THE UNIVERSITY OF HULL

CernoCAMAL: A Probabilistic Computational Cognitive Architecture

A thesis submitted for the degree of Doctor of Philosophy

by

HOSSEIN MIRI

30 September 2012

Thesis Advisor: Dr. D.N. Davis

Department of Computer Science, University of Hull, UK

Acknowledgements

Every Ph.D. thesis is, to some degree, a collaborative effort between the research student and the supervisor. I would like to thank my supervisor, Dr. D.N. Davis, for advising and supporting me over the past four years, and for giving me freedom to explore and discover new areas of Cognitive Robotics and Artificial Intelligence. Without him, this thesis would simply have not existed. He suggested the project and has continually supplied me with insights. I would like to acknowledge the overwhelming contributions and endless hours invested in me by him. His tremendous service to this thesis through careful reading and checking of the drafts – to ensure that it meets his high standards of scholarship and accuracy – deserves unreserved gratitude. It is for all of these reasons that I extend to him a heartfelt *"thank you ..."*

I also wish to thank the members of my supervisory committee: Dr. C. Kambhampati and Prof. Y.I. Papadopoulos – for their help in reviewing my progress, the numerous inspiring discussions we have had, and for giving valuable comments in writing and reading of the drafts of my thesis. Many thanks to the members of the secretarial team and administrative staff for all their help in whatever matter during my stay here as a Ph.D. Research Student and Teaching Assistant. I am very thankful to them, especially because most of their work facilitates the smooth running of the Department, but unfortunately goes unnoticed, unrecognised, and unrewarded.

Other members of staff have also been very supportive and helpful. Dr. Helen Wright proved, on separate occasions, to be a meticulous and thorough reviewer, and brought a useful outsider's perspective to my first paper, and encouraged me to make it more accessible to a wider range of readers – although any failings in this regard are of course my fault. It is impossible to list everyone who has so graciously given me the benefit of their time and expertise. Dr. Mike Brayshaw, Mr. Brian Thompson, Mr. Darren McKie, Mr. Mike Bielby, Mr. Adam Hird, Mr. Mark Bell, Mrs. Liz Sandland, Mrs. Emma-Jane Alexander, and Mrs. Amanda Milson – they have all been exceptionally attentive to me . . .

Abstract

This thesis presents one possible way to develop a computational cognitive architecture, dubbed CernoCAMAL, that can be used to govern artificial minds probabilistically. The primary aim of the CernoCAMAL research project is to investigate how its predecessor architecture CAMAL can be extended to reason probabilistically about domain model objects through perception, and how the probability formalism can be integrated into its BDI (Belief-Desire-Intention) model to coalesce a number of mechanisms and processes.

The motivation and impetus for extending CAMAL and developing CernoCAMAL is the considerable evidence that probabilistic thinking and reasoning is linked to cognitive development and plays a role in cognitive functions, such as decision making and learning. This leads us to believe that a probabilistic reasoning capability is an essential part of human intelligence. Thus, it should be a vital part of any system that attempts to emulate human intelligence computationally.

The extensions and augmentations to CAMAL, which are the main contributions of the CernoCAMAL research project, are as follows:

- The integration of the EBS (Extended Belief Structure) that associates a probability value with every belief statement, in order to represent the degrees of belief numerically.
- The inclusion of the CPR (CernoCAMAL Probabilistic Reasoner) that reasons probabilistically over the goal- and task-oriented perceptual feedback generated by reactive sub-systems.
- The compatibility of the probabilistic BDI model with the affect and motivational models and affective and motivational valences used throughout CernoCAMAL.

A succession of experiments in simulation and robotic testbeds is carried out to demonstrate improvements and increased efficacy in CernoCAMAL's overall cognitive performance. A discussion and critical appraisal of the experimental results, together with a summary, a number of potential future research directions, and some closing remarks conclude the thesis.

Table of Contents

List of Figures

List of Equations and Tables

List of Abbreviations and Acronyms

a-CRIBB	affective CRIBB (see CRIBB below)
ACT-R	Adaptive Control of Thought – Rational
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ARIA	Advanced Robotic Interface for Applications
BBAI	Behaviour Based Artificial Intelligence
BDI	Belief Desire Intention
CAMAL	Computational Architecture for Motivation, Affect, and Learning
CogAff	Cognition Affect (project)
CPR	CernoCAMAL Probabilistic Reasoner
CRIBB	Children's Reasoning about Intentions, Beliefs, and Behaviour
DegBel	Degree of Belief
EBS	Extended Belief Structure
GC5	Grand Challenge 5
P3DX	Pioneer 3DX (robot)
SOAR	State, Operator, And Result
TOK	Theory of Knowledge
UTC	Unified Theory of Cognition

1 Introduction

This chapter begins with a brief description of the *problem* tackled in this thesis. After setting the scene and defining the context, a description of a potential *solution* proposed to overcome this problem is presented. The research issues concerning the current cognitive architecture under investigation – CernoCAMAL – are then highlighted, followed by the objectives of the research described here. It is also pointed out how the proposed solution and its consequent augmentation have led to the development of a probabilistic reasoner for CernoCAMAL that can deliberate probabilistically over the generated perceptual feedback. The chapter concludes with an outline of the structure of the thesis.

1.1 Problem Statement

There is considerable evidence that probabilistic thinking and reasoning is linked to cognitive development and plays a role in cognitive functions, such as decision making and learning (e.g. Piaget 1928; Piaget, Inhelder 1951; Fischbein 1975; Shaughnessy 1981; Green 1983; Peard 1995; J.Truran 1996; K.Truran 1996; Fischbein, Schnarch 1997; Way 2003; Oaksford, Chater 2007, 2009). This leads us to believe that a probabilistic reasoning capability is an essential part of human intelligence. Thus, it should be a vital part of any system that attempts to emulate human intelligence computationally. In other words, probabilistic reasoning is an essential aspect of the process of cognition and therefore must be considered in any adequate description of it, such as a computational cognitive architecture.

The problem is that CAMAL has never addressed or included any probabilistic reasoning capability. As a multi-tier cognitive architecture, however, it must incorporate this fundamental cognitive capability. CAMAL uses a variant of the a-CRIBB reasoning model (Davis, Lewis 2003, 2004) i.e. a BDI model and an affect model, plus a motivational blackboard. At the deliberative level, affective values and affordances can be associated with processes and predicates, and then relayed as control signals to instantiate and modify aspects of motivators and their associated representations and behaviours (Davis 2010).

One of the limitations of the BDI model, however, is the lack of any explicit mechanism to express *degrees of belief*. The belief statements used by CAMAL's BDI model are represented by clauses of the form:

```
belief ( Descriptor, Source, Time ).
```

This particular syntax represents beliefs as *categorical* states. Therefore, they cannot be adequately valenced via affective values and affordances, in line with the affect and motivational models. Given that the current CAMAL research presents an affect- and affordance-based core for mind, it seems reasonable to conjecture that beliefs, too, should be grounded in the use of affect with the aim to be consistent across different domains, tasks, and levels of processing. The belief structures are vital to this work – hence the early introduction of the belief predicate above – but they are merely an artefact of the implementation that demonstrate the principles argued for in the thesis.

1.2 Proposed Solution

It is proposed to represent CernoCAMAL's belief statements as *graded* states, using probability formalism. Put differently, it is proposed to use an Extended Belief Structure (EBS) to represent the degrees of beliefs numerically and then manipulate them. The belief statements used by CernoCAMAL's BDI model are represented by clauses of the form:

```
belief ( Descriptor, Source, Time, DegBel ).
```

A probabilistic reasoning capability will be incorporated in CernoCAMAL using the proposed belief predicate. The EBS associates a probability value `DegBel` with every belief statement in CernoCAMAL which defines the degree to which the belief statement is believed to be true. This will enable the computation of changing degrees of belief given apriori, and subsequently using the BDI, affect, and motivational models to determine the agent's intentions, actions, or behaviours. In other words, it will allow the entire BDI model to run using numeric affective values and affordances to prioritize choices over the current belief set.

The proposed EBS and consequently the BDI model will be compatible with the way in which the affect and motivational models operate throughout CernoCAMAL: having an associated affective magnitude that can fluctuate according to success or failure associated with that element. Put differently, affect will serve as a *decision metric* and affective values as a *currency* across the entire architecture, including beliefs and the BDI model. This way, belief structures, too, will be grounded in the use of affect with the aim to be consistent across different domains, tasks, and levels of processing.

The proposal to incorporate the EBS has also led to the development of the CernoCAMAL Probabilistic Reasoner (CPR) that can deliberate probabilistically over the generated perceptual feedback: feedback generated by reactive sub-systems. The CPR consistently reasons about the domain model objects and their instances, and keeps CernoCAMAL's model of its surroundings up to date.

1.3 Research Questions

In general, this project builds on research into computational models of cognition (Davis 1996…2010) and how affect and motivation can be used to improve reasoning about perceptions and belief formations (Davis 1996…2010). It also applies a development of these models, used in simulated and real environments (i.e. virtual and physical testbeds) to the control of a recently-built cognitive mobile robot – RoboCAMAL (Gwatkin 2009).

There clearly exists a need to address and incorporate probabilistic reasoning and inference in CAMAL. The primary aim of the CernoCAMAL research project is exactly to tackle this need with the formal probability theory. This research therefore attempts to address the following specific research questions in the current cognitive architecture under investigation – CernoCAMAL:

- Can CernoCAMAL reason probabilistically by exploiting the proposed EBS? Can the integration of the proposed EBS facilitate probabilistic reasoning and inference in CernoCAMAL?
- Can the BDI model run compatibly with the affect and motivational models, and affective and motivational valences used throughout the whole architecture? Can this ensure a consistent and systematic metric across all aspects of affect, reasoning, and domain model management?
- Can the probabilistic deliberation results of the CPR be used for computing changing degrees of belief given apriori, and subsequently using the BDI, affect, and motivational models to determine the agent's intentions, actions, and behaviours?
- Can the CernoCAMAL cognitive architecture be applied to virtual and physical cognitive agents using synthetic testbeds and mobile robots?

1.4 Thesis Structure

The remainder of this thesis is divided as follows: Chapter 2 presents some necessary background knowledge pertaining to this work. Chapter 3 reviews the original overarching CAMAL architecture as an important pivotal axis for the development of CernoCAMAL. Chapter 4 lays out the motivation and impetus for the development of CernoCAMAL followed by an introduction to its framework and main components, namely EBS and CPR. Chapter 5 introduces the testbeds, followed by an overview to the different types of controlled experimentation carried out in these environments. Chapter 6 presents a succession of experiments in the simulation and robotic testbeds, followed by the experiments' outcomes. Chapter 7 analyses the obtained experimental results, followed by a discussion over them. Chapter 8 concludes the thesis with a summary, a number of potential future research directions, and some closing remarks.

2 Essential Background

This chapter begins with a brief introduction to the fields of Cognitive Science and Artificial Intelligence. Two major approaches to the study of these two inseparable areas of science are pointed out: Symbolism and Connectionism. The concepts of Cognitive Modelling and Cognitive Architectures are explained, and commonly used types of cognitive architectures are highlighted: Reactive, Deliberative, and Hybrid. The notion of Theory of Mind is introduced, and a flavour of the familiar phenomena of Emotion and Motivation is presented. This is followed by some background knowledge in Autonomous Cognitive Agents and Mobile Robots research. Three major approaches to the study of this area of research, and the possibility of achieving synthetic intelligence in autonomous cognitive agents and mobile robots are presented: Physical Grounding hypothesis; Perceptual Symbol System theory; and Control System approach. The latter – as the cornerstone of this work – is discussed in detail, followed by an introduction to the Control State theory. The chapter concludes with presenting the notion of Affective and Motivational Control States and how they are used in developing cognitive architectures.

2.1 Cognitive Science and Artificial Intelligence

Cognitive Science is the study of cognition and intelligence, and their computational processes in humans, animals, computers, and in the abstract (Kaplan, Simon 1989) with contributors from various fields, including Neuroscience, Psychology, Philosophy, and Computer Science. It is a broad field that covers a wide array of topics related to cognition. It is, essentially, the study of cognitive processes and how they are integrated to form a mind (Franklin 1995).

One of the main topics that Cognitive Science is concerned with is Artificial Intelligence (AI). AI involves the study of cognition, cognitive phenomena, and intelligence in machines, with an ultimate goal of producing working models with human-like or animal-like mental properties and capabilities, whether in order to provide detailed scientific explanations for aspects of human or animal cognition, or in order to solve real-world problems (Franklin 1995).

Nearly thirty years ago, Donald Norman (1980) set an agenda of important topics for Cognitive Science. He argued that there are at least twelve issues that should comprise this agenda: consciousness; perception; memory; language; thought; belief; emotion; interaction; performance; learning; skill; and development. This long, though non-exhaustive, list has played an important role in determining where researchers have focused their work in AI.

But there exists a number of concerns: whether there is only one correct and unique way of integrating cognitive processes to build an artificial mind; whether there are any common principles with which an artificial mind can be built; whether there is any need for a set of general assumptions for constructing a cognitive model of mind. The eminent *Unified Theories of Cognition* (Newell 1990) attempts to address these concerns. It defines a Unified Theory of Cognition (UTC) as a single set of mechanisms and processes for all cognitive behaviour[1]. It argues that we need a set of general assumptions for building an artificial mind. It also asserts that any UTC must explain several key issues, such as how to represent knowledge, how to define goals, etc.

This view of integrating various cognitive components and building an artificial mind is a prevalent view in Cognitive Science and Artificial Intelligence. It implies that mind can be decomposed into separate modules, as in distinct processing units, such as a unit for handling memory, a module for handling visual data, a centre for emotion processing, etc. This view is often referred to as the *modularity of mind* (Fodor 1983). This view is strongly opposed by the *multimodality of mind* view that denies the existence of such separate modules (Barsalou 1999). As a matter of fact, multimodal integration has been found in many different locations in the brain (e.g. Gallese, Lakoff 2005). Sensory modalities like touch, vision, hearing, and so on are actually integrated with each other and with motor control and planning. All this appears to suggest that there are no distinct regions whose only job is to link supposedly separate brain areas for distinct sensory modalities (Gallese, Lakoff 2005).

[1] Behaviours are directed to the achievement of goals, whether explicit or implicit (Toates1998). They are actions performed on an environment. The most primitive type is a reflex. More sophisticated type on an environment may be sub-systems built to achieve specific tasks, e.g. reactive sub-architectures.

2.2 Symbolism and Connectionism

In mid 1950s when access to digital computers had just become possible, AI research began to explore the possibility that human intelligence could be reduced to symbol manipulation. Essentially, symbolic AI systems manipulate symbols instead of numbers. Furthermore, these systems operate under a set of rules, and their actions are determined by these rules. So, their general form is a symbol-processing program consisting of rules of some kind, stored in some sort of memory, along with some appropriate data structures. The rules, then, operate on the data structures producing some impression of intelligence. John Haugeland coined the term *good old fashioned AI* for these symbolic approaches (Haugeland 1985).

Much of the work done in symbolism (symbolic AI) is based on the Physical Symbol System hypothesis. The hypothesis states that a physical symbol system, such as a digital computer, has the necessary and sufficient means for general intelligent action (Newell, Simon 1976). The hypothesis identifies a class of systems as embodying the essential nature of symbols, and as being the necessary and sufficient condition for a generally intelligent agent (Newell 1982). The hypothesis implies that computers, when we provide them with the appropriate symbol-processing programs, will be capable of intelligent action. It also implies, as Newell and Simon wrote, that the symbolic behaviour of man arises because he has the 'characteristics' of a physical symbol system.

A physical symbol system consists of a set of entities, called *symbols*, which are physical patterns that can occur as components of another type of entity, called an expression or symbol structure (Newell, Simon 1972). Thus, an expression or symbol structure is composed of a number of symbols, related to each other in some way. This symbolic structure can be realized in a physical system, such as a digital computer. The adjective *physical* denotes an important feature that such a system can be realized by engineering together physical components. The 'characteristics' Newell and Simon referred to – highlighted in the previous paragraph – should be clear now.

A good example of a physical symbol system is a production system (Newell, Simon 1976). A production system consists of a collection of rules (production rules), a set of conditions and facts that reside in a working memory, and a series of search mechanisms. These search mechanisms comprise a control system that infers new facts from the existing stored facts. A production system usually takes the simple form of `if(condition), then(action)` which means that if the conditions or facts in the working memory are true, then the actions or behaviours are carried out and, possibly, new facts are generated.

There is another popular approach known as connectionism (connectionist AI). Artificial Neural Networks (ANNs) are well-known and widely-used example of connectionist models. Connectionist models are usually networks of large numbers of simple, but highly interconnected units, running in parallel. They are simplified models of the brain, and their constituent units (artificial neurons) are the analogues of biological neurons, together with weights that measure the strength of connections between the units. The behaviour of the network as a whole is a function of the initial state(s) of activation of the units and of the weights on its connections (Haykin 1994).

The connectionist approach takes the idea that intelligence arises from the interactions of such highly-interconnected neurons (Clark 1989). Information from one neuron flows to another neuron across a synapse. Electrical signals carry information from one end of a neuron to the other, while a chemical reaction is responsible for passing signals from one neuron to another through their interconnected dendrites. ANNs are not programmed specifically to solve problems – they are trained. In other words, they are not programmed as such, but learn how to perform a task. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well.

