unsuccessful combinations (these trail off after point a). Only five associations achieve an insistence value significantly greater than the default value (i.e. above 0.75), and once beyond point a in Figure 7 (at around 50 cycles), only four different associations are selected.
Initially robo-CAMAL tries various associations in order to hit the redrobot, all of which fail. However, at point a, the association(environment(dynamic), hit(redrobot), architecture(sonar, priority, hit)) succeeds. This can be seen in Figure 7 as the line with the increasing association value. After some time robo-CAMAL fails to hit the redrobot. This can be seen by the fall in association value at point b. At this point as the association value is high, the only reason the association is not chosen must be because the redrobot cannot be sensed and the relevant belief is no longer present. An alternative association is used to find the redrobot, association(environment(sparse), hit(redrobot), architecture(sonar, priority, find)), both at this point and again 50 cycles later. This association subsequently fades. An alternative find behaviour (architecture(vision, priority, find)) is chosen at point c, but this too subsequently fades, as the redrobot is subsequently never lost by the robot's perceptual system.
If the redrobot has not been found within 25 deliberative processing cycles, then all beliefs regarding the redrobot are removed. With no object beliefs present, robo-CAMAL will deduce that the environment is static and sparse. Between point c and d, the predominant association in this experiment (architecture(sonar, priority, hit)) contains to be selected, except where the robot is sensed but not close enough to be hit, in which case the adaptive BDI schema sees a tracking behaviour chosen (architecture(vision, priority, track)). For the remainder of the experiment, beyond point d, the robot is close enough for a hit behaviour to succeed. However beyond point e, an alternative perceptual system works better at times (architecture(vision, winner, hit)). This is represented in Figure 7 as the increase in association value from point e to f. Beyond point f, it can be seen that the two main associations closely mirror each other, and have almost have identical shapes. This occurs because robo-CAMAL is switching between the two associations, due to the way the Belief revision model works. Initially the association with the belief environment(sparse) succeeds. The belief is considered true so is left unmodified. However, as the belief found(redrobot) is held, the belief environment(dynamic) is deduced. This belief is updated with the current time value. This means that the environment(dynamic) belief is more recent than the environment(sparse) belief.
One of the factors affecting the choice of associations is the age of the belief (as shown in Equation 1). In this case, as the environment(dynamic) belief is more recent, the association containing this belief is therefore chosen. This cycle repeats for the environment(sparse) belief, thereby causing the chosen association to swap every deliberative processing cycle. If the experiment were left to continue, no doubt further associations would rise to the surface, as the goal (hit(redrobot)), like many others in our experimentation, requires a series of actions (and goals) to be performed. Figure 7 shows three hit, three track and two find associations; but others exist in the 144 possible associations. robo-CAMAL does not use a Hierarchical Task Network [52], but requires a rational design of any task specific Domain Model. In this case, the goal hit(redrobot), requires robo-CAMAL to have first found the redrobot, then to have used the track behaviours to be close enough to the goal object to collide with it.
To assess the effectiveness of robo-CAMAL's learning ability two criteria were set for each association to meet. The first is the association insistence value. If this value goes above a specific threshold, then the association is recorded. The thresholds chosen were 0.65 and 0.75. These values were chosen as the association value is considered to be the likelihood of success of that association. For example, an association with a value of 1 should have a 100% success rate, where as a value of 0 would always fail. The association values of 0.65 and 0.75 therefore represent likelihood values of 65% and 75%. These are considered reasonable likelihood values to consider an association as being an accurate reflection of a specific belief-goal-action mapping.
The second criteria relates to the amount of time the association spends above the value threshold. Five time thresholds were chosen. These were one deliberative processing cycle, 30 seconds, 1 minute, 1 minute 30 seconds and 2 minutes 30 seconds. The number of processing cycles spent above the association threshold divided by the total number of deliberative processing cycles gives a percentage of the time the association spent above its threshold. This can easily be converted into a real time value as each experiment ran for five minutes. If an association persisted longer than the persistence threshold it was recorded.
Table 2 shows the number of correct and incorrect associations found during the learning experiment. It also shows the ratio of correct to incorrect associations as shown in Equation 4.