To summarize [2] symbolic AI people are advocating a *computer* model of mind and connectionist AI people are arguing for a *brain* model of mind. To paraphrase Franklin (1995):

[2] There are a number of common approaches to modelling the human mind and cognition, such as Symbolic, Connectionist, and Probabilistic. The latter is the cornerstone approach adopted by this work. See 4.2 for a detailed discussion.

- Symbolism constructs its model of mind using *computation* as metaphor – mind is like a digital computer, processing a symbolic language; so mental activity is like the execution of a stored symbolic program.
- Connectionism bases its model of mind on a *nervous system* metaphor – mental activity is like the settling of a neuronal network into a stable configuration.

The notion that intelligence requires the use and manipulation of symbols has been extremely influential in AI. Several leading minds in AI research maintain this computational approach, asserting that the kind of computing that is relevant to understanding cognition involves various operations on symbols (e.g. Newell 1980, 1982, 1990; Fodor 1976, 1987; Pylyshyn 1987, 1989; Simon 1967, 1996, 1999). In contrast, connectionism proposes to design systems that can exhibit intelligent behaviour without storing, retrieving, and operating on structured symbolic expressions (e.g. Clark 1989). This method has grown very popular, particularly among those who believe that cognition can only be understood if we study it as Neuroscience (e.g. Arbib 1975). As described by Fodor and Pylyshyn (1988), almost everyone who was discontent with the information processing models of mind and symbolism, rushed to embrace the connectionist alternative!

2.3 Cognitive Modelling and Cognitive Architectures

There are two widely-used, though ill-defined, terms throughout AI literature: *model* and *architecture*. A model is an abstract representation of a system from a particular point of view. It is an artificial system (or a design for one or a theory of one) that behaves in almost the same way as a natural system. We build models to simulate a natural phenomenon in a useful manner, in order to perform experiments with them. A cognitive model is an abstract representation of human or animal cognition (predominantly human). It is an approximation to cognitive processes and is intended to be an explanation of how some aspect of cognition is accomplished by a set of primitive biological and computational processes.

Architecture means structure. A computational architecture shows various components and the control flow through them. It is some level of abstraction of some functional system that provides functional differentiation into interacting components (Sloman 1993). A cognitive architecture refers to the design and organization of the mind and cognition. It shows various components and the control flow through those components. It also specifies the underlying mechanisms and processes of a cognitive system. Essentially, it provides a working model of some set of cognitive phenomena.

A cognitive architecture could also be defined as an embodiment of a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task (Ritter, Young 2001). Put simply, a cognitive architecture is a theory for simulating and understanding some aspect of cognition. Some well-known examples of cognitive architectures are ACT-R, SOAR, and CogAff (see 7.2). Cognitive architectures are designed and implemented to be capable of performing certain behaviours and functions based on our understanding of human (and sometimes animal) minds. A model of a task constructed in a cognitive architecture can be run to produce a sequence of actions or behaviours. These sequences can then be compared with the sequences produced by human users, to assess the quality of a particular model.

Cognitive modelling refers to the investigation of mind and cognition through developing cognitive architectures, in order to simulate cognitive processes (Anderson 1990). It must be noted that all modelling is context-dependent; i.e. a model has a scope defined as the set of circumstances under which the model works. This scope is related to the concepts within which the model was developed and validated (Edmunds 2003). Attempts to construct cognitive models have been considerably assisted by the availability of appropriate languages for specifying and implementing them, such as Prolog. This thesis deals almost entirely with symbolic AI which, consequently, necessitates the need for explicit representation of beliefs, goals, actions, etc. One of the three software programming languages adopted for the purposes of this work is indeed Prolog that facilitates the convenient expression of symbols and symbolic structures (see appendix II).

In 1993 Professor Aaron Sloman put this conjecture forward: "*In order to account for the main characteristics of the human mind, it is more important to get the architecture right than to get the mechanisms right. Architecture dominates mechanism.*" (Mind as a Control System, Sloman 1993, p.1). He also suggested that we should try to characterize suitable classes of mechanisms and processes at the highest level of generality, and then expand with as much detail as is needed for our purposes. This is referring to broad and shallow architectures (Reilly, Bates 1993) that we should aim to start with. That is, the architectures should accommodate and integrate a wide range of general functions, such as vision, motivation, emotion, various kinds of action, various kinds of problem-solving, etc. This contrasts with deep and narrow systems, e.g. systems to analyse images, or understand sentences, etc. It may be necessary for a while to tolerate relatively shallow and simplified components as we explore the problems of putting lots of different components together. Later, we can gradually add depth and realism to the systems we build (Sloman 1993; Reilly, Bates 1993).

As mentioned earlier, Norman set an agenda of important topics for Cognitive Science. An important aspect of Norman's agenda is actually the use of an *architectural* perspective (Norman 1980). One such example architecture based on his view could identify five interactive processes: the reception of incoming signals (perception), the generation of output (action or intention or behaviour), a reactive or regulatory system, a deliberative or cognitive system, and an emotional or affective system. He suggested that this is the type of architecture needed to address the twelve issues included in his agenda.

A brief overview of the main types of architectures used in designing and developing autonomous cognitive agents and mobile robots is presented next.

2.3.1 Reactive Architectures

Reactive architectures are a relatively simple way of generating behaviours that allow an agent to interact with a virtual or physical environment in real time. Reactive architectures rely on quick responses. They can, therefore, be fast which makes them efficient and suitable to dynamic (changing) environments. Reactive architectures are based on the assumption that intelligent behaviour can be generated without explicit symbolic representation or explicit symbolic reasoning. By linking responses in an agent's perceptual systems directly to its actuators (also termed effectors or motors), the need to create an internal representation of the environment is eliminated (Brooks 1991).

The classic or traditional approach to AI (symbolism) decomposes intelligence into functional information-processing modules, e.g. memory, vision, planning, learning, etc. Each individual module does not actually produce any behaviour – only when all the various modules are combined, will an agent produce behaviour. In contrast, the reactive approach decomposes intelligence into behaviour-generating modules; i.e. each module produces a distinct behaviour, independent of other modules. Thus, the overall behaviour is determined by the combined effects of individual units acting independently. This makes reactive systems more robust than those built based on the symbolic approach, since the modules are placed in parallel, with information from the sensors passed to all units. Therefore, if a module breaks down or is removed, information is still passed to all the other modules and they can still operate. This is the well-known *subsumption* architecture (Brooks 1986).

2.3.2 Deliberative Architectures

Deliberative architectures are those that reason about actions and events, take into consideration the outcomes of their intentions, and deliberate to build a set of purposeful behaviours in order to achieve a specific goal. Most deliberative architectures make use of symbols to represent their current state and the environment they are situated in, in order to reason about their actions and events and determine their output.

Deliberative architectures have some major problems though. A first problem is the Frame problem which is essentially the problem of how to keep an agent's model of its environment consistent and up to date (McCarthy, Hayes 1969; Pylyshyn 1987). A second problem is the Symbol Grounding problem, which is basically the problem of how symbols are created that represent objects and events in an agent's environment (Harnard 1990). These issues can be simply put as: how to translate the environment into an appropriate symbolic description and how to symbolically represent information about dynamic and complex environments, and all that in time for an agent to act properly (Davis 2000). These two problems are issues that most such architectures need to address (Coradeschi, Saffiotti 1999, 2003; Shanahan 2005). This type of architecture on its own usually lacks reactivity in real-time contexts.

2.3.3 Hybrid Architectures

As the name suggests, this approach combines reactive and deliberative processes, to gain the advantages of both types. The processes are used asynchronously which allows both quick responses and planned behaviours based on reasoning. Hybrid architectures usually use a reactive component to interact with the environment, and a deliberative component to reason about the environment, events, objects, and actions. Behaviours such as obstacle-avoidance that are difficult to achieve using symbolic methods, consequently become trivial by using a reactive layer, as there is no need for the agent to create an internal model or internal representation of its environment.

Also, since the use of reactive components to interact with the environment filters out unnecessary information, only relevant information is passed to the deliberative component. This allows the deliberative processes to reason and plan in a more efficient way. In addition to the above two advantages, in some cases a deliberative architecture could suspend processing, to allow full reactive use (e.g. RoboCAMAL: Gwatkin 2009). These major reasons make hybrid architectures a sensible approach for developing autonomous cognitive agents and mobile robots.

2.4 Theory of Mind, Emotions, and Motivation

Franklin and Sloman have maintained for years, and still do, that the most useful way to look at a mind is as a *control system* for a cognitive agent. The continuing task of a mind, according to both thinkers, is to produce the agent's next action; i.e. to answer the only really significant question there is: what shall I do next (Franklin 1995). Any theory specifying how to go about answering this question is a Theory of Mind. A Theory of Mind is computationally plausible if it can be implemented or modelled on a computer. Put differently, an acceptable Theory of Mind is one that maps onto a computational design, and onto at least one implementation. If a Theory of Mind is to be implemented or modelled on a computer or agent, we must have in hand a computationally plausible cognitive architecture. If we have succeeded in implementing our cognitive architecture on a computer or agent (or better still a mobile robot with real sensing and acting capabilities and embedded in the real world!) so that it supports our Theory of Mind, we have produced an instantiation of an artificial mind or synthetic intelligence, based on that particular cognitive architecture.

Some researchers in the fields of Psychology and Cognitive Psychology hold a slightly different opinion of what a Theory of Mind actually is. It is common to see Theory of Mind referring to the ability to represent, conceptualize, infer, and reason about one's own or others' mental states from their experiences or their behaviours (Bartsch, Wellman 1989). This particular interpretation recognizes that the mind has the ability to attribute mental states (e.g. beliefs, desires, intentions) to oneself and others. Thus, a Theory of Mind in these fields refers to a process of generating inferences about the beliefs, desires, and intentions of others.

A specific area that has been intensively studied in Developmental Psychology under the heading Theory of Mind is the development of children's understanding of beliefs and intentions as representational entities. In this context, the Theory of Mind studies the development of children's competency in reasoning about mental states such as beliefs, desires, and intentions. This is a specific cognitive ability to understand others as intentional agents (Gopnik 1993).

The nature of emotions has been studied for a considerable time, with many contrasting theories and views having been formed (e.g. Darwin 1872; Duffy 1941; Camras 1992; Elliott 1992, 1994; Damasio 1994; Bates 1994; Bower 1994; Clore 1994; Ekman 1992…1999; Ekman et al. 1977…1994; Simon 1967…1999; Picard 1997…2010; Sloman 1993…2003; Clocksin 2004; Aube 2005; Davis 1996…2010; Davis, Lewis 2003, 2004). The majority of the theories developed can be largely examined in terms of two components:

- Emotions are *cognitive,* emphasizing their *mental* component.
- Emotions are *physical,* emphasizing their *bodily* component.

Research on the cognitive component focuses on understanding the situations that give rise to emotions, whereas research on the physical component emphasizes the physiological responses that co-occur with emotions, or rapidly follow them (Picard 1997). The distinction is not crucial to this particular research work. What is fundamental is the accumulating evidence that implicates the significant role of emotions in a variety of cognitive functions, such as reasoning, learning, perception, memory, and decision making (Duffy 1941; Camras 1992; Elliott 1992, 1994; Damasio 1994; Bates 1994; Bower 1994; Clore 1994; Ekman 1992…1999; Ekman et al. 1977…1994; Simon 1967…1999; Picard 1997…2010; Sloman 1993…2003; Aube 2005; Davis 1996…2010; Davis, Lewis 2003, 2004). This leads us to believe that emotions are an essential part of human intelligence. Thus, they should be a vital part of any system that attempts to emulate human intelligence computationally.

Further research proposes that motivation, too, plays a fundamental role in a variety of cognitive functions (e.g. Duffy 1941; Beck 2000; Westen 1996; Simon 1967, Davis 1996…2010; Sloman 1993…2003; Sloman, Logan 1999; Arkin 1998). In psychology, motivation is the driving force behind all actions of an organism. If it is taken that actions are performed to achieve a positive internal state or avoid a negative internal state, then motivation is the search for positive internal states and avoidance of negative internal states. According to leading researchers (e.g. Simon 1967; Izard 1991, 1993; Davis 1996, 2001, 2008, 2010; Sloman 2001, 2003) emotions and motivations serve as *filters* that guide perception and action, determine the input into the evaluation processes, and manipulate the evaluation of perceptual information.

In summary, emotional and motivational experiences are essential aspects of the process of cognition, and therefore must be considered in any adequate description of it. In common language, emotional and motivational states are feelings, moods, sentiments, attitudes, etc. Of course, each of these terms has a somewhat different usage. In the work ahead, however, the term 'affect' which is less semantically-overloaded is used instead of the term 'emotions' (Davis 2004). Affect can be considered a broader concept, and can be experienced as positive, negative, and neutral valences. Research on computational and robotic models of affect has been very active over the last decade. In her seminal book – *Affective Computing* – MIT Professor Rosalind Picard characterizes this new field that is forming in Computer Science and the scope of its research as: Computing that relates to, arises from, or deliberately influences emotions or other affective phenomena. She explains that this area is effectively the study of emotions in human-computer interactions, artificial intelligence, and cognitive architectures.

Affective Computing includes modelling and implementing emotions or other affective phenomena, and therefore can aid the development and testing of new and old emotion theories. Affective Computing also includes many other things, such as giving a computer the ability to recognize and express emotions, and developing its ability to respond intelligently to human emotions. An affective computer should have the skills of emotional intelligence, including an ability to manage its own affective mechanisms and processes, and to use them for improving its cognitive and rational functioning (Picard 1997). A fascinating example of a cognitive-affective machine is ShyBot, developed at the MIT Media Lab (Lee, Kim, Breazeal, Picard 2008). ShyBot is a personal mobile robot designed to both embody and elicit reflection on shyness behaviours of children living with autism. ShyBot can detect human presence and familiarity from face detection and proximity sensing, in order to characterize people as friends or strangers to interact with. It can also reflect elements of the anxious states of its human companion. The ShyBot project opened up an alternative way of considering human-machine interaction: deploying the use of cognitive-affective intelligence that machines can bring about. This led to a wide variety of cognitive models for Affective Computing to be proposed, including a-CRIBB (Davis, Lewis 2003, 2004).

The premises of Affective Computing still have not been universally accepted within the AI research community though. Despite its advantages, it is no surprise to see affect marginalized or sometimes even neglected in models of cognition constructed by computer scientists, since the developers of computers are largely biased towards the notion of 'thinking' rather than 'feeling' (Picard 1997). Another way to say this is that because intelligence can be seen as based on logical reasoning or abstract problem-solving, emotions are undesirable and can only lead to distraction or error! This view point, however, should be rectified. Affective experience appears to arise from elementary reactions of the central cognitive processes in the course of their construction. In this way, affect constitutes a primitive core in cognitive operations: It directs attentional processes; It influences the retrieval of memories; It modifies the structuring of thoughts; It influences the formation of perceptions; It may depress certain types of experience and it may heighten others (Blumenthal 1977). In a nutshell, emotions and affect constitute an ever-present substrate or foundation to everyday intelligent behaviours (Clocksin 2004).

2.5 Autonomous Cognitive Agents and Mobile Robots

The design of autonomous cognitive agents is a branch of Artificial Intelligence and, more latterly, Software Engineering. What constitutes an autonomous cognitive agent is a controversial topic though. In this text, autonomous cognitive agents are defined as follows: an autonomous cognitive agent senses its environment and acts upon it in the service of its agenda, e.g. humans, animals, mobile robots, software agents, etc. Thus, an autonomous cognitive agent is situated within some environment (e.g. our world) or an artificial environment within a computer (simulation world) or within an operating system or a database or even a network. The autonomous cognitive agent actively senses its environment and acts upon it, so as to further its goals and agenda. Therefore, the control of behaviour is vital to an autonomous agent (Franklin, Graesser 1996). Cognitive architectures are the backbone of autonomous cognitive agents.

Autonomous cognitive agents are capable of independent, purposeful, real-world actions, and respond to external stimuli in a timely fashion. They sense their environment through some sensors and act upon that environment through some motors (also referred to as actuators or effectors). An agent's perceptual input at a given instant is called a *percept* and the entire history of its perception is called *percept sequence*. Agents with human-like or animal-like cognitive capabilities have a single concern: what to do next? This concern needs to be addressed by producing an appropriate action or behaviour in response to the nature of the environment, the agent's agenda, and what are currently feasible and desired actions.

In humans, mind as a control system and main seat of natural intelligence balances all the mechanisms and processes, so that appropriate actions and behaviours are continually generated in a timely fashion. In cognitive agents, a similar control system is obviously needed, so that they can operate without the direct intervention of humans or other agents, and have some kind of control and autonomy over their actions and internal states (Franklin 1995).

In order for such an agent to have such control and autonomy over its actions and internal states, *knowledge and information* need to be stored and retrieved in an efficient way. Knowledge can be embedded in an agent either *structurally* or *symbolically*. ANNs are an example of *structurally-embedded* knowledge. The stimulus is related causally to the action it produces. So, the action depends both on the stimulus and on the internal state of the autonomous agent. With *symbolically-embedded* knowledge, in contrast, stimuli are first transformed into symbols, and then symbol manipulation leads to the action. Work on ANNs and symbolic systems has shown that structural embedding is often faster and more reactive, whereas symbolic embedding is often more flexible (Haykin 1994).

Numerous research groups across the globe are undertaking research using autonomous cognitive agents and mobile robots. There are several research issues that are being actively investigated. They pose important challenges to the fields of Cognitive Science and Artificial Intelligence, which effectively outline some major research activities, including:

- How to describe enough of the properties of an autonomous cognitive agent or mobile robot, its abilities, and its environment, to allow it to make high-level decisions on how to act or what to do next.
- How to store, manipulate, express, and retrieve the knowledge and information about the environment and the agent itself into its cognitive architecture.
- How to include concepts such as common-sense, free-will, and the like.

Below, are concepts that are areas of study in their own rights, but here they follow our discussion on the study of autonomous cognitive agents and mobile robots as three major approaches to investigating the possibility of achieving synthetic intelligence and artificial cognition.

2.6 Physical Grounding Hypothesis

In his controversial paper published in 1990, Professor Rodney Brooks of MIT criticized the Physical Symbol System hypothesis (symbolism). He claimed that this traditional approach to AI (symbolic AI) has emphasized the abstract symbol manipulation whose grounding in physical reality has rarely been achieved. He also argued that this classic approach of symbol manipulation is fundamentally flawed, and as such imposes severe limitations on the fitness of its progeny (Brooks 1990).

The core of Brooks' argument was that when intelligence is approached with strict reliance on interfacing to the real world through perception and action, reliance on symbolic representation disappears. In his later papers *Intelligence without Representation* and *Intelligence without Reason* published in 1991, he outlined a method to incrementally building complete intelligent creatures. It is noteworthy that by 'incrementally' he described the fact that each new module of an intelligent system, each new layer of its architecture, is built independently upon the previous module. Whereas the traditional methodology bases its decomposition of intelligence into functional information-processing modules whose combinations provide overall system behaviour, this methodology bases its decomposition of intelligence into individual behaviour-generating modules whose co-existence and co-operation let more complex behaviour emerge (Brooks 1991).

This approach, which was dubbed the Physical Grounding hypothesis, provided a different methodology for building intelligent systems than that pursued for the last few decades (back in 1990). Essentially, it states that to build a system that is intelligent, it is necessary to have its representations grounded in the physical world. Once this commitment is made, the need for traditional symbolic representations fades. The key observation to be made here, described eloquently by Brooks himself, is the fact that the world is its own best model – it is always exactly up to date and it always contains every detail there is to be known. The trick is to sense it appropriately and enough (Brooks 1990). Therefore, to build a system based on the Physical Grounding hypothesis, it is necessary to connect it to the world (ground it in the world) via a set of sensors and motors.

Physical Grounding hypothesis works well for reactive agents, as demonstrated in experiments carried out in the MIT AI Lab (1990, 1991). This is simply because reactive systems react directly to the world as it is sensed, avoiding the need for a representational knowledge. Also, the use of environment as its own model allows reactive control to use perceived states of the environment to avoid explicit symbolic representation and reasoning. The approach, therefore, allows fast, robust, and flexible robotic control systems to be built, and is often referred to as Reactive Planning or more generally as Behaviour-Based AI or BBAI (Brooks 1986, 1989,1991, 1999; Agre, Chapman 1987; Kaebling 1989; Kaebling, Rosenschein 1994). If, however, the reactive agent wishes to make use of symbolic representation, reasoning, planning, etc, problems arise. For example, how is knowledge about the agent's environment and its physical form instantiated into the agent's cognitive architecture.

2.7 Perceptual Symbol System

A common philosophical position is that perception and cognition reflect independent or modular systems in the brain – perceptual symbols pick up information from the environment and pass it on to separate systems that support the various cognitive functions, such as language, memory, and thought. This position assumes that cognition is inherently non-perceptual.

One recent theory that relates cognition strongly to perception is that of Lawrence Barsalou, Professor of Psychology, who argued that this view is fundamentally wrong (Barsalou 1999). He suggested that instead, cognition is inherently perceptual, sharing systems with perception at both the cognitive and neural levels, and therefore knowledge has a strong perceptual character. The basic assumption underlying Barsalou's Perceptual Symbol System hypothesis is that sub-sets of perceptual states in sensory-motor systems of the brain are extracted and stored in long-term memory to function as symbols. As a result, the internal structure of these symbols is modal, and they are analogically-related to the perceptual states that produced them. He also forwarded the conjecture that the schematic nature of perceptual symbols falls out from two assumptions about attention: selective attention (1) isolates information in perception and (2) stores the isolated information in long-term memory (Barsalou 1999).

Perceptual symbols do not exist independently of one another in long-term memory, Barsalou clarifies. Instead, related symbols become organized into a 'simulator' that allows the cognitive system to construct specific simulations of an entity or event in its absence. Once established, these simulators implement a basic conceptual system that represents types, supports categorization, and produces categorical inferences. These simulators further support productivity, propositions, and abstract concepts, thereby implementing a fully-functional conceptual system (Barsalou 1999). The formulation of the Perceptual Symbol System theory can be viewed as a high-level functional account of how the brain could implement a conceptual system using sensory-motor mechanisms, Barsalou believes. Once the possibility of such an account has been established, later work can develop computational implementations and ground them more precisely in neural systems.

2.8 Control System Approach

Although BBAI works well for reactive agents, it seems inappropriate for cognitive agents which need not only be able to operate reactively in response to events and tasks, but also deliberate in a purposive manner and demonstrate cognitive qualities, such as planning, reasoning, learning, etc. When trying to find an alternative starting point for a theory about the nature of mind and cognition, and then investigate the possibility of achieving synthetic intelligence and artificial cognition in (as opposed to adopting BBAI for that purpose) there are many options.

Some philosophers start from the notion of rationality or from a small number of familiar aspects of human mentality, such as beliefs, desires, and intentions. Sloman suggests that it would be more useful to step back to the general notion of a *control mechanism* that interacts with a dynamic environment, including parts of itself, in a way that is determined by the changeable internal state of the mechanism, the state of the environment, and the history of previous interactions (Sloman 1993). Sloman maintains that this is a deeply causal concept – the concept of a *control system.* However, as he further explains, this is still too general, because the notion of such a control system covers many physical objects, both naturally occurring and manufactured, that clearly lack minds. By adding extra constraints to this general concept, he believes, we may be able to home in on a set of interesting special cases, more or less like human beings or other animals.

Common philosophical questions about this alternative approach to the study of mind and cognition, and the possibility of achieving synthetic intelligence and artificial cognition usually include whether all mental events and cognitive phenomena can be reduced to some sub-set, e.g. whether all mental states can be defined in terms of collections of beliefs, desires, and intentions. But the variety found in children, brain-damaged people, or even some animals suggests that there are no absolutely necessary conditions for the existence of mental capabilities – only a collection of different designs with different properties (Sloman 1993).

In particular, this approach leads to a new analysis of the concept of representation. As representation is obviously part of a control system, different kinds of representations play different roles in control mechanisms and processes. The Control System approach allows the mind and cognition (or a cognitive agent with a mind) to be considered as a set of control mechanisms and processes capable of supporting a number of control states (Simon 1967; Sloman 1993). This approach is based on developing architectures and theory of mind arising from the works of Sloman, Davis, and some others (Sloman, Croucher 1987; Sloman 1993; Sloman, Beaudoin, Wright 1995; Sloman 1996; Sloman, Logan 1999; Davis 1996…2011).

2.8.1 Control State Theory

Sloman has maintained for years that it is more fruitful to construe the mind or an agent with a mind as a *control system* than as a computational system, although computation can play a role in control mechanisms and processes (Sloman 1993). He explains that the view that a mind is a control system is not provable or refutable – it merely defines an approach to the study of mind. Sloman also holds that the Control System approach to mind builds on an assumption that mind is a collection of many different control processes and mechanisms passing data between them asynchronously, as opposed to a single computation that can be reduced to a single state at any given time (Sloman 1987, 1993). Davis (2001) holds a similar view on this. He considers mind as a dynamic structure of asynchronous data, information and knowledge-processing mechanisms, and the information-rich control states they support.

A *control state* is a behaviour internal to an agent. Control states can exhibit external behaviours, such as obstacle-avoidance, or reflect and control internal states, such as beliefs. In essence, control states can be a number of things, such as beliefs, desires, intentions, motivators, reflexes, etc. They can be considered as general behaviours within an agent, and therefore can be modified or updated at any time by some other processes. This view, which is also shared by Minsky (1985) is effectively built on the seminal work of Simon and Newell (Newell, Simon 1972, 1976; Simon 1967, 1996, 1999).

Control states do not need to exhibit external behaviours though – they can be used to control the agent's internal states. What is required is a specification on how the control states interact with one another (Davis 2008). This includes how a control state modifies other control states or is modified by them, the type and amount of information passed to other control states or received from them, whether the information passed and received is direct or filtered by some other control state, and finally if a control state produces behaviours directly or modifies other control states to achieve its ends (Sloman 1993, 1999).

Specific control states can exist at several levels, requiring different types of processing (Davis 1998). For example, reflexes that are reactions to unexpected stimuli may involve body actions and deflection of cognitive processing from current tasks, and may also influence future high-level processes. Also, it must be noted that control states need not be symbolic in nature, e.g. a reflex that may be known at the deliberative level, which is symbolic, but is actually reactive.

The view point that mind is a control system provides an important perspective to the design of autonomous cognitive agents and mobile robots. It builds on the assumption that cognition is an epistemic process that can be modelled using information processing architectures (Sloman 1993; Beaudoin 1994; Davis 2001…2010). Such information processing architectures can be in any number of control states, and the processing of information can give rise to changes in the current set of control states. Since the nature of information processing is dependent upon the control states, the same information may be processed differently in different control states (Davis 2010). This type of approach also requires us to specify our theories from the standpoint of how things work – how perception works, how motives are generated, how decisions are taken, how learning occurs, and so on (Davis 2008). Here are a few of these control states of particular relevance to this research work:

- **Beliefs**: Beliefs are internal models, possibly assumed or inferred from perceptions and perceptual acts, or from information arising as the result of internal processes or other control states. They are states in which an agent is confident about the truth of a proposition.
- **Goals**: Goals are internal or external states that an agent wishes to achieve, prevent, or maintain. They are desirable end states that are either qualitative, quantitative, or a combination of both.
- **Desires**: Desires are symbolic statements that define a specific preferred state. They underpin goals and other purposeful behaviours that are related to beliefs and other control states.
- **Intentions**: Intentions are plans of actions – a set of intended tasks – that can be inferred from beliefs and desires of an agent, or explicitly provided, influenced by other control states such as affective and motivational states.

- **Motivators**: Motivators are dispositions and tendencies to assess situations and respond to those situations and assessments in a certain way. They can provide a *context* and *impetus* for reasoning about events, and also a *basis* for goal-directed behaviours. They are, therefore, often used as a generic framework that draw all these control states (beliefs, goals, etc) together.

2.8.2 Affective and Motivational Control States

Scientific evidence highlights the fundamental role of *affect* in rational and intelligent behaviour (e.g. Darwin 1872; Duffy 1941; Camras 1992; Elliott 1992, 1994; Damasio 1994; Bates 1994; Bower 1994; Clore 1994; Ekman 1992…1999; Ekman et al. 1977…1994; Simon 1967…1999; Picard 1997…2010; Sloman 1993…2003; Clocksin 2004; Aube 2005; Davis 1996…2010; Davis, Lewis 2003, 2004). Since there appears to be a requirement for something analogous to affect in artificial cognitive systems, it has been conjectured to use *affective control states* which makes affect the basis of a consistent and systematic control language across a cognitive architecture (Sloman 1993…2003; Davis 2001…2004). This control language is grounded in the use of affect with the aim to be consistent across different domains, tasks, and levels of processing (Davis 2010).

Further research highlights that *motivation* too plays a fundamental role in a variety of cognitive functions (e.g. Duffy 1941; Beck 2000; Westen 1996; Simon 1967, Davis 1996…2010; Sloman 1993…2003; Sloman, Logan 1999; Arkin 1998). Motivation cannot be observed directly, but can be inferred from the observable behaviour of an agent. From the viewpoint of mind as a control system, motivation can thus be thought of as a control state – facilitating the use of *motivational control states*. Motivations arise from the perception of the world and evaluation of events relevant to goals (Duffy 1941). Each kind of evaluation gives rise to a distinct signal that reflects the priority of a goal to an individual, which then influences the resulting behaviour. Therefore, motivations drive behaviour or action selection (Arkin 1998). Motivations are essentially more encompassing than goals though. The reason is that motivations include not only the descriptions of goals, but also an affective context for those goals (Davis 2001).

Psychologists typically see motivation as the force that lies behind and gives sense to changes in behaviour. Thus, motivation controls the changes in behaviour, as well as drive behaviour or action selection. Put simply, motivation is the driving force behind all actions of an organism. On the other hand, affect exerts further control by triggering certain mechanisms and processes that may influence the intensity of the selected behaviour or action, or perhaps enable it or even prevent it (Arkin 1998). In general, some of these internal forces are more of a physiological kind, such as needs (hunger, thirst, etc), while others are more of a psychological kind, such as emotions (anger, fear, etc). These two types of motivational states are those most thoroughly-studied so far, but probably do not exhaust all of the internal forces that determine behaviour (Aube 2005).

According to leading researchers (e.g. Simon 1967; Izard 1991, 1993; Davis 1996, 2001, 2008, 2010; Sloman 2001, 2003) affect and motivations serve as *filters* that guide perception and action, determine the input into the evaluation processes, and manipulate the evaluation of perceptual information. Thus, emotional and motivational experiences are essential aspects of the process of cognition, and therefore must be considered in any adequate description of it. As motivators provide a context and impetus for reasoning about events and also a basis for goal-directed behaviours, they are often used as a generic framework that draw all the control states of a cognitive architecture together. A motivator often includes the following components (Sloman 1994; David 2001; Davis 2010):

- semantic content that represents a proposition denoting a possible state of affairs, which may be (categorically or partially) true or false.
- motivator attitude to the semantic content that represents the motivator's tendency for acting towards the semantic content (e.g. make true, keep true, make false).
- belief status that is an indication of current belief about the status of the semantic content (e.g. true, false, partially true).
- actors, agents, and objects referenced by the motivator.
- behaviours associated with any intention or plan set or past similar motivators.
- current commitment status of the motivator (e.g. rejected, undecided, complete).

- goal importance or *urgency* (in the range `[0, 1]`) that is managed by the BDI-affective processes in motivator-evaluation ad goal-revision.
- association value or *insistence* (in the range `[0, 1]`) that is managed by the BDI-affective processes, and is strengthened by goal-completion.
- motivator value or *intensity* (in the range `[0, 1]`) that is a value associated with the difference between the current and desired state of the motivator.

It now seems reasonable to conclude that motivation and affect can be co-joined in perception and cognition. Given the fact that motivations are essentially more encompassing than goals (since they include not only the descriptions of goals but also an affective context for those goals) plus the fact that motivators are often used as a generic framework (since they can draw all the control states of a cognitive architecture together) the use of control states to develop cognitive architectures leads to the use of *affective and motivational control states* (Sloman 1993…1999; Davis 2001…2004; Davis 2010). While this theory is in development, and arguably currently incomplete, the presented argumentation above provides sufficient support for the design of the current cognitive architecture under investigation – CernoCAMAL.

2.9 Summary

This chapter began with a brief introduction to the fields of Cognitive Science and Artificial Intelligence. Two major approaches to the study of these two inseparable areas of science were pointed out. The concepts of Cognitive Modelling and Cognitive Architectures were explained, and commonly used types of cognitive architectures were highlighted. The notion of Theory of Mind was introduced, and a flavour of the familiar phenomena of Emotion and Motivation was presented. This was followed by some background knowledge in Autonomous Cognitive Agents and Mobile Robots research. Three major approaches to the study of this area of research, and the possibility of achieving synthetic intelligence in autonomous cognitive agents and mobile robots were presented. The latter – as the cornerstone of this work – was discussed in detail, followed by an introduction to the Control State theory. The chapter concluded with presenting the notion of Affective and Motivational Control States and how they were used in developing cognitive architectures.

3 Literature Review

This chapter begins with an introduction to a general class of computational cognitive architectures known as CAMAL. A review of the original overarching architecture is presented, as an important pivotal axis for the development of CernoCAMAL. It is described what progress has been made since the start of the CAMAL research. Its priors (e.g. CogAff) and its spin-offs to date are introduced, and subsequently its components and organization are presented, including the BDI (Belief-Desire-Intention) and affect models. This is then followed by a brief discussion on the works of CRIBB and a-CRIBB reasoning models, and how their architectures make use of BDI and affect schemas. The chapter continues with introducing the association and motivator constructs. The concept of a motivational blackboard is presented, followed by explaining how CAMAL's blackboard facilitates the operation of its motivational constructs. The role of a domain model is highlighted, and its components are then pointed out. A brief operational overview of CAMAL is given before proceeding to a broader literature review in the field of Cognitive Architectures. The chapter concludes with reviewing the most notable and significant examples of cognitive architectures, in particular the recent work of RoboCAMAL.

3.1 Introduction to CAMAL

CAMAL is an acronym for **C**omputational **A**rchitectures for **M**otivation, **A**ffect, and **L**earning. The CAMAL cognitive architecture is an example of a general class of integrative cognitive architectures; drawing together a number of threads in Cognitive Science and Artificial Intelligence, such as perception, action, decision making, motivation, affect, and learning. CAMAL has been in development for nearly eleven years. It was proposed by Davis (2001) and was developed from ideas incorporated in Guardian (Hayes-Roth 1995), ACT-R (Anderson, Lebiere 1998; Anderson, Matessa 1998; Anderson et al. 2004), CRIBB (Wahl, Spada 2000), CogAff (Sloman 2001), and the cognitive architectures of Singh and Minsky (Singh, Minsky 2002). It is, essentially, a UTC that tries to answer many of the questions that comprise Norman's Cognitive Science agenda (1980).