$$ratio = \frac{correct \quad associations}{incorect \quad associations} \tag{4}$$

Any ratio value above 1 indicates that robo-CAMAL has successfully found more correct associations than incorrect ones. The higher this ratio, the more accurate the learning mechanism.
The results show that overall robo-CAMAL correctly identifies the appropriate association to achieve its goal more often than not. However, if the shortest time constraint is chosen, robo-CAMAL's performance is only slightly better than random. As the ratio for this constraint is only just above 1, robo-CAMAL is identifying the correct association only slightly more than 50% of the time. If an association is required to persist for more than 30 seconds, then the accuracy of the learning mechanism increases almost three fold. It is clear from the results that the longer the association is required to persist, the greater robo-CAMAL's learning accuracy.
This increased accuracy comes at a price. It is also clear that the longer the association is required to persist, the fewer the number of associations to be found. This can be seen in Table 3, which shows the average

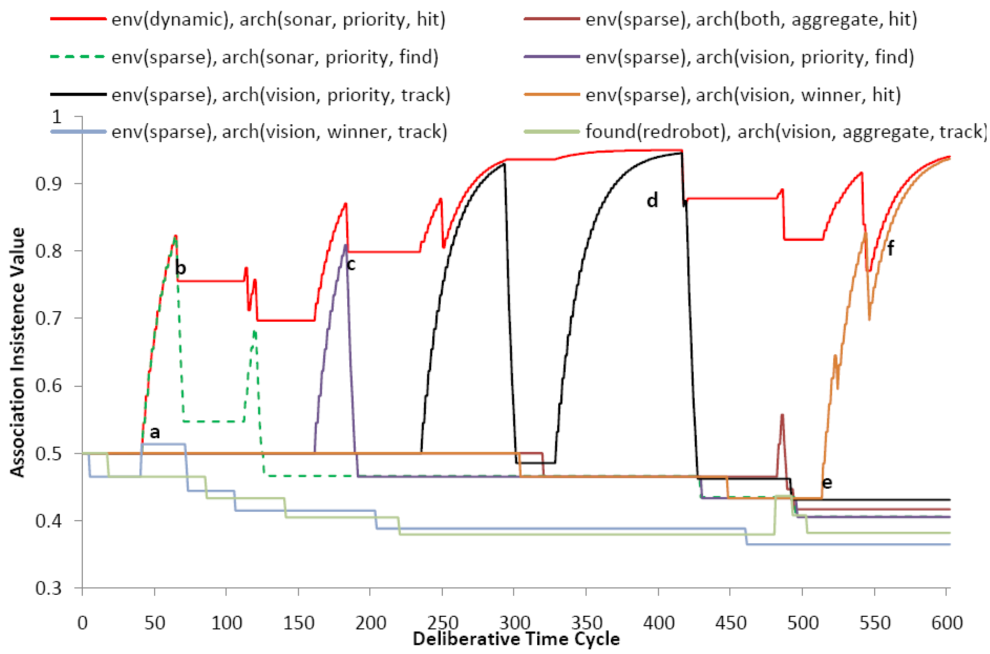

Figure 7. *Variation in association insistence in the hit(redrobot) learning experiment.*

Table 2. *Total number of correct and incorrect associations found.*

Time above threshold	Associations above value threshold					
	0.65			0.75		
	correct	incorrect	ratio	correct	incorrect	ratio
One processing cycle	210	159	1.32	181	106	1.71
30 sec	135	36	3.75	114	22	5.18
1 min	106	22	4.81	98	18	5.44
1 min 30 sec	91	16	5.69	81	8	10.13
2 min 30 sec	68	3	22.6	52	0	N/A

Table 3. *Total number of correct associations found in 10 minutes.*

Time above threshold	Correct associations found per 10 min	
	Value threshold 0.65	Value threshold 0.75
One cycle	2.59	2.23
30 sec	1.67	1.40
1 min	1.31	1.21
1 min 30 sec	1.12	1.00
2 min 30 sec	0.84	0.64

number of associations found in a ten minute period. The total time of the experiment was 810 minutes. This means the total number of accurate associations found divided by 81 gives the average number of associations found over a 10 minute period.