However, due to the fact that a UTC, as a single set of mechanisms and processes for all cognitive behaviour, addresses a vast undertaking, the CAMAL spin-off projects have necessarily focused on a sub-set of issues, e.g. adding affect and affordances (Lewis 2004), its use in controlling autonomous mobile robots (Gwatkin 2009), etc. The main purpose of this computational cognitive architecture is to investigate synthetic intelligence and artificial cognition in autonomous cognitive agents and mobile robots, drawing on qualities found in natural minds.

Previous work on CAMAL has mainly involved agents in simulated environments. Using agents and representation of mind as a collection of agents has a long history. Philosophical backgrounds of the multi-agent mind can be found in the Pandemonium theory of mind as a collection of demons (Selfridge 1959) and in Marvin Minsky's ideas of mind as a society of agents (Minsky 1985). Note the similarity of these theories to Baars' theatre metaphor and global workspace (Baars 1997) as hypothesized algorithms for the workings of the highest and most general levels of brain organization. Considerable research has been conducted in monitoring such a group of agents in cognitive robotic testbeds and then making changes, to study their behaviours and decision-making.

Preliminary work (Davis 1996…2001) inspired and highly influenced by the CogAff project (Sloman 1993; Beaudoin 1994) centred on goals and motivators; how they came into being and how they were managed. Since then, various closely-aligned projects have focused on using affect and affordances to drive goal selection and behaviours (Nunes 2001; Bourgne 2003), how to relate the BDI model to affect and affordances (Lewis 2004), the role of metacognition in directing the focus of cognitive architectures (Venkatamuni 2008), and the use of the architecture in controlling mobile robots (Gwatkin 2009). These five research projects have all concluded successfully.

- *Investigation of Motivation in Agents using the Simulation of 5-aside Football* (Nunes 2001)
- *Affect-Based Multi-Agent Architecture for a 5-aside Football Simulation* (Bourgne 2003).
- *a-CRIBB: Computational Models of Emotion and Affect* (Lewis 2004)
- *A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture* (Venkatamuni 2008)
- *RoboCAMAL: Anchoring in a BDI Motivational Cognitive Robot* (Gwatkin 2009)

Nothing in these major works and similar ones by Davis and Sloman so far has really undermined the four-layer principle to a theory of mind and cognitive architectures; meaning a four-layer architecture provides an appropriate guiding model. Furthermore, the Control State approach is still valid, though it may not necessarily be an exhaustive description of what could be occurring within a mind.

The most recent phase of this research (Gwatkin 2009) focused on translating the concepts of motivation, affect, and learning from a simulated agent to an embedded robot. Effectively, it moved from computer-simulated agents towards situated and embodied agents[3]. The concept of *'situated and embodied cognition'* is a research approach that relates the activities of a cognitive agent with the environment within which the agent's activities take place (Clancy 1997; Clark 1998; Pfeifer, Scheier 1999). Put differently, situated and embodied cognition refers to the role the environment plays in the development of cognitive processes within an agent; meaning an agent's behaviours and cognitive processes are context-dependent.

'Situated' refers to the fact that the agent is 'present' within some environment – it is situated within some environment. 'Embodied' refers to the fact that the agent has a 'physical presence' within some environment. Essentially, the situated and embodied cognition concept postulates that the mind should not be treated as a simple information processor, but that it is tied to the environment and the physical body of the cognitive agent. The distinct components of the domain model (see 3.7) – namely the environment domain model and the attributes domain model – reflect the situated and embodied nature of an agent.

Also, it is noteworthy that the word 'situated' is not actually meant to imply that cognition is fixed to localized situations. Instead, situated cognition emphasizes that there is an inescapable environmental context for cognitive activity, and that cognition takes place within a totality of activity, including social activity (Clocksin 2004). The situatedness of cognition is actually a consequence of embodiment. The embodied nature of cognition suggests that cognition is not a mental machine working on abstract problems, but it is an activity connected with a body that requires cognition to make it function (Wilson 2002) or as Sloman puts it (Sloman 1993)

[3] Note that an agent in a simulation world can also be considered *situated*. However, this statement is meant to emphasize the fact that a cognitive mobile robot has a physical presence within a physical environment.

mind is a control system. Moving towards embodiment can be seen in the transition from the CRIBB and a-CRIBB projects to the recent work of RoboCAMAL (Gwatkin 2009). Before outlining the components and organization of CAMAL, it would be instructive to briefly reflect on the cognitive architecture of CogAff that has inspired the development of CAMAL. Later on (see 7.2) the thesis refers to CogAff again as a well-known representative cognitive architecture. The name CogAff is used both for the project and as a label for a generic schema proposed by Sloman (2001) for a wide variety of architectures, both natural and artificial. CogAff itself was inspired by the Three Tower / Three Layer model proposed by Nilsson (1998) with vertical divisions, shown in Figure 3.1. There are different versions of this model, depending on the sophistication of the perceptual, central, and motor sub-systems. Arrows from left to right (thick arrows) indicate the flow of perceptual data. Arrows from right to left (thin arrows) indicate the flow of feedback.

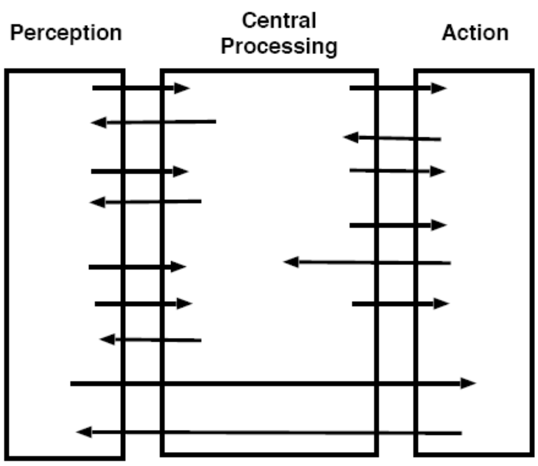

Figure 3-1: Three Tower Model (Nilsson 1998)

The Three Layer model, depicted in Figure 3.2, attempts to account for the existence of a variety of more or less sophisticated forms of information processing and control, which can operate concurrently. The version shown here postulates three concurrently-active layers which are found in different biological species. The three layers account for different sorts of processes, found in different kinds of animals and humans.

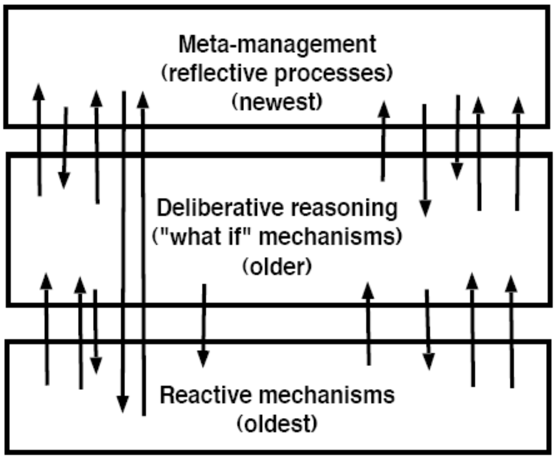

Figure 3-2: Three Layer Model (Nilsson 1998)

The first layer contains *reactive* mechanisms which automatically take action as soon as appropriate conditions are satisfied. The *deliberative* layer provides high-level reasoning capabilities (e.g. what-if reasoning) required for planning, decision-making, etc. Relatively few organisms have this, and again the forms can vary widely. The *reflective* or *metacognitive* layer provides the ability to monitor, evaluate, and control internal processes and strategies. Within the reactive layer, when conditions are satisfied, actions are performed immediately (they may be external or internal actions). By contrast, the deliberative layer, instead of always acting immediately in response to conditions, can contemplate possible actions, compare them, evaluate them, reason about them, and select among them. At least in humans, chains of possible actions can be considered in advance, though there are individual differences in such capabilities. The human deliberative system can also consider hypothetical past or future situations not reachable by chains of actions from the current situation, and can reason about their implications (Sloman, Logan 1999). The illustration of Figure 3.3 shows the complete CogAff architecture.

In a nutshell, CogAff is an architectural framework designed to support interaction between cognition and affect. It posits three distinct levels of processing: a reactive level uses condition-action associations that respond to immediate environmental situations; a deliberative layer operates over mental goals, states, and plans to reason about future scenarios; and a meta-management mechanism lets the agent think about its own thoughts and experiences.

Most of the three-layer architectures do differentiate between the three towers of perception, central processing (cognition), and action across all layers. This provides a useful starting point to consider the contexts for processing across a cognitive architecture. However, as Sloman points out, this can lead to an inherent bias to simplistic left-to-right processing models based on sense-think-act cycles (Sloman 2001). Such cognitive models may suffice for simple agents with limited tasks, but can restrict the development of more sophisticated information processing that includes feedback, directed perception, and sophisticated behaviour-environment interactions (Davis 2008).

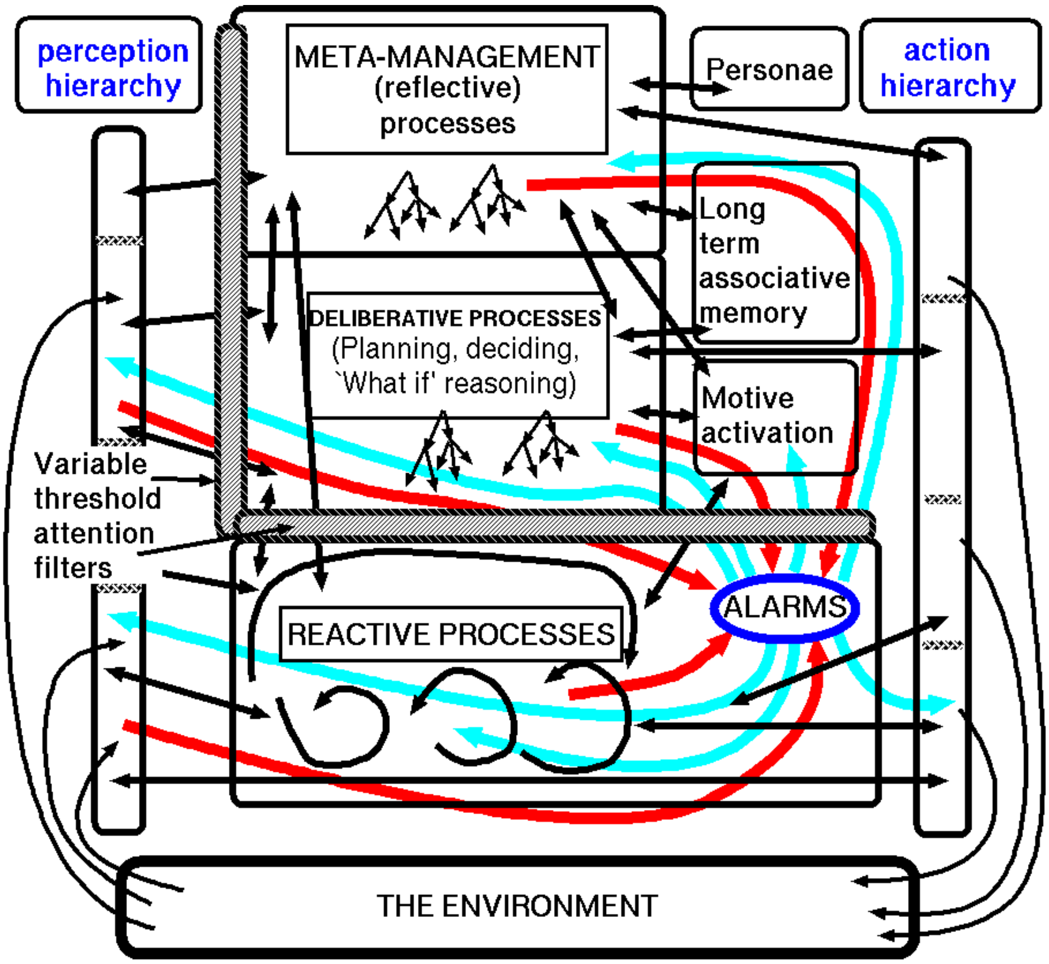

Figure 3-3: The CogAff Architecture (Sloman 2001)

3.2 Layers and Organization

In CAMAL, as depicted in Figure 3.4, there are two more columns in addition to the three towers of perception, central processing (cognition), and action present in architectures such as CogAff and Nilson's. The *affect* column conceptualizes the affect model, which in actuality is distributed across the entire architecture. The *motivation* column contains the architecture's motivational blackboard, itself containing motivational constructs. This is all coordinated by the cognition and deliberative layers that make use of various cognitive processes, such as the BDI and affect reasoning models.

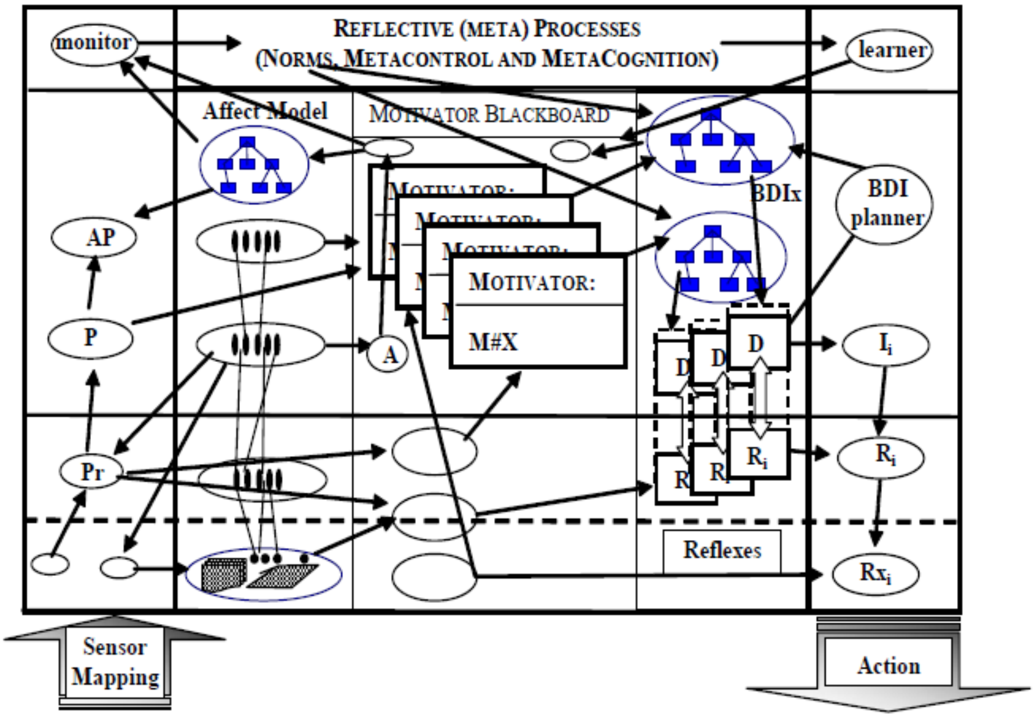

Figure 3-4: The four-layer five-column CAMAL Model (Davis 2010)

The lowest level is the *reflexive* level that uses a perception-action process to generate behaviours. The data from the agent's perceptual systems are collected and, based on their values, simple actions are performed. These actions can be combined to produce simple, default behaviours. At this level, there is no explicit representation of the agent's environment.

The next level is the *reactive* level that builds on reflexive mechanisms to generate more complex behaviours, such as goal-oriented behaviours. The goals at this level are usually reactions to the agent's internal and external states. It may be recognized that the reflexive and reactive layers together comprise a 'Brooks-style' controller that provides flexible and robust low-level control. Again, at this level, there is no explicit representation of the agent's environment.

The *deliberative* level is more complex than reflexive and reactive layers. It provides deliberation, planning, decision-making, and problem-solving capabilities that can be used for carrying out different tasks, such as achieving goals, etc. Unlike the first two levels, the deliberative layer holds a representation of the agent's environment. This representation should be constantly updated and maintained by examining the inputs from agent's perceptual systems, and also the interactions of reflexive-reactive layers with the agent's environment.

The purpose of the *reflective* or *metacognitive* layer is to monitor and control the processes generated by the interaction of the reflexive, reactive, and deliberative levels. By monitoring these processes, the reflective layer looks to suppress unwanted behaviours and promote useful ones. Put simply, it serves to monitor the overall behaviour of the agent.

3.3 BDI Model

With the rise of AI and widespread discussions about whether machines can think, understand, have emotions, etc, the idea that machines might have *mental states* that resemble our own has been pursued with some powerful tools, such as cognitive modelling and the development of cognitive architectures. The philosophical idea of cognitive agents that possess beliefs, desires, and intentions has been particularly studied in Cognitive Systems and Cognitive Robotics research. These are known as BDI agents, where beliefs, desires, and intentions represent mental states that are being given clear semantics (Bratman 1987; Georgeff, Rao 1995; Georgeff et al. 1999).

Such systems demand considerable cognitive flexibility, requiring the ability to reason over collections of beliefs and knowledge, deal with high levels of uncertainty, respond to unexpected circumstances, etc. In such systems, decision making and planning begin with some collection of beliefs. Under certain circumstances, a goal will be adopted. Where there are a number of options in a situation, a decision must be made. Arguments for and against different possible decisions can be developed and weighed. Depending upon the arguments, an intention or action or behaviour may be adopted and then carried out. Actions may lead to new situations and, possibly, new beliefs.

In effect, the BDI framework can be considered as a Logical Theory of Rational Agency (Georgeff, Rao 1991, 1995). The leading agent theories in this area more or less share similar logical properties, bringing together an informational component (in order to represent the agent's beliefs and knowledge), a motivational component (in order to represent the agent's desires and goals), and a dynamic component (in order to represent the agent's activities). Thus, the BDI approach typically combines informational aspects, motivational aspects, and dynamic aspects of propositional logic (Fisher et al. 2007). Unfortunately, this framework lacks an explicit mechanism to express the plausibility of the agent's beliefs.

More recently, it has been proposed to integrate into the BDI approach of Bratman, Georgeff, and Rao *psychological notions* in order to develop more complete cognitive models of agents capable of sustaining more human-like interactions with people, especially ordinary people involved in conversational activities with assistant agents. For example, Gratch and Marsella (2004) have proposed a model of emotions based on SOAR, with a significant impact upon the SOAR cognitive architecture. Emotions have, also, been integrated into the BDI framework, for instance with eBDI (Jiang et al. 2007). Although there has been a lot of research works about the effects of *personality* on agents' behaviours in the virtual agents community (one of the most recent one being the SEMAINE project: Bevacqua et al. 2010) they generally focus more on their impact on the animated agent (e.g. gaze or facial expressions) than on the rational decision processes (Sansonnet, Bouchet 2011).

Within the CAMAL architecture, the BDI model provides a method to control the flow of information through the deliberative component, determining the intentions, actions, or behaviours of the agent based on its beliefs and desires. Obviously, the confidence an agent can have in a belief can vary, and depends on the source of the belief. Beliefs are based on inputs from the agent's perceptual systems and also its previously-held beliefs. They could also be assumptive (default) beliefs, or possibly deducted from reasoning and perceptual acts. The same applies to desires and goals.

By coupling the agent's beliefs and desires, a set of intentions or actions or behaviours can be generated as to how to achieve the agent's goals. But do we categorically need a BDI model within CAMAL as a cognitive architecture? The answer is yes. A simple example will elucidate why: One way of achieving an agent's goal is to simply perform an action based on the current beliefs that the agent has. If the environment changes, the agent's beliefs should be modified accordingly, thus the previous action may no longer be valid. As the action has no access to the new beliefs, it will still act and may fail to achieve the agent's goal. It is for this reason that intentions are required. Intentions are akin to plans of actions or behaviours or possibly reactive sub-architectures that are based on the agent's desires, and have access to the agent's beliefs. If the environment changes and the agent's beliefs are updated, then the agent's intentions should be modified accordingly to prevent the agent from failing its goal.

3.3.1 CRIBB Model

A number of scientists including Wimmer, Perner, Wahl, and Spada (Wimmer, Perner 1983; Wahl, Spada 2000) set up a series of experimental tests to check whether children between three and five years of age were able to attribute a false belief to someone else. In one of these experiments, children watched a scene in which a character, called Maxi, put a bar of chocolate in a drawer and went away. While he was away, his mom took the chocolate and put it somewhere else. When Maxi came back, the clip was stopped and the children were asked by the experimenter: Where will Maxi look for his chocolate? The results showed that children over five did not have any problems in attributing to Maxi a false belief, whereas children below five predicted that Maxi would look for the chocolate where his mom put it!

Wahl and Spada later developed a computer model called CRIBB (Children's Reasoning about Intentions, Beliefs, and Behaviour) to investigate reasoning in young children. The model is based upon Bartsch and Wellman's general theory for Belief-Desire reasoning (Bartsch, Wellman 1989) and simulates knowledge and inference processes of a competent child solving problems and false-belief tasks. It was designed using a symbolic cognitive modelling approach and was implemented computationally to allow for the detailed examination of representational and operational demands in the Theory of Mind tasks in children.

The CRIBB model represents propositions about physical states of a given situation and about another person's intentions, beliefs, perceptions, and behaviour. Inferences are drawn from its knowledge base by inference schemata. This set of inference schemata simulates the knowledge that we assume to be a central part of a child's Theory of Mind. The BDI model of CRIBB can be seen in Figure 3.5. The inputs into this model are perception, basic emotions, and physiology. The agent's beliefs are determined by its perception of the environment and its previously-held beliefs. The agent's desires are determined by its basic emotions, physiology, and its previously-held desires. The coupling of the agent's beliefs and desires manifests itself as intentions or plans of actions or behaviours. These can lead to reactions that may consequently change the agent's environment and possibly the agent's emotional state and physiology. These changes, in turn, can alter what the agent perceives of its environment, and so on.

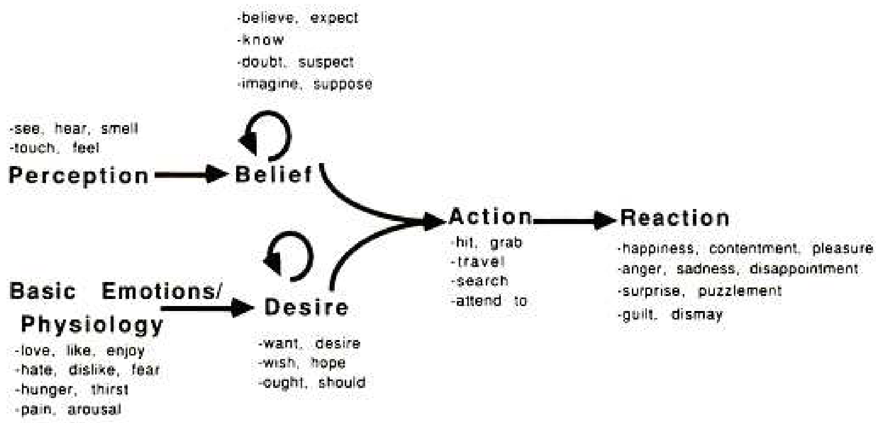

Figure 3-5: Belief-Desire Reasoning Model (Bartsch, Wellman 1989)

A further element of CRIBB is a Consistency Mechanism as illustrated in Figure 3.6. It detects and resolves any contradictions in the system's beliefs. It is invoked each time a new proposition is added, in order to ensure the consistency of the knowledge base. If a contradiction is found, then the more certain proposition is added to the knowledge base. The certainty of a proposition is determined by its source. For example, if a proposition is formed based on a situational knowledge, then it is considered more certain than a proposition that is formed based on an assumption.

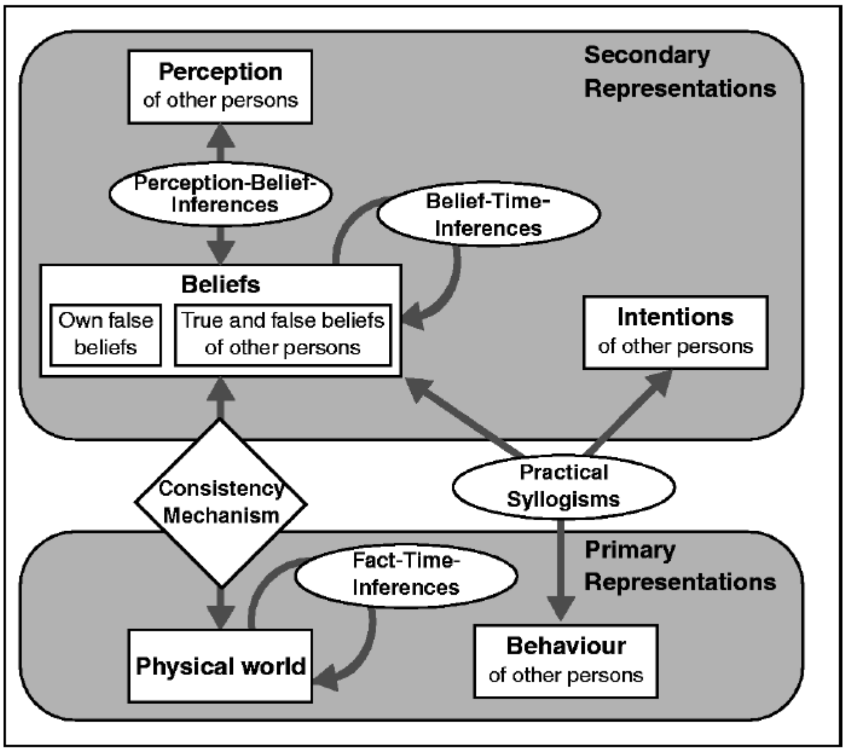

Figure 3-6: High-Level View of CRIBB Architecture (Wahl, Spada 2000)

3.4 Affect Model

There was a growing consensus among AI thinkers (Minsky 1985; Sloman, Croucher 1987; Franklin 2000; Sloman 2001; Davis 2001) that artificial minds and cognitive agents, to be complete and believable, require a computational equivalent to emotion in order to complement their behavioural and cognitive capabilities. This requirement was originally highlighted in earlier prominent research (Simon 1967; Norman 1980). Later work by Davis (2001…2004) questioned this 'requirement for emotion' attitude in intelligent systems. Davis's thesis was that emotion per se is not really a requirement for the majority of synthetic intelligence theories or cognitive systems (Davis 2004).

As emphasized earlier, Davis also suggested that a direction given by the less semantically-overloaded term 'affect' would be more appropriate for synthetic intelligence and artificial cognition than the term 'emotion'. Davis (2004) then conjectured the alternative Theory of Affect which draws on theories such as Control States (Simon 1967; Sloman 1987, 1993; Davis 2000, 2001) and Perceptual Affordances (Gibson 1979). His argumentation for *affect as a control mechanism* makes use of the Control States approach to mind, his experimental work with cognitive agents, and his on-going designs for synthetic intelligent systems.

Affect is generally defined in terms of information processes and representational structures across a cognitive architecture. It is qualitatively defined as negative, neutral, or positive (i.e. it can be experienced as negative, neutral, or positive valences) and can be mapped numerically over the interval `[-1.0, +1.0]`. The use of affect and affective control states makes affect the basis of a consistent and systematic control language across a cognitive architecture. It also allows external events and objects to take valenced affordances, and internal mechanisms to be prioritized via valenced processes (Davis 2010) [4].

In many cognitive systems, an affective module is implemented as a separate independent part of the architecture that interacts with other sub-systems (e.g. TOK Architecture of Reilly and Bates 1993). However, as highlighted earlier, research into affect shows that affective states have wide-ranging permeating effects on numerous cognitive functions, such as reasoning, learning, perception, memory, and decision making. This leads to the conclusion that affect is not really a separate entity; it is a distributed mechanism across the entire cognitive architecture. Put simply, affect is distributed across all the sub-systems, hence the earlier statement on CAMAL layers and organization that the affect column *conceptualizes* the affect model, which in actuality is distributed across the entire architecture (see 3.2).

[4] Valence refers to the intrinsic attractiveness (positive-ness) or aversiveness (negative-ness) of an event, situation, or object. For example, negative emotions such as fear and anger have negative valence, whereas positive emotions such as joy and pleasure have positive valence.

Davis's earlier work on cognitive agents and control states focused on goal processing and addressed how goals and related control states need to be valenced in a number of different ways. Davis's current model of affect allows various elements within the architecture to have an associated *magnitude* that can fluctuate according to success or failure associated with that element. The affective valencing of information processes and representational structures can be given (i.e. defined) or the agent can adapt or learn appropriate affordances according to its role and current environment (Davis 2010).

3.4.1 a-CRIBB Model

Based on our daily lives and lived experiences, there is no doubt that emotions can disrupt reasoning under certain circumstances, and that misdirected or uncontrolled emotions can lead to irrational behaviour. For many years, the dominant view was that emotions are quite distinct from the processes of rational thinking and decision making, and are often a major impediment to those processes. To paraphrase Davis, they were often considered to be *"the Achilles' heel of reason!"* (Davis 2000, p.1). Damasio made a strong case that this traditional view (which he dubbed *Descartes' Error*) is wrong, because emotions and affect actually play a fundamental role in rational thinking and decision making (Damasio 1994). Damasio maintained that when a system's underlying affect do not function properly, rational decision making breaks down.

Other researchers and theorists such as Sripada and Stich (2005) have also challenged this attitude. They have argued that if we view affect through the longer lens of evolutionary theory, we can see that much of what looked to be irrational is actually part of an effective strategy for achieving an agent's goals and maximizing its success. Therefore, an affective element should be considered as an essential part of any cognitive system. Today, this is an accepted consensus among theorists and designers of cognitive systems that synthetic minds and cognitive agents, to be complete and believable, require a computational equivalent to affect, in order to complement their behavioural and cognitive capabilities (Simon 1967; Norman 1980; Minsky 1985; Sloman, Croucher 1987; Franklin 2000; Sloman 2001; Davis 2001; Robinson, El-Kaliouby 2009; Vallverdu, Casacuberta 2009; Davis 2010).

Wahl and Spada's computer model of CRIBB did not incorporate basic emotions that are present in the original schema of Bartsch and Wellman's (1989) Belief-Desire reasoning model. To incorporate this important aspect of cognition and investigate its use, the CRIBB model was extended with an *Affect and Affordance Model* and was dubbed *a-CRIBB* (Davis, Lewis 2003, 2004). Therefore, the a-CRIBB reasoning model is an extension of the BDI model used in CRIBB. Adding affect and affordances resulted in more effective cognitive processing and task management, based on extensive experiments carried out in various simulation terrains.

The first major change compared to CRIBB is that the framework is implemented as a situated agent in a synthetic world of a fungus eater. The agent's motivation is to collect minerals and fungus. Other extensions to the CRIBB model, which are effectively the main contributions of the a-CRIBB model, are as follows:

- A distributed model of affect across all the sub-systems.
- The inclusion of affective affordances applied to the perception of the system, which is an extension to Gibson's (1979) Theory of Affordances.
- A revision of the consistency mechanism used to resolve belief contradictions by incorporating affective correspondences.
- A description of a goal base, including the goal-maintenance, goal-importance, and goal-achieved mechanisms.
- A central monitoring system using Oatley and Johnson-Laird's (1987) monitoring mechanism to overcome the problem of communication in the system.

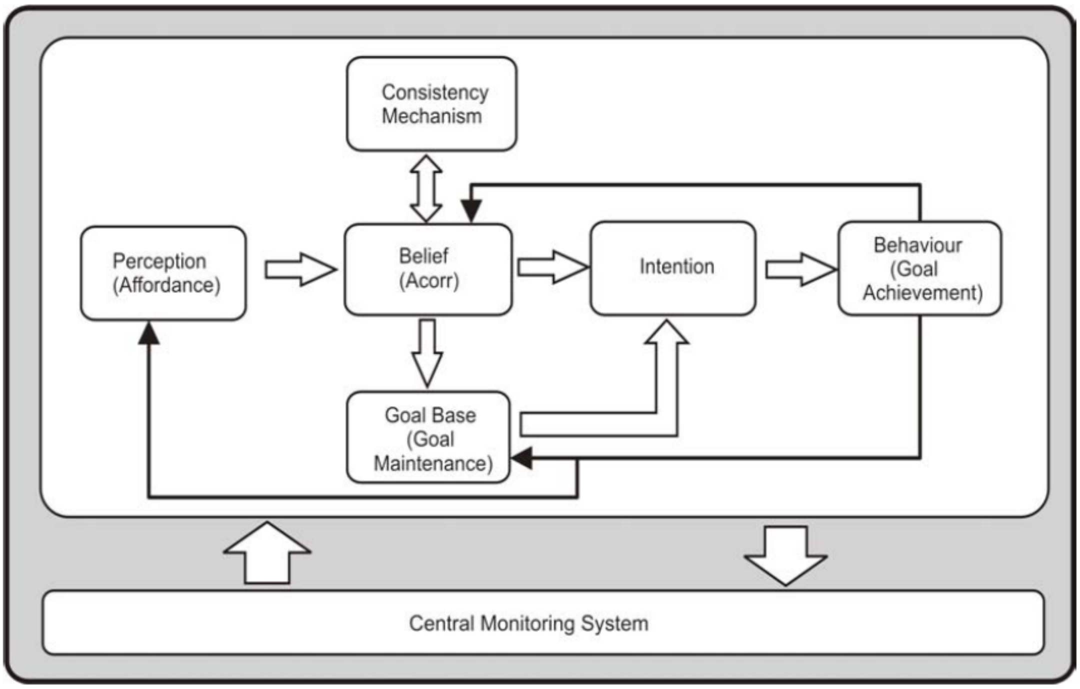

Figure 3-7: High-Level View of a-CRIBB Architecture (Davis, Lewis 2003, 2004)

3.5 Associations

The a-CRIBB reasoning model was developed to investigate the incorporation and use of affect and affordances within the CRIBB reasoning model. Essentially, it added several new elements to the original CRIBB – the main two being the *affect model* and *associations*. CAMAL uses a variant of the a-CRIBB reasoning model (Davis, Lewis 2003, 2004) i.e. a BDI model and an affect model, plus a motivational blackboard that uses affective affordances to instantiate and modify motivators. At the deliberative level, affective values and affordances can be associated with processes and predicates, and then relayed as control signals to instantiate and modify motivators and their associated representations and behaviours. The affect model distributes affect values and affordances across the entire cognitive architecture, rather than have a centralized module. This is made possible by the use of associations.

An association is a construct that contains a belief-desire-intention combination, as well as an associated affect value, termed *insistence* (Davis 2004). This association value is one of the main factors in deciding, at the deliberative level, a cognitive agent's next action. A cognitive agent maintains various beliefs about the environment it is operating in, and several possible goals that relate to that environment or the objects within it. It also has a number of different plans of actions or intentions or behaviours. Associations provide a method for a cognitive agent to keep track of all these possible belief-goal-action combinations, as well as containing a key value that indicates their relevance and significance to the agent at a given time – their insistence. These combinations detail the correct action required to achieve a specific goal given a specific belief. Associations take the following form:

```
association ( Belief, Desire, Intention, Insistence ).
```

The associations can be pre-defined (pre-programmed or coded) prior to runtime (typically a small number related to high priority tasks for specific environment configurations) or generated and formed when the architecture is initialised, or dynamically created by using all possible B-D-I combinations.