It is clear that increasing the time an association is required to persist for increases the time it takes for robo-CAMAL to learn the correct associations. For example, even though an association persistence of 2 minutes and 30 seconds provides a very accurate learning threshold, it takes robo-CAMAL over 10 minutes to learn a new association. This behaviour is expected. The longer an association is required to persist, the less time is available to try alternative associations. From Tables 2 and 3 it can be seen that increasing the association threshold value has the same effect on robo-CAMAL's learning accuracy and speed, as increasing the association's persistence threshold.

6. Discussion

The experiments presented the first use of the CAMAL architecture with an embodied agent. Although not all elements of that architecture are incorporated into robo-CAMAL, the results offer promise. The links to other architectures, for example CogAff [4], are obvious as the work on motivation in CAMAL arose from an earlier CogAff project [6]. CAMAL is different in many ways, and most noticeably in using the a-CRIBB BDI reasoning schema. CogAff is now also being used to control robots [53]. While we have yet to directly compare robo-CAMAL with these architectures using standard benchmarks, this is an issue that future work will address. However such considerations do raise some of the problems raised by Hawes et al. [53], and Hanks et al. [54] earlier, in that while single task comparisons allow direct comparison, the nature of research and implementations at different institutes will necessarily invoke differences.

Belavkin in his analysis of emotion considers valence and arousal to be useful in designing a cognitive model [55]. While the metrics he uses in his ACT-R related system are based on probability and entropy, there are similarities in the use of these two concepts. CAMAL uses affective valence to select and modify goals, associations and motivations. Arousal relates to goals, and in CAMAL high goal importance relates to elevated arousal, and the achievement of goals leads to the strengthening of the valences linking goals, belief basis and intended behaviour. CAMAL however, is not a rule based system adapted to use affect, or in the case of Belavkin, ACT-R adapted to use emotion. ACT-R [56] and SOAR [57] are rational models of cognition adapted to incorporate emotion (or affect). Indeed a central tenet of CAMAL, and indeed CogAff, is that motivation is the core of the cognitive model, and cannot be a (simple) bolt-on mechanism.

What these, and further experiments, see [11], demonstrate is that motivation in robo-CAMAL is grounded and deliberative. The only possible doubt is robo-CAMAL's formation of expectancies. It could be argued that as the agent requires these expectancies to be pre-defined via the domain model, its motivation is not fully deliberative. However, due to the way in which robo-CAMAL adheres to the individuation and affective response criteria, it is clear that its behaviour is far more deliberative than reactive.

Of course, other architectures have addressed the research issues presented in this paper. Most pertinent to the overall aims of this research, is the mobile robotics work of Stoytchev and Arkin [17] in that their architecture combines three components: deliberative planning, reactive control, and motivational drives. They identify that the mapping of high-level deliberative commands onto a reactive controller solves some of the problems associated with purely reactive control, but is equally difficult to solve. The approach taken in robo-CAMAL (and other CAMAL variants) offers a solution, at least in the limited experiments performed to date. The BDI schema used with the association mappings offers a generic solution that can be tailored to specific environments and tasks, through adaptation and learning of the domain model. The motivational model used in CAMAL is developed from a goal based model of emotion and works at both the deliberative and reactive level. The Arkin research uses a physiological model with behavioural triggers at the reactive level. Both map onto motivational variables and provide valid motivational and indeed complimentary models; in other research we have used low level models in fungus eater experiments in conjunction deliberative motivation management [31, 48]. Both approaches fit the criteria for motivation model s as defined by Epstein [45]. However, the Arkin model has problems in resolving conflicts between the internal motivations and goals of the robot and the goals that people set for the robot; the conflict resolution and predicate management across the CAMAL BDI schema suffers no such problem. We argue that the approach taken in CAMAL, and CogAff, provide for more flexible approaches where the architecture needs to reconfigure and optimise its tasks to goals that are achievable in the environment within which the robot finds itself. In the CAMAL architectures the distributed model of motivation (and affect) includes the confidence the agent has in belies, the importance and threat values of goals, the insistence values of the associations and the persistence of motivations.

The robo-CAMAL implementation described here uses a naive version of reinforcement learning to adapt association (as shown in the third set of experiments). Once associations have been generated (or pre-defined), it can optimise the association insistence value over time to produce near-optimal configurations of the reactive-subsystems for specific belief-goal combinations. The affect model is also used to manage goals throughout run-time. Konidaris and Barto [58] make the point that a motivational system is central to agent autonomy. Like Stoytchev and Arkin [17], they use a low level (quantitative) drive system which is used to provide metrics for a reinforcement learning system in a simulated agent in Spier's domain [59]. In should be noted that Spier's work is heavily related to that of McFarland [14]. Our work draws on McFarland at the theoretical and experimental level in the CAMAL research and related studies, e.g. [31, 48]. While their work has a more rigorous learning mechanism, a direct comparison is not valid as Konidaris and Barto use a simulated agent. However, a direct comparison with Konidaris and Barto [58] using Spier's domain, should be possible using the full CAMAL architecture but is left for future work.

Rodriguez et al. report on the use of a motivational system for a robot that combines reinforcement learning and a genetic algorithm [60]. They show promising results for a robot standard wall following problem, but no quantitative comparison with the current work is possible as metrics for this problem are not supplied. The find(redrobot) experiment is qualitatively similar in that it involves finding and tracking an object in a robot environment. However, the motivational system used by Rodriguez et al. is not well developed, and we suggest the motivational system and BDI schema presented in this paper is more widely adaptable to tasks and environments, even if the learning mechanism is not, as yet, fully developed. Other benchmark comparisons, such as the Keepaway Task used by Whiteson et al. [61] in their experiments with different learning mechanisms for action selection, are possible. The comparisons of Whiteson et al. are interesting in that they show that different learning mechanisms offer differing levels of performance across different task categories. A deeper design and implementation of CAMAL may capture these task defined optimality issues using the meta-deliberative Norm constructs which define operational constraints on the currently extant architecture.

7. Conclusion

The robo-CAMAL architecture was designed from the reductionalist viewpoint, and conceptualised using the theory of mind as a control system. This means that the robo-CAMAL architecture can be decomposed into separate functional components, and was designed to control the actions of a mobile robot. The architecture makes use of a number of different subsystems that have been developed over many related projects.

At a more general level, robo-CAMAL makes use of a hybrid, reactive-deliberative architecture. It uses this set up in order to make use of a reactive system's ability to respond and act quickly in a rapidly changing environment. It can also use the deliberative system's ability to reason about events within its environment, and solve problems.

The robo-CAMAL architecture controls and directs the actions of a mobile robot through the use of a BDI schema, a motivational blackboard, and motivational control states. This research has addressed the argument that to be grounded, motivation must be an interactive process between the agent's actions and its environment [42]. In regards to this argument it is clear that the interactive nature of robo-CAMAL's motivation that it is grounded in its environment. In addition, based on the indices of motivation model [45], the motivational control states used by robo-CAMAL are predominantly deliberative in nature.