When there is a list of associations, the agent will extract only those that have a belief-desire combination that correspond to the current stated belief and goal of the agent. Then, of the remaining associations, the one with the highest insistence is chosen, which will give the intention or action or behaviour to be taken. The association value is then modified and updated based on whether the corresponding action failed or succeeded to achieve the agent's goal. If it failed, the value would be reduced; if it succeeded, the value would be increased. Therefore, association values fluctuate and are based on feedback from the agent's previous actions (Gwatkin, Davis 2007). Based on this principle, when CAMAL dynamically generates associations, it can then learn which combination makes more sense (meaning which association has a higher insistence and thus likelihood of success) for the current task set and environment. This process ensures that successful associations will develop higher association values, while unsuccessful associations will develop lower ones.

3.6 Motivational Blackboard and Motivators

The term *blackboard* is a metaphor for *global* context; meaning the blackboard is like a global workspace and communication structure that holds the relevant information about an agent's environment, attributes, properties, beliefs, desires, current state, previous states, etc. (Corkill 1991). This metaphorical blackboard, analogous to Baars' theatre metaphor and global workspace (Baars 1997) is potentially accessible by all the processes of an agent.

Using blackboard systems is a well-established approach to solving problems of control, cooperation, and communication between different agents and problem solvers of a cognitive or expert system. During 1991, the blackboard system was considered an 'emerging technology'. Gray et al. (1991) predicted a meta-level expert system, with the aid of a blackboard system, to share information between the individual expert systems. Blackboard architectures are even adopted in the context of genome annotation projects (e.g. Descorps-Declère et al. 2006) where they allow methodological and biological knowledge to be updated.

A blackboard system consists of three components: the blackboard, the knowledge sources, and the control component (Corkill et al. 1986). The knowledge sources are independent modules that contain the knowledge and information needed to solve some problem. They can access the blackboard or specific areas on it, extract the relevant information, manipulate it, and then post their contributions back on the blackboard, so that the results are again accessible to all the processes. These posts could be anything, such as an updated belief, the addition or deletion of a goal, a preferred intention, etc [5].

The knowledge and information held on the blackboard can be divided into several distinct areas: beliefs that the agent can have about its environment, desires that the agent can have about the objects in its environment, intentions that the agent can have to achieve its goals, associations that are used to manage the BDI and affect models, and a *motivator* that contains the result of the operation and execution of the various knowledge sources of the blackboard.

[5] Sometimes knowledge sources may be restricted to accessing and modifying only specific parts of the blackboard.

The motivator construct is such an important part of the deliberative processes that the blackboard is usually referred to as the *motivational blackboard*. The motivational blackboard not only holds the relevant knowledge and allows the various cognitive processes access to the information they require to carry out their tasks, but also controls and coordinates the flow of information through the architecture. Put differently, the focus of a cognitive architecture is often on the motivator construct as a representational form that enables perception, affect, cognition, and behaviour to interact (Davis 2010). In other words, a motivator is a representational schema that is used as a generic framework to bring together many aspects of perceptual and cognitive processing, such as perception, affect, cognition, and behaviour.

As stated before (see 2.8.2), the use of control states to develop cognitive architectures led to the use of affective and motivational control states. These control states are managed by the use of motivators. As a matter of fact, motivators not only manage and manipulate the affective and motivational control states present at the deliberative level, but may also trigger appropriate intentions or actions or behaviours at the reactive level, e.g. selecting a suitable reactive sub-architecture [6]. Motivators take the following form:

```
motivator ( Goal, Association, Deterministic, Cycles, Intensity ).
```

The `Intensity` element is an affect value that gives the importance of the motivator to the agent (see 2.8.2). The schema also contains the agent's chosen goal plus the appropriate association chosen to achieve that goal, which contains the intention of the agent as well.

The `Cycles` element gives the number of cycles that the reactive component should run for. This value is dependent on the agent's goal and is a component of the domain model; meaning it is pre-defined prior to runtime. The `Deterministic` element is set to either `false` or `true`. If set to `true`, the motivator would override any failure in achieving a goal until the end of the cycle or until the goal is met.

[6] Reactive sub-architectures are akin to actions, intentions, or behaviours. They are designed to achieve specific reactive tasks, e.g. obstacle avoidance.

Meaning, the motivator becomes strictly goal-oriented and determined, until it achieves the goal or runs out of cycles. If set to `false`, the motivator would return the first possible action that the deliberative-reactive interface selects, based on the appropriate goal-association combination; meaning the motivator tries to achieve the goal, but if failed would exit even if there are still cycles left. Once a goal and a relevant association are chosen, the motivator-update knowledge source of the motivational blackboard uses them to update the motivator.

The control component of a motivational blackboard, generally, manages the course of problem-solving and makes run-time decisions about the expenditure of problem-solving resources. The control component of CAMAL is comprised of the reasoning and updating mechanisms that build a motivational construct. The motivational construct is, thus, a representation of the outcome from those reasoning and updating mechanisms.

3.7 Domain Model

In addition to the BDI and affect and motivational models, CAMAL has a domain model. The domain model consists of all the elements that describe some part of the cognitive agent's physical form or environment (real environment or simulation testbed). It is vital that relevant information about the cognitive agent's attributes and properties along with its surrounding environment be incorporated into its cognitive architecture. This incorporation, which is partly done through encoding (pre-programming) is achieved by the use of a domain model. In addition to this method of instantiating the relevant information into the cognitive architecture, CAMAL also allows an extended domain model where learning occurs through dynamical association generation.

The inclusion of a domain model makes the BDI and affect and motivational models *generic*, as they can be implemented on any agent in any environment – all that is needed would be the use of an appropriate (given or learnt) domain model to instantiate the relevant information about the agent and its enclosure or terrain into its cognitive architecture.

The domain model is distributed across the entire cognitive architecture, at both the reactive and deliberative levels. It provides information on the agent's attributes and its surrounding environment; it defines the types of objects to be found within the agent's environment (simulation or physical); it defines some of the possible beliefs that are most relevant to the agent; it defines the relationships between the stated beliefs; it defines the goals that the agent can have; it provides a list of all the possible actions the agent can undertake; it provides the objects' perceptual profiles (i.e. the information required to recognize an object), etc.

There are several components of the domain model that need to be introduced at this point, as they are vital to understanding how the architecture operates (for the rest, see appendix I). Statements about the agent's environment at the deliberative level pertain to the possible beliefs the agent can have. These constitute the beliefs used in the BDI model. These statements, whether for simulated or embodied agents, are present on the motivational blackboard. These belief statements are represented by clauses of the form:

belief (Descriptor, Source, Time).

This particular syntax represents beliefs as categorical states. They can be prioritized only by CAMAL preference model, using belief preference operators based on their source (see appendix I). However, they cannot be adequately valenced via affective values and affordances.

Statements pertaining to the agent's attributes are present at the deliberative level. These statements are on the motivational blackboard, and provide information on the goals that the agent can have and the actions that the agent can take. These constitute the desires and intentions used in the BDI model. Goals take the following form:

goal (Descriptor, SuccessCondition, Importance).

The *Descriptor* is a description of the agent's goal. The *SuccessCondition* is the belief descriptor required for the goal to be achieved. The *Importance* is an affect value detailing the goal's affordance. It controls how important the goal is to the agent.

Intentions are plans of actions – a set of intended tasks – that can be inferred from beliefs and desires of an agent, or explicitly provided (encoded / pre-programmed). They are akin to plans of actions or behaviours or reactive sub-architectures that are based on the agent's desires, and have access to the agent's beliefs. If the environment changes and the agent's beliefs are updated, then the agent's intentions should be modified accordingly to prevent the agent from failing its goal(s). The architecture at any one time makes use of only one of a number of alternative reactive sub-architectures. The BDI model is responsible for selecting the agent's current goal (desire) and the intention (reactive sub-architectures used to establish that goal). For the majority of work in simulation and robotic worlds, variations of an initial domain model and motivational blackboard have been used (see appendix I).

In summary, CAMAL is built around the concept of control states and, in particular, motivators. The primary conjecture is that the further development, design, and implementation of such cognitive architectures can proceed using a consistent and systematic control language across all aspects of reasoning and domain model management – something that the development of CernoCAMAL has taken advantage of. This control language can be grounded in the use of *affect and affordances* with the aim to be consistent across different domains, tasks, and levels of processing (Davis 2010). Together, they can be used to guide both internal and external decision-making and activities.

3.8 Operational Overview

CAMAL uses a BDI model to drive a motivational blackboard. Perceptual updates lead to belief formations, and then belief revision in the BDI model, which then gives rise to goal revision. The association-update then uses the new belief set and goal set to determine the relevant action or intention or behaviour. The motivator-update enables goal revision and the selection of the next goal, based on goal importance (urgency) and goal success and current beliefs. This, in turn, drives motivator revision using the association construct, which in turn enables belief-desire-intention combinations to be ranked based on the likelihood of their success (association value or insistence). The goal urgency, association insistence, and motivator intensity are all underpinned by *affordances*; i.e. they are all consistently grounded in *affect*.

Therefore, affect and affordances are the means by which the agent can weigh its processes and also control the economics of its reasoning and processing. Together, they allow motivators to persist or be updated by new goals, associations, etc.

A more extensive implementation of CAMAL allows for the use of metacontrol and metacognitive control processes (Venkatamuni 2008). Consulting Figure 3.4 again illustrates the column that conceptualizes the metacontrol and metacognitive control processes (reflective layer). It, also, illustrates how the BDI model is represented as the arcs from the perception processes across the motivational blackboard associated with the BDI hierarchies in the fourth column of the architecture schematic.

The architecture, BDI model, and motivational model are configured at run-time using various domain models that allow the architecture to instantiate itself in a specific configuration. The processes interact through the use of affect and affordances. For any particular configuration of the architecture, there is a default behaviour, e.g. avoidance. This default behaviour (or possibly a set of default behaviours) is determined by the deliberative architecture – an example of many interactions taking place between the various layers of CAMAL. For a specific domain model, there are several behaviours associated with it; which represent intentions in the BDI model, prioritized and ranked by the affordances set by the affect model.

3.9 Representative Cognitive Architectures

This chapter began with an introduction to CAMAL and a focused literature review on it, as the cornerstone and pivotal axis of this work (CernoCAMAL). It is now time to reflect on RoboCAMAL to complement this review.

3.9.1 RoboCAMAL

RoboCAMAL was the most recent achievement of CAMAL research (Gwatkin 2009). The robot that runs RoboCAMAL is an ActiveMedia AmigoBot mobile robot, sketched in Figure 3.8 below.

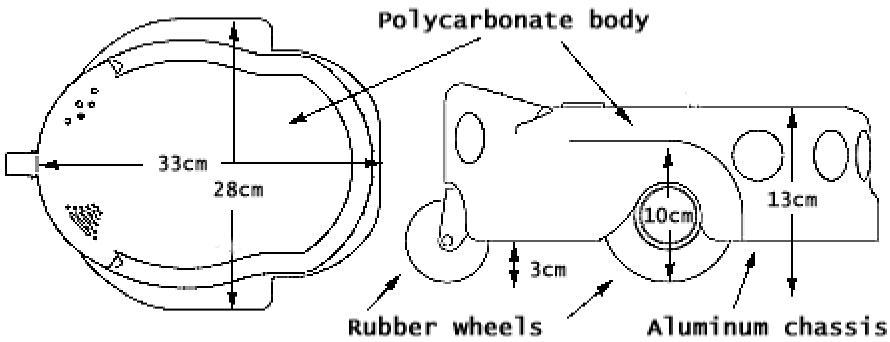

Figure 3-8: ActiveMedia AmigoBot Mobile Robot (ActiveMedia 2007)

The robot's environment is an enclosed area approximately two meters square. There is a one-meter partition wall roughly halfway along one side. There are various possible objects in this closure that RoboCAMAL's vision system is designed to detect: blue ball, red robot, black robot. The black tape alongside the base of the walls is meant to aid the vision system by increasing the contrast between the walls and the floor. See Figure 3.9 below.

Figure 3-9: RoboCAMAL's Enclosure (Robotics Lab, Hull University)

RoboCAMAL has eight sonar sensors (4 facing forward, 2 rear-facing, and 1 on each side) and communicates with a Linux desktop via a pair of wireless radio modems (InfoWave modems). RoboCAMAL is controlled using the ARIA software suite (see 5.3). ARIA consists of a set of functions that access the information sent by the server platform (i.e. the robot). This enables the user to define a set of actions, called micro-behaviours. These micro-behaviours can be added to the main robot connection loop, which will enable them to have access to the sensor buffers.

Each micro-behaviour is fired once every processing cycle, and hence can access the relevant sensory data once every processing cycle. Each micro-behaviour uses the sensory information to produce an action. Each action consists of a desired heading and a desired speed that will be used by the robot's specific functions to control the left and right motor speeds. The micro-behaviours are all *reactive* in nature. Once a micro-behaviour has been fired, it produces a set of actions that it wishes to carry out. These desired actions from each fired micro-behaviour are placed onto an *action list.* The next step is obviously to choose an appropriate action from the action list, and subsequently an attempt to carry it out.

The module that determines the specific action to be sent to the robot via the wireless radio modem to be performed is the *resolver module.* The original resolver has been modified to include three extra arbitration methods in addition to ARIA's indigenous arbitration method (ArPriorityResolver). What happens in reactive RoboCAMAL is that the robot's sensors gather information and pass it to the sensor buffers, via the radio modem and the USB cable of the camera. The *perception module* uses the information in the sensor buffers to determine whether an event has occurred. If so, feedback messages are sent to the deliberative component. The deliberative component processes this information and returns appropriate control messages.

The *configuration module* uses these control messages to configure the sensor buffers to the appropriate object profile, and also configure the resolver module to use an appropriate arbitration method. It, then, determines an appropriate reactive architecture, i.e. an appropriate combination of reactive micro-behaviours. Following this, active micro-behaviours fire and produce a list of desired actions. This list is used by the resolver module to determine the final action. The action command is, then, sent to the robot via the radio modem. Upon receiving the command, the robot performs that action.

In addition to the sonar array, RoboCAMAL has been fitted with an omni directional vision system – ODV device (Gwatkin & Davis 2007). Omni directional vision involves the capture and interpretation of images that depict full 360 degrees view of the surroundings. It offers the convenience of dealing with the rotation of the camera, or the robot equipped with the camera, as this will not make objects disappear from view, but only change their relative image positions. For reasons of simplicity and low cost, the ODV device in RoboCAMAL has been implemented using a single web-camera and a single spherical mirror bolted on top, depicted in Figure 3.11. Once an omni directional image has been captured by the camera and sent to the client (Linux machine) via the USB cable of the camera, it needs to be interpreted. An important tool used in image interpretation is image *segmentation*. The main goal of image segmentation is to define regions within an image that correlate with objects in the physical environment. There are various techniques used for image segmentation, such as thresholding algorithms, edge-based algorithms, region-based algorithms, etc. Since RoboCAMAL's vision system must be fast, it adopts a Sobel edge-detection technique along with a simple thresholding method.

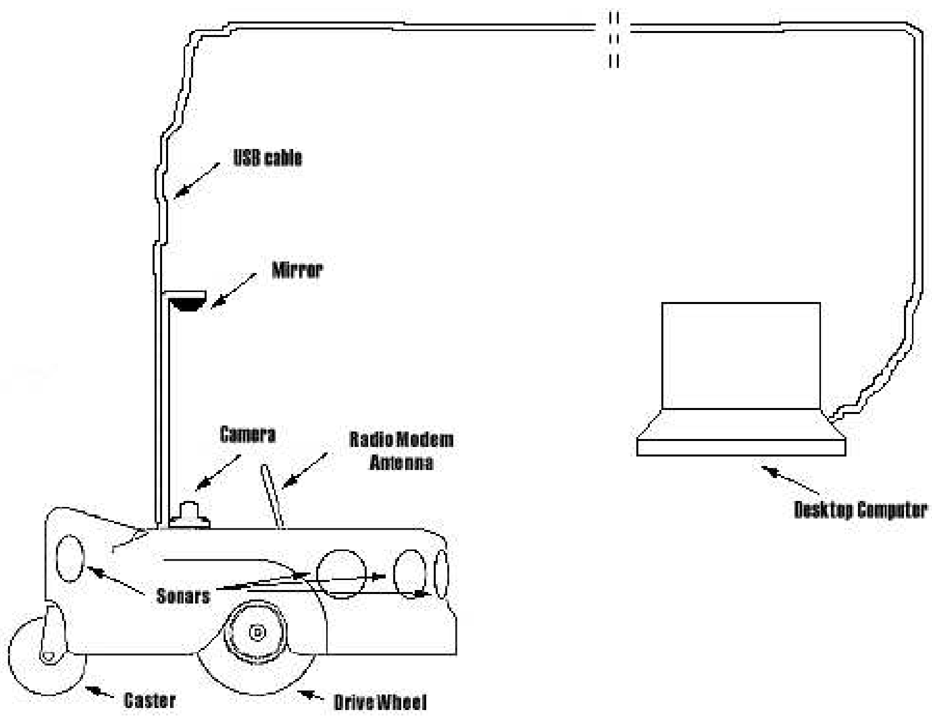

Figure 3-10: RoboCAMAL's Omni-Directional Vision (Gwatkin and Davis 2007)

Once a captured image is transformed into an edge image by running a Sobel operator over it, the formed edges need to be linked together to construct a continuous node chain. This chain will then be used to separate one area from the next. RoboCAMAL's segmentation algorithm actually investigates an area of interest (AOI) within an image. The AOI is defined as an area around the robot. It is this area that is analysed to identify objects – objects that RoboCAMAL is designed to detect.