The learning experiments demonstrate that robo-CAMAL has the ability to anchor events and object to pre-defined symbols. It uses the domain model to recognise an object such as a blueball, and an event such as hit(blueball). While the learning mechanism is currently naive, current and future research will have to address whether the learning process of all associations for all tasks, for more complex architectures and sets of behaviours, is scalable? Current research is addressing this through formalising affect and learning model using Bayesian Reasoning concepts - it is an open question. Again further experimentation with the full CAMAL architecture in synthetic worlds and with robo-CAMAL may provide stronger evidence to substantiate the qualitative comparisons made here.

The agent's performance can be improved in two main ways. The first involves a more sophisticated domain and perceptual model, to enable (perceptual) learning about new objects in its environment. This, however, will raise new issues relating to symbol grounding and the generation of "artificial" symbols. Following on from a long-standing analysis of the work of Barsalou [62, 63], new research using robo-CAMAL is looking to neural learning mechanisms. This, in turn, will require motivation to be mapped to new interpretations of control states, possibly at a meta-cognitive level as in the research performed by Venkatamuni [48]. CAMAL will then enabled to adapt to unfamiliar and less structured environments by swapping attitudes, embodied as norms at a meta-cognitive layer above any of the BDI or reasoning schema in Figure 3. The second improvement involves the addition of a meta-cognitive layer to manage the agent's goals and attitudes, as well as a variable domain model, for example changing the agent's attitude from a free roam to a learning mode.

In conclusion, robo-CAMAL has demonstrated an ability to anchor symbols to perceptual data with the use of a domain model. This anchoring mechanism performs very well in a controlled environment, but currently struggles in a more unstructured environment. This failure is partly due to a lack of sophistication in the domain model, and the relatively shallow learning model. This highlights just how important the anchoring process is to a deliberative agent if it is to function in its environment. Current work on the robotic embodiment of the CAMAL architecture looks to deepen and improve the link between the deliberative and reactive architecture through extending the BDI constructs with Bayesian reasoning; and to address the issues related to perceptual learning using mechanisms similar to those proposed by Barsalou [62, 63].

References

[1] C. Stachniss and W. Burgard, Mobile robot mapping and localization in non-static environments. In Proceedings of the Twentieth National Conference on Artificial Intelligence, pages 1324-1329, 2005.

[2] D. Wolf and S. Sukhatme, Mobile robot simultaneous localization and mapping in dynamic environments. Autonomous Robots, 19(1): 53-65, 2005.

[3] M. Fiala and A. Basu, Robot navigation using panoramic tracking. Pattern Recognition, 37(11):2195-2215, 2004.

[4] A. Sloman, The mind as a control system. In Hookway, C. and Peterson, D. editors, Philosophy and Cognitive Science. Cambridge University Press, 1993.

[5] A. Stoytchev, Some Basic Principles of Developmental Robotics, IEEE Transactions on Autonomous Mental Development, 1(2): 122-130, 2009.

[6] D.N. Davis, Reactive and Motivational Agents, Agent Theory, Architecture and Language (ECAI-96), Budapest, 1996.

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

[8] D.N. Davis, Linking perception and action through motivation and affect, Journal of Experimental and Theoretical Artificial Intelligence, 20(1):37-60, March 2008.

[9] D.N. Davis and M.V. Vijayakumar, A "Society of Mind" Cognitive Architecture based on the Principles of Artificial Economics. International Journal of Artificial Life Research. 1(1): 51-71, 2010.

[10] B. Subagdja, L. Sonenberg and I. Rahwan, Intentional learning agent architecture, Autonomous Agents and Multi-Agent Systems, 18(3), June, 2009.

[11] J. Gwatkin, robo-CAMAL: Anchoring in a Cognitive Robot, PhD Thesis, Department of Computer Science, University of Hull, 2009.

[12] R. Arkin, Integrating behavioural, perceptual and world knowledge in reactive navigation. Robotics and Autonomous Systems, 6:105-122, 1990.

[13] R. Beck, Motivation: Theories and Principles. Pearson Education, 4/e, 2000.

[14] D. McFarland, What it means for robot behaviour to be adaptive. In J Meyer and S Wilson, editors, From animals to animats: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour. MIT Press, 1991.