In summary, RoboCAMAL observes its environment and forms beliefs (environmental beliefs). It, then, chooses an appropriate action based on those beliefs, its internal state, and its pre-set goal(s). It, finally, executes that action. Robo-CAMAL's anchoring mechanism deals with the formation of beliefs; the BDI schema and the Affect model are responsible for choosing an appropriate action; and the reactive level is responsible for interpreting the control messages received from the deliberative layer and executing that action. The RoboCAMAL research project was successfully concluded in 2009.

The research has now gone into the applications of complete and believable cognitive agents for simulated training environments (e.g. Tambe et al. 1995), computer tutoring systems (e.g. Koedinger, Anderson, Hadley, Mark 1997), and interactive computer games (e.g. Magerko, Laird, Assanie, Kerfoot, Stokes 2004). In particular, as Langley et al. (2009) point out in their review of research issues and challenges for cognitive architectures, over the past few decades at least three invited symposia have brought together researchers in this area (Laird 1991; VanLehn 1991; Shapiro, Langley 2004) and there have been at least two edited volumes (Sun 2005; VanLehn 1991). Many researchers have proposed and developed cognitive architectures over the past four decades.

As it is impossible to survey the entire space of cognitive architectural theories, only the following six distinct frameworks are briefly reviewed in this thesis, due to two reasons: Firstly, because they have appeared in the literature with reasonable frequency. Secondly, because they have exhibited different degrees of concern with explaining human behaviour. There is a variety of other frameworks as well, but the representative sample reviewed here is an excerpt taken from the highlights of Langley et al. (2009) that gives some intuitions about the space of cognitive architectures.

No connectionist networks, exemplar-based models, or production systems approaches are included in this thesis though, since none has yet demonstrated the broad functionality associated with cognitive architectures in the sense discussed in this work. They have, on occasion, served as important components in larger-scale architectures, as in Sun et al. (2001) CLARION framework. But as must be clear by now, this work focuses on a symbolic account of cognition. The following describes some representative cognitive architectures that serve as concrete examples, reported in the literature with reasonable frequency. Note that CogAff was previously introduced and reviewed at the beginning of this chapter as an inspiration to the proposal and development of CAMAL.

3.9.2 ACT-R

(Adaptive Control of Thought – Rational) (Anderson 1976; Anderson, Lebiere 1998; Anderson, Matessa 1998; Anderson et al. 2004; Anderson 2007) is a cognitive architecture realized as a symbolic goal-oriented production system, written in Lisp. It is primarily concerned with modelling human behaviour and semantic memory. It has seen continuous development since the late 1970s and is, therefore, cited frequently as a well-known example of a cognitive architecture. The most recent instantiation of it includes a declarative memory for facts and a procedural memory consisting of production rules. The architecture operates by matching productions on perceptions and facts, mediated by the real-valued activation levels of objects, and executing them to affect the environment or alter declarative memory. Learning in ACT-R involves creating new facts and productions, as well as updating base activations and utilities associated with these structures.

3.9.3 SOAR

(Newell 1986; Laird et al. 1987; Rosenbloom 1993; Lewis 2001) encodes procedural long-term memory as production rules, whereas working memory contains a set of elements with attributes and values. The performance system matches productions against elements in working memory, and generates sub-goals automatically when it cannot continue. When processing in the sub-goal lets the agent overcome this impasse, the architecture adds a new ‘chunk’ to long-term memory that summarizes the sub-goal processing.

In recent versions, episodic and semantic learning store working memory elements as structures in long-term memory, while reinforcement learning alters weights associated with rules that select operators.

3.9.4 CLARION

(Sun et al. 2001; Sun 2007; Sun 2009) stores both action-centred and non-action knowledge in implicit form, using multi-layer neural networks, and in explicit form, using symbolic production rules. Corresponding short-term memories contain activations on nodes and symbolic elements that the architecture matches against long-term structures. Performance involves passing sensory information to the implicit layer, which generates alternative high-value actions, and to the explicit layer, which uses rules to propose actions. The architecture then selects the candidate with the highest expected value. Learning involves weight revision in the implicit system, using a combination of reinforcement learning and back-propagation to estimate value functions, and construction of production rules by extraction from the implicit layer, error-driven revision, and instantiation of rule templates.

3.9.5 GLAIR

(Shapiro, Ismail 2003) stores content at a knowledge or cognitive level, a perceptual-motor level, and a sensory actuator level. The highest layer includes generalized structures that define predicates in logical terms, ultimately grounding abstract concepts and procedures in perceptual features and behavioural routines at the middle layer. The system supports inference, belief revision, planning, execution, and natural language processing, inferring high-level beliefs from perceptions and deriving commands at the sensory actuator level from the agent's goals and plans.

3.9.6 ICARUS

(Langley, Choi 2006; Choi 2011) represents long-term knowledge in separate memories for hierarchical skills and concepts, with short-term beliefs, goals, and intentions cast as instances of these general structures. The performance element first infers all beliefs implied by its concepts and its perceptions of the environment, then selects an applicable path through the skill hierarchy to execute.

Means-ends problem solving occurs when no skills relevant to the current goal are applicable, whereas learning creates new skills based on traces of successful problem solving.

3.9.7 PolyScheme

(Cassimatis, Trafton, Bugajska, Schultz 2004) is a cognitive architecture designed to achieve human-level intelligence by integrating multiple representations, reasoning methods, and problem-solving techniques. Each representation has an associated specialist module that supports forward inference, and other basic operations, which match against a shared dynamic memory with elements that are grounded in perception and action. PolyScheme makes a stronger semantic commitment than most architectures, encoding all structures with a basic set of relations about time, space, events, identity, causality, and belief.

3.10 Summary

This chapter began with an introduction to a general class of computational cognitive architectures known as CAMAL. A review of the original overarching architecture was presented, as an important pivotal axis for the development of CernoCAMAL. It was described what progress had been made since the start of the CAMAL research. Its priors (e.g. CogAff) and its spin-offs to date were introduced, and subsequently its components and organization were presented, including the BDI (Belief-Desire-Intention) and affect models. This was, then, followed by a brief discussion on the works of CRIBB and a-CRIBB reasoning models, and how their architectures made use of BDI and affect schemas. The chapter continued with introducing the association and motivator constructs. The concept of a motivational blackboard was presented, followed by explaining how CAMAL's blackboard facilitated the operation of its motivational constructs. The role of a domain model was highlighted, and its components were then pointed out. A brief operational overview of CAMAL was given before proceeding to a broader literature review in the field of Cognitive Architectures. The chapter concluded with reviewing the most notable and significant examples of cognitive architectures, in particular the recent work of RoboCAMAL.

4 CernoCAMAL Architecture

This chapter begins with a brief overview of relevant research conducted prior to CAMAL and CernoCAMAL, with regards to probabilistic thinking and reasoning. The fact that none of the CAMAL spin-off projects to date has addressed the cognitive capability of probabilistic reasoning is underlined, followed by emphasizing the main motivation of extending CAMAL to develop this fundamental cognitive ability. A brief introduction to the probability theory is given, followed by a review of the basics of the probability calculus and framework. The structure of CAMAL's belief statements is explored, from which the CernoCAMAL's Extended Belief Structure (EBS) is developed. The EBS is discussed and its advantages over CAMAL's inadequate belief statements are pointed out. The chapter then goes on to introduce CernoCAMAL's Probabilistic Reasoner (CPR) and explaining how it reasons probabilistically over the feedback generated by reactive sub-systems of CernoCAMAL. A comprehensive design criteria for the belief predicates constituting goal- and task-oriented feedback generated by reactive sub-systems is outlined. The memory facility used in the design of the CPR is addressed, followed by the norm facility incorporated in the structure of motivators. The chapter concludes with presenting an operational overview of CernoCAMAL.

4.1 Introduction

Since the 1950s, Cognitivism (also known as Cognitive Psychology) has been the predominant perspective within which human learning research has been conducted and cognitive theories of human learning have evolved. The roots of human interaction and learning theories can be found in research dating back to the 1920s and 1930s. For example, Jean Piaget the famous Developmental Psychologist, portrayed children as active and motivated learners who, through numerous interactions with their physical and social environments, construct an increasingly complex understanding of the world around them (Piaget 1928). Piaget and Inhelder (1951) later linked the development of children's probabilistic thinking and reasoning to Piaget's general theory of cognitive development (Way 2003).

In more recent years, a number of researchers (e.g. Fischbein 1975; Shaughnessy 1981; Green 1983; Peard 1995; J.Truran 1996; K.Truran 1996; Fischbein, Schnarch 1997) have contributed to the growing body of knowledge in regards to probabilistic thinking and reasoning. There is now considerable evidence that probabilistic thinking and reasoning is linked to cognitive development and plays a role in cognitive functions, such as decision making and learning.

It is also noteworthy that similar to child's development, the architecture of an intelligent system is not static – it develops over time. Put differently, the behaviour of an intelligent agent depends on the challenges that arise from being situated and embodied, and being in interactions. It is a product of the on-going development of the agent in interactions with its environment. New learning and reasoning bring about new capabilities which, in turn, modify the capabilities for further learning and reasoning.

All this evidence leads us to believe that a probabilistic reasoning capability is an essential part of human intelligence. Thus, it should be a vital part of any system that attempts to emulate human intelligence computationally. In other words, probabilistic reasoning is an essential aspect of the process of cognition and, therefore, must be considered in any adequate description of it. This is the core motivation for the design, development, and implementation of CernoCAMAL, as this cognitive capability has never been implemented or included in CAMAL before. CAMAL has pursued a perspective informed by affective and motivational control states, rationalized by cognitive models of reasoning and learning. There clearly exists a need to incorporate probabilistic reasoning.

The development of the current cognitive architecture under investigation – CernoCAMAL – is mainly based on projects and experiments in the theory, design, and implementation of affect- and motivation-based architectures of Davis (Davis 1996...2010) (see 3.1). It builds on the work of Simon's Control State theory (1967, 1996, 1999) and is conducted within a conceptual framework. The viewpoint of *'mind as a control system'* is a major constituent of this conceptual framework (Sloman 1993). Another comprising part of the CernoCAMAL's conceptual framework is the viewpoint that *'architecture dominates mechanism'*. It sums up the view that architecture has a greater influence on the capacities of an intelligent system than the mechanisms it consists of (Sloman 1993).

4.2 Probability Theory

How best to reason about uncertain situations has always been of concern to humans. Rationality and rational reasoning and decision making, however, have always seemed to concern reasoning according to the rules of logic. Piaget viewed logical reasoning as defining the end-point of cognitive development. Even contemporary psychology of reasoning has focused on comparing human reasoning against logical standards. An example would be contradictory statements about uncertain situations and degrees of belief or doubt (such as negations existing simultaneously: $P \;\&\; \neg P$ which is false) which normally cause logic-based models of cognition to fail, whereas in probability calculus, $P(x) = 0.9 \;\&\; P(\neg x) = 0.1$ is acceptable.

Formal logic has its foundations in the studies of human rationality which is an evident characteristic of the human cognition. Cognition means many things to many people though. It is, however, taken necessarily to include the higher-level information processing stages that are carried out by the human and animal brains, such as thinking, reasoning, learning, planning, looking ahead, and even consciousness (Blumenthal 1977). Reasoning and learning are fundamental and crucial components of cognition (Taylor 2005). There are a number of common approaches to modelling the human mind and cognition, such as:

- Symbolic: performing logical inferences on symbols, e.g. language structures.
- Connectionist: obtaining inferences from ANNs at a sub-symbolic level.
- Probabilistic: performing probabilistic inferences and using the results for reasoning and, possibly, learning and adaptability.

The probabilistic approach is key here. Probabilistic thinking and reasoning is shown to be linked to cognitive development, based on a substantial body of research (e.g. Piaget 1928; Piaget, Inhelder 1951; Fischbein 1975; Shaughnessy 1981; Green 1983; Peard 1995; J.Truran 1996; K.Truran 1996; Fischbein, Schnarch 1997; Way 2003; Oaksford, Chater 2007). In addition, the probabilistic approach to human cognition has become established. Oaksford and Chater argue that cognition is better understood in terms of probability theory – the calculus of uncertain reasoning, rather than in terms of logic – the calculus of certain reasoning (Oaksford, Chater 1999, 2007, 2009).

In *Bayesian Rationality* (Oaksford, Chater 2009) the case is made that cognition in general and human everyday reasoning in particular are better viewed as solving probabilistic, rather than logical, inference problems. It is also argued that rather than viewing people as flawed logicians, the focus should be instead on the spectacular success of human reasoning under uncertainty. The basic formalism of probability calculus has been available from the 17^{th} century, and provides a promising framework for coping with uncertainty. As a matter of fact, the probability calculus was specifically invented in the 17^{th} century by Fermat and Pascal in order to deal with the problems of physical uncertainty introduced by gambling.

But it did not take long before it was noticed that the concept of probability could also be used as an approach to scientific reasoning about uncertainty. It, then, quickly took on a larger and deeper significance as a formal framework for how rational cognitive agents should reason in situations of uncertainty. A cognitive agent running the CAMAL architecture, however, assumes that propositions are either true or false or unknown, unless it is equipped with some process of assigning degree or extent or strength of beliefs and then handling the probability computations, as this cognitive capability has never been implemented or included in CAMAL before (addressing the main motivation for the design, development, and implementation of CernoCAMAL).

An appropriate set of questions at this point would be whether we should bother with uncertainty at all; whether we need to take into consideration the uncertainty inherent to a cognitive agent's environment; and whether, specifically, the CAMAL belief propositions mechanism (true, false, or unknown) is really inadequate for a cognitive architecture. It must be noted that reasoning about any realistic domain always requires that some simplifications be made. The very act of preparing knowledge to support reasoning (such as reasoning over a set of beliefs) requires that we leave many facts unknown, unsaid, or crudely summarized. For example, using a domain model to instantiate into CAMAL's cognitive architecture knowledge about its environment, beliefs, goals, and tasks will have many details and exceptions which we simply cannot afford to enumerate – not to mention the conditions under which these rules apply! An alternative to the extremes of 'ignoring' or 'enumerating' all details and exceptions is to use reasoning under uncertainty.

Also, it must be noted that when humans reason rationally, they are actually using a BDI model and an affect model, along with an underlying motivational schema; all together and asynchronously. This, however, is not the complete picture of reasoning rationally. Performing probabilistic reasoning and inference, and allowing degrees of belief or doubt to outcomes of events are also employed in reasoning and learning processes. This is evident in our daily lives as well.

4.3 Basics of Probability Calculus

The probabilistic inference framework can provide a formal, general approach to computing posterior probabilities given some apriori. This section briefly introduces the basics of probability calculus. First, a few notations:

$P(x)$ denotes probability of x.

$P(x \mid \theta)$ denotes probability of x given θ.

$P(x, \theta)$ denotes joint probability of x and θ.

Probability is the measure of how likely an event is. An event is one or more outcomes of an experiment. An outcome is the result of a single trial of an experiment. An experiment is a situation involving chance or probability that leads to outcomes and results. Let U be the universe of possible events. Then, the maximum probability must apply to the *true* event lying within U. By convention, this maximum probability is set to 1 resulting in the first axiom for probability:

$$P(U) = 1$$

Equation 4-1: Axiom One

This probability mass always sums or integrates to 1. For simplicity, it is assumed that this probability mass is evenly distributed or spread over U, so that the probability of any region is proportional to its area. Obviously, for any region such as x its area cannot be negative (even if x is empty) resulting in the second axiom for probability:

$$\forall x \subseteq U, P(x) \geq 0$$

Equation 4-2: Axiom Two

The third axiom[7] results from the need to compute the probability of combined events x and y:

$$\forall x,y \subseteq U, P(x \cup y) = P(x) + P(y) - P(x \cap y)$$

Equation 4-3: Axiom Three

In summary, a probability is a measure over a set of events that satisfies the above three axioms. The Bayesian probability assumes belief measures obey these three basic axioms (Pearl 1988, 2003). The basic expressions in the Bayesian probability are statements about conditional probabilities. Thomas Bayes (1702 - 1761) was an 18th century minister who made his main contribution to the science of probability by associating the phrase "`... given that I know Y`" with the now-famous ratio formula, which has become the definition of conditional probabilities. He introduced a key theorem which serves as the mathematical basis of probabilistic inference:

$$\forall x,y \subseteq U, P(x \mid y) = \frac{P(x,y)}{P(y)} = \frac{P(x \cap y)}{P(y)}$$

Equation 4-4: Theorem One

The heart of Bayesian techniques lies in the inversion formula:

$$P(h \mid e) = \frac{P(e \mid h) * P(h)}{P(e)}$$

Equation 4-5: Bayes' Rule

The Bayes' rule states that the belief we accord a hypothesis h upon obtaining evidence e can be computed by multiplying our previous belief $P(h)$ by the likelihood $P(e \mid h)$ that e will materialize if h is true (Pearl 1988). $P(e)$ is just a normalizing factor. $P(h \mid e)$ is called the *posterior* probability of h given e, and $P(h)$ is called the *prior* probability of h.

$$posterior = \frac{likelihood * prior}{evidence}$$

Equation 4-6: Probabilistic Inference Theorem

[7] It is noteworthy that in some text books, a simplified version of this (where x and y are mutually exclusive) is introduced as the Third Axiom, and then the complete version is presented as a Theorem or Consequence. Another consequence that is commonly used is: $P(\neg x) = 1 - P(x)$

The posterior probability $P(h|e)$ could also be interpreted as the *degree of belief* in h conditioned on the observation of e. Probability calculus provides a powerful tool for expressing uncertainty in a cognitive model or domain. In the context of Cognitive Science and Artificial Intelligence however, probability refers not to 'objective' facts about gambling devices or anything else, but rather it describes a reasoner's degrees of belief. Probability theory is, thus, a calculus not for solving mathematical problems about objects in the world, but for expressing degrees of belief and subsequently updating beliefs accordingly. This perspective is the 'subjective' or Bayesian view of probability.