[15] D. Westen, Psychology: Mind, Brain, and Culture. John Wiley and Sons, Inc. 1996.

[16] R. Manzotti and V. Tagliasco, From behaviour-based robots to motivation-based robots, Robotics and Autonomous Systems, 51 (2), p.175-190, May 2005.

[17] A. Stoytchev and R. Arkin, Incorporating Motivation in a Hybrid Robot Architecture, Journal of Advanced Computational Intelligence and Intelligent Informatics, 8(3): 269-274, May 2004.

[18] S. Harnad, The symbol grounding problem. Physica D, 42:335, 346, 1990.

[19] S. Coradeschi and A. Saffiotti, An introduction to the anchoring problem. Robotics and Autonomous Systems, 43:85-96, 2003.

[20] W. Clancey, Situated Cognition. Cambridge University Press, 1997.

[21] A. Clark, Embodiement and the philosophy of mind. In A O'Hear, editor, Current Issues in Philosophy of Mind: Royal Institute of Philosophy Supplement 43. Cambridge University Press, 1998.

[22] R. Pfeifer and C. Scheier, Understanding Intelligence. MIT Press, 1999.

[23] E. Thelen, G. Schoner, C. Scheier and L. Smith, The dynamics of embodiment: A field theory of infant perservative reaching. Behavioral and Brain Sciences, 24:1-86, 2001.

[24] E. Alpaydin, Introduction to Machine Learning 2/e. MIT Press, 2010.

[25] R. Sutton and A. Barto, Reinforcement Learning. MIT Press, 1998.

[26] E. Sontag, Mathematical control theory: Deterministic finite dimensional systems. Springer, New York, second edition, 1998.

[27] G. Bourgne, Affect-based multi agent architecture (for a 5-aside football simulation). MSc thesis, School of Computer Science, University of Hull, 2003.

[28] M. Georgeff, B. Pell, M. Pollack, M. Tambe and M. Wooldridge, The belief-desire-intention model of agency. In J. Muller, M. Singh, and A. Rao, Eds., Intelligent Agents V: Agent Theories, Architectures and Languages, pp. 1-10, Springer-Verlag, 1999.

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

[30] S. Wahl and H. Spada, Children's reasoning about intentions, beliefs and behaviour. Cognitive Science Quarterly, 1:5-34, 2000.

[31] S. Lewis, Computational Models of Emotion and Affect. PhD thesis, School of Computer Science, University of Hull, 2004.

[32] D.N. Davis, Affect and Affordance: Architectures without Emotion, AAAI 2004, Stanford University, California, USA.

[33] R. Picard, Affective Computing. MIT Press, 1998.

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

[35] J. Tooby and L. Cosmides, The past explains the present: emotional adaptations and the structure of ancestral environments. Ethology and Sociobiology. 11: 375-424. 1990.

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

[37] K. Scherer, Appraisal Considered as a Process of Multilevel Se-

quential Checking. In K. Scherer, A. Schorr, T. Johnstone (Eds.), Appraisal Processes in Emotion. Oxford University Press. New York, 2001.
[38] K. Oatley, Best Laid Schemes. Cambridge University Press, 1992.
[39] T-T. Teoh, Y-Y. Nguwi and S-Y. Cho, Towards a portable intelligent facial expression recognizer, Intelligent Decision Technologies, 3(3): 181-191, 2009.
[40] A. Zhu and Q. Luo, Study on Speech Emotion Recognition System in E-Learning. In: Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2007.
[41] D. Corkill, Blackboard systems. AI Expert, 6(9):40-47, 1991.
[42] T. Savage, The grounding of motivation in artificial animals: Indices of motivational behavior. Cognitive Systems Research, 4:23-55, 2003.
[43] N. Sharkey and J. Heemskerk, The neural mind and the robot. In Brown, A. editor, Neural Network Perspectives on Cognition and Adaptive Robotics. IOP Press, 1997.
[44] N. Sharkey and T. Ziemke, Mechanistic verses phenomenal embodiment: Can robot embodiment lead to strong AI? Cognitive Systems Research, 2:251-262, 2001.
[45] A. Epstein, Instinct and motivation as explanations for complex behavior. In Pfaff, D. editor, The Physiological Mechanisms of Motivation. Springer, 1982.
[46] V. Braitenberg, Vehicles, Experiments in Synthetic Psychology. MIT Press, 1984.
[47] R. Brooks, Intelligence without reason. In Proceedings of the International Joint Conference on Artificial Intelligence, 1:569-595, 1991.
[48] V.M. Venkatamuni, A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture, PhD Thesis, Computer Science, University of Hull, August 2008.
[49] M. Bratman, Intentions, Plans and Practical Reason. Harvard University Press, 1987.
[50] T. Phung, M. Winikoff and L. Padgham, Learning within the BDI Framework: An Empirical Analysis, In: Lecture Notes in Computer Science, Volume 3683 (Knowledge-Based Intelligent Information and Engineering Systems), Springer Berlin/Heidelberg, 2005.
[51] B. Subagdja and L. Sonenberg, Learning Plans with Patterns of Actions in Bounded-Rational Agents, In: R. Khosla et al. (Eds.): KES 2005, LNAI 3683, pp. 30-36, 2005.
[52] M. Lekavy and P. Navrat, Expressivity of STRIPS-Like and HTN-Like Planning. LNAI Vol. 4496 Agent and multi-agent Systems. Technologies and applications. 1st KES International Symposium, KES-AMSTA 2007, Wroclaw, Poland, May/June 2007. - Germany, Springer-Verlag Berlin Heidelberg, 2007. pp. 121-130.
[53] N. Hawes, J. Wyatt and A. Sloman, Exploring Design Space For An Integrated Intelligent System. Knowledge-Based Systems, 22(7), pages 509–515, Elsevier. Published as a best paper from Artificial Intelligence 2008 (AI-2008), The 28th SGAI International Conference on Artificial Intelligence. September 2009.
[54] S. Hanks, M. Pollack and P.R. Cohen , Benchmarks, Test-beds, Controlled Experimentation, and the Design of Agent Architectures, AI Magazine, 14(4):17-42, 1993.
[55] R.V. Belavkin, On the Relation between Emotion and Entropy, In C. Johnson (Ed.), Proceedings of the AISB'04 Symposium: Emotion, Cognition and Affective Computing, York, 2004.
[56] J.R. Anderson and C. Lebiere, The Atomic Components Of Thought. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.
[57] A. Newell, Unified Theories of Cognition. Harvard University Press, Cambridge, Massachusetts, 1990.
[58] G. Konidaris and A. Barto, An Adaptive Robot Motivational System, From Animals to Animats 9, Lecture Notes in Computer Science, Springer Berlin / Heidelberg 1611-3349 Volume 4095 2006.
[59] E. Spier, From Reactive Behaviour to Adaptive Behaviour: Motivational Models for Behaviour in Animals and Robots. PhD thesis, Balliol College, University of Oxford, 1997.
[60] M. Rodriguez, R. Iglesias, C.V. Regueiro, J. Correa and S. Barro, Autonomous and fast robot learning through motivation, Robotics and Autonomous Systems, 55(9), p.735-740, Sep 2007.
[61] S. Whiteson, M.E. Taylor, P. Stone, Empirical Studies in Action Selection with Reinforcement Learning, Adaptive Behavior, 15(1), 2007.
[62] L.W. Barsalou, Perceptual symbol systems. Behavioral and Brain Sciences, 22, 577-609, 1999.
[63] L.W. Barsalou, Simulation, situated conceptualization, and prediction. Philosophical Transactions of the Royal Society of London: Biological Sciences, 364, 1281-1289, 2009.