From its origins, probability theory was viewed as both mathematics and psychology. Reconciliation is, thus, long overdue between the mathematics of probability as a vital tool in building theories of cognition and the psychology of probabilistic reasoning and inference (Chater et al. 2006). Probabilistic models of cognition such as CernoCAMAL provide a framework for reasoning and updating beliefs realistically. Moreover, they can be applied in various ways to the problems that a cognitive system faces, such as the problem of incorporating degrees of belief in their BDI model.

In summary, many – if not all – aspects of human cognition depend fundamentally on inductive inference: evaluating degrees of belief in hypotheses given weak constraints imposed by observed data (Tenenbaum, Mozer 2000). In logic-based models of cognition, the currency of belief is a *binary truth value*. In connectionist models of cognition, the currency of belief is an *activation level*. In probabilistic models of cognition such as CernoCAMAL, the currency of belief is a *probability value*. Generally, in Cognitive Science and Artificial Intelligence applications, probabilities refer to degrees of belief or belief affordances. Thus, a cognitive agent's degree of belief that a particular entity is a specific object (say might be *P1*) might well increase to a greater number (say *P2*) as new perceptual evidence or reactive feedback comes in. At the same moment, the experimenter knows the exact number of that specific object in the testbed. Thus, the two cognitive beings are viewing the very same event, but their belief states and hence their subjective probabilities might differ. This particular approach of combining prior information (apriori) and evidence is the subjective or Bayesian view of probability. As evidence accumulates, the degree of belief in a hypothesis ought to change accordingly.

It is noteworthy that this inference method may be biased due to initial beliefs (default assumptive beliefs, such as `environment(sparse)`) that the cognitive agent holds before any evidence or perception or reactive feedback is ever collected. This form of inductive bias, though valid, does not detract from the calculated degrees of belief over time, as these belief affordances are after all numerical 'estimates' that a cognitive agent works out and are always subject to modification.

4.4 Extended Belief Structure (EBS)

In CAMAL beliefs are core. They are represented by clauses of the form:

```
belief ( Descriptor, Source, Time ).
```

It is, then, the BDI and affect models that determine the intentions, actions, or behaviours of the agent based on its beliefs and goals. One of the limitations of the BDI model, however, is the lack of any explicit mechanism to express the degree to which the belief statement `Descriptor` is believed to be true. Since affect is used across the cognitive architecture as a *decision metric*, affective values can be thought of as a *currency*. Meaning, since affect serves as a *decision metric*, affective values can serve as a *currency*. Put differently, the BDI model lacks an affective decision metric consistent with the affordances used in the affect and motivational models. Given that the current CAMAL research presents an affect- and affordance-based core for mind (Davis 2010) it seems reasonable to conjecture that beliefs, too, should be grounded in the use of affect, with the aim to be consistent across different domains, tasks, and levels of processing. This can be achieved by extending the belief structure to incorporate probabilities as degrees of belief associated with different information sources. The extended belief structure associates a probability value `DegBel` with every belief statement in CernoCAMAL which defines the degree to which the belief statement is believed to be true:

```
belief ( Descriptor, Source, Time, DegBel ).
```

The `DegBel` element represents the degree of `belief` in the plausibility of `Descriptor`. Obviously, the following belief functions must satisfy the rules of probability theory:

$0 \ll DegBel \ll 1 , \qquad DegBel = 0 , \qquad DegBel = 1$

`DegBel = 0` means definitely false and `DegBel = 1` means definitely true. Note that extent of beliefs are represented by real numbers, and there is a correspondence with common sense!

CernoCAMAL uses this extended belief structure to represent the degree or extent or strength of beliefs numerically and then manipulate them. The idea behind equipping the original CAMAL with such an extended belief structure is based on the fact that the uncertainty and incompleteness of information (e.g. perceptual information) can induce some uncertainty in the validity of its conclusions; i.e. formed beliefs. Therefore, every belief clause must be associated with an assessment of its plausibility or reliability – belief affordance. This extension will allow the entire BDI model to run using numeric affective values to prioritize choices over the current belief set. The use of such an affective or BDI model will make affordances the basis of a consistent and systematic control language across the entire cognitive architecture. It will also allow external events and objects to take valenced affordances, and internal mechanisms to be prioritized via valenced processes.

The resulting formalization of the BDI model enables CernoCAMAL to deal with dynamic environments, as demonstrated by experiments performed in simulation and real testbeds. It does so by facilitating the use of a consistent and systematic metric across all aspects of reasoning and domain model management. This way, the BDI model can run using numeric affect values similar to the affordance values employed in the affect and motivational models. The primary conjecture here is that the incorporation and implementation of the EBS can proceed using the same systematic control mechanism based on affordances. This control mechanism is, essentially, grounded in the use of affect. This has subsequently led to the development of a probabilistic belief reasoner for CernoCAMAL that can deliberate probabilistically over the feedback generated by the reactive layer.

In summary, by exploiting this new probabilistic belief-affordance structure in CernoCAMAL, the initial probabilities (apriori) can be specified as assumptive (default) degrees of belief; then as evidence about a situation is gathered by perceptual systems, that evidence can be propagated through the entire BDI model, revising and updating other beliefs and their affordances (degrees). This enables the architecture to select a focused belief set that mirrors its current activities, as highlighted by actions, objects, and agents referenced in a current motivator. By utilizing the EBS, the entire BDI model along with the affect and motivational models are 'pulled together' operating in a formalized manner.

4.5 CernoCAMAL Probabilistic Reasoner (CPR)

The development and implementation of the EBS in CernoCAMAL allows for the inclusion of degrees of belief for different information sources. The following is the initial domain model assumptions made over the apriori of various sources of belief:

```
degree_of_belief ( perception,      0.9   ).
degree_of_belief ( deduction,       0.7   ).
degree_of_belief ( assumption,      0.5   ).
```

It is also noteworthy that a belief descriptor could be the apriori probability of an object being present in the environment:

```
apriori_prob ( Object, DegBel ).
```

which is clearly in the form of assumed knowledge. CernoCAMAL's CPR, given the list of domain model objects and assumed `degree_of_belief` for various sources of belief, can compute the posterior probabilities, assign them to the appropriate belief descriptors, and reason probabilistically about the number of objects and their instances that may be present in the environment [8]. As an example, consider a possible initial configuration of having one sphere, one prey, one predator, and no unidentified objects in the DND Tile World simulation environment (see 5.2). This configuration can be represented by the following statements:

[8] Note what 'uncertainty' is referring to here, in the context of the CPR using the EBS.

```
belief ( apriori_prob ( object, 0.0 ),    assumption, 1, 0.5 ).
belief ( apriori_prob ( sphere, 0.3 ),    assumption, 1, 0.5 ).
belief ( apriori_prob ( pred,   0.3 ),    assumption, 1, 0.5 ).
belief ( apriori_prob ( prey,   0.3 ),    assumption, 1, 0.5 ).
```

This initial configuration assumes that if there is one object, then it is equally likely to be a sphere, a prey, or a predator, hence the probability of every domain object being present is `1/3` or `0.3` with a `0.5` degree of belief since the statements are assumptive. Known exemplars of the above object classes are identified via *instances*, e.g. `prey2`. The feedback messages received from the reactive layer take the following form: `[ reactive_cycles ( N ) | Messages ].`

The deliberative layer uses the reactive feedback messages to construct updated beliefs. It, then, updates the time and activates the deliberative component with the updated beliefs. Table 4.7 illustrates possible elements constituting the reactive feedback list, accompanied by their interpretations and actions to be taken by CernoCAMAL's CPR. What follows afterward is a brief overview of the design criteria for the belief predicates constituting goal- and task-oriented feedback generated by reactive sub-systems. A detailed list of these predicates that are specific to the predator-prey testbed (see 5.2) can be found in the next chapter (see 5.4).

Feedback Item	***Meaning***	***CPR Action***
reactive_cycles	Number of reactive cycles [9]	Update reactive cycles.
fail	Current goal has not succeeded; i.e. architecture's instantiated behaviour has failed to achieve the current goal.	Display a message stating that the current goal-directed behaviour has failed.
non-belief clause	Invalid predicate; i.e. predicate does not belong to the belief predicate set.	Display a message stating that a non-belief predicate is not of interest.

Table 4-7: CernoCAMAL's CPR Actions

[9] The deliberative component may trigger the reactive component to run for 10 cycles. This means that for every one deliberative cycle, the reactive component will run for 10 cycles. The reactive component will, therefore, run for 10 cycles before calling the deliberative level, unless a significant event occurs.

The power of CernoCAMAL's CPR lies in its probabilistic interpretations of the predicates pertaining to various instances contained in reactive feedback lists. Note that the instances that were destroyed; i.e. eaten (see 5.2) and could not be re-generated when stumbled upon and the instances that were lost, but could be re-generated when stumbled upon. This intelligent probabilistic reasoning capability that keeps track of the instances is made possible by CernoCAMAL's `memory` facility.

4.5.1 Memory Facility

From a functional perspective, memory is seen as a capability for storage and retrieval of data. From an experiential perspective, memory is often thought of as episodes in which strong feelings of recall briefly dominate a person's awareness (Middleton, Edwards 1990; Nader 2003; Neisser, Fivush 1994). It is, however, difficult to justify separating memory from emotions and affect. Based on our daily lives and lived experiences, there is no doubt that affect can disrupt reasoning under certain circumstances, and that misdirected or uncontrolled emotions can lead to irrational behaviour. These feelings of affect strongly influence the way we *recall* as well. They influence what we remember and how we remember what it is that we remember. Memory, in essence, consists of much more than some local measurements of recall though. The experience of memory forms parts of on-going interactions that are emotionally-charged and are embedded in a broader context (Clocksin 2004).

CernoCAMAL's CPR `memory` facility is best understood by considering Cerno[10] in its dynamic and uncertain simulation environment. Now, suppose that Cerno starts moving around and stumbling across various objects in the testbed. Cerno's findings are represented as *reactive feedback lists* that are passed to the deliberative component for reasoning and deliberation. Having found a domain object or perhaps an unidentified one, Cerno reasons about the probabilities of various objects and their instances being present in the environment. Now, suppose that Cerno loses or eats (destroys) an item, e.g. a sphere or prey. That lost or eaten (destroyed) object or instance is withdrawn from the simulation world.

[10] Cerno refers to the implementation instance – the cognitive agent – whereas CernoCAMAL refers to the computational cognitive architecture. Put differently, Cerno is an instance of CernoCAMAL.

The way Cerno limits the scope of the propositions it must re-consider and re-evaluate in the light of its actions is by the loss' or destruction's *memory trail* that is formed by the CernoCAMAL's CPR. The object and instance reasoning of CernoCAMAL's CPR using memory trails is perhaps the most significant part of the probabilistic reasoning carried out in the deliberative layer of the architecture, as well as the most significant contribution of this work. It is for this reason that the greatest emphasis is placed on the tests and experiments that validate the correct functionality of the CPR (see 6.2.1 & 6.3.1). In a nutshell, when an instance is found [11] :

- If the instance list of that object is empty (i.e. if that object is definitely the first to be found), then a new unique instance is generated.

- If there is a memory trail of a previously lost instance, then that previously generated lost instance is re-created, instead of re-generating a new incremental one.

- If the found instance can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list), then it refers to that previously found instance.

- If the found instance has been found for a second time, then the first one refers to a previously found instance and the second one generates a new unique instance.

- If an instance was eaten, then the trail for that instance will be marked as 'destroyed' so that upon finding an instance, the identification is made based on the fact that it cannot possibly be that destroyed instance. Put crudely, an eaten object or instance cannot come back to life.

[11] Other predicates (such as near, hit, and touched) are also treated as found; meaning hitting something is akin to finding it.

4.5.2 Norm Facility

So far, a probabilistic reasoning capability has been incorporated in CernoCAMAL using the BDI model. This has enabled the inclusion of degrees of belief (belief affordances) and subsequently using the BDI and affect and motivational models to determine the agent's intentions, actions, or behaviours. It has, effectively, allowed the entire BDI model to run using numeric affective values to prioritize choices over the current belief set. The proposal to incorporate the EBS has also led to the development of the CPR that can deliberate probabilistically over the generated reactive feedback, using the memory facility.

Recall that a motivator contains the result of the operation and execution of the various knowledge sources of a blackboard. It is a representational form that enables perception, affect, cognition, and behaviour to interact. Put differently, it is a unifying schema that brings together many aspects of perceptual and cognitive processing, such as perception, affect, cognition, and behaviour:

(Extended Belief Predicate x Desire x Intention) +

Goal + Association → Motivator

Also, recall that at the deliberative level of CernoCAMAL, affective values and affordances can be associated with processes and predicates, and then relayed as control signals to instantiate and modify motivators and their associated representations and behaviours. Probabilistic formalism can be integrated into the motivational construct as well, by means of a *shallow metacognitive norm*. Metacognition can be simply defined as thinking about thinking (Wilson, Keil 1999). Any knowledge or cognitive process that refers to monitoring and controlling some aspect of cognition can be considered metacognitive.

This mechanism makes the motivator selection and updating a configurable probabilistic process. Therefore, the one motivator that is eventually chosen by deliberative processing not only reflects the result of the operation and execution of the various knowledge sources of the CernoCAMAL's motivational blackboard, but also performs in line with the BDI model and takes into consideration the associated degree of belief:

(Extended Belief Predicate x Desire x Intention) +

Goal + Association → Probabilistic Motivator

The described norm used in CernoCAMAL takes the form:

norm (motivator, selection, DegBel_Time_Desire_Intention).

Other Metacognitive norms that are used in CernoCAMAL include:

norm (belief, belief_decay_threshold, 15).

norm (goal, failed_goal_interval, 15).

The first involves the age of a belief. If a belief is older than (in terms of the elapsed deliberative time) a pre-defined threshold, then it is removed (here it is *15*). The second involves the consecutive number of times a goal has failed (here it is *15*). If so, then it is removed. These norms are used in experimentation to confirm that a probabilistic motivator yields a higher overall performance, success count, task effectiveness, and goal achievement. They provide a powerful tool towards developing efficient computational cognitive models.

4.6 Operational Overview

CernoCAMAL uses a BDI schema to drive a motivational blackboard. The blackboard, essentially, represents the whole deliberative component which has been written in Prolog. The inclusion of degree-of-belief (belief affordance) in the structure of CernoCAMAL's belief predicates enables the architecture to select a *focused* belief set that reflects its current activities, as highlighted by actions, objects, and agents referenced in a current motivator. The motivator enables goal revision and the selection of the next goal, based on goal importance and goal success and also current beliefs. The deliberative processing of these constructs allows the selection of an appropriate intention or action or behaviour related to specific objects and tasks. This, in turn, drives motivator revision using the association construct, which in turn enables belief-desire-intention combinations to be ranked based on the likelihood of their success or association values.

The belief degree (*belief affordance*), goal importance (*urgency*), association value (*insistence*), and motivator value (*intensity*) are all underpinned by *affordances*; i.e. they are all consistently grounded in *affect*. Therefore, affect and affordances become the means by which the agent can weigh its beliefs and processes, and also control the economics of its reasoning. Together, they allow motivators to persist or be updated by new goals, associations, etc.

At the reactive level, perceptual data from the testbed's sensors are passed to the deliberative layer. This perceptual message is posted to the motivational blackboard and reasoned about by the CernoCAMAL's CPR (see 6.2.1 & 6.3.1). The re-implemented belief-update module 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 set and goal set to determine the relevant action or intention or behaviour.

Whether in simulated or robotic testbeds, sensory information is mapped onto belief structures. Belief affordance or degree of belief defines the degree to which the belief statement is believed to be true. The insistence measure allows the control of external behaviour through the building of associations that link beliefs, goals, and intentions. In a nutshell, the rationality of the BDI model and its CPR is modulated with affective mechanisms and processes, allowing degree-of-belief updating to structured and controlled environments through the use of its extended belief predicates.

This is finalized by the use of a shallow probabilistic metacognitive norm that takes into account the selected belief degree. The term 'shallow' underlines the fact that the probabilistic metacognitive norm is hand-coded. This limitation is addressed in the final chapter as one of the possible improvement areas for the CernoCAMAL future work. The motivation for the inclusion of the shallow probabilistic metacognitive norm was a previous CAMAL spin-off work (Venkatamuni 2008) that investigated the concept of metacognition in a society of agents as a powerful catalyst for control of affective-motivational-BDI architectures, with respect to reasoning, planning, decision-making, and learning. It was concluded, based on extensive experimental results, that using norms through a metacognitive (reflective) layer would improve the performance of the CAMAL architecture in general, as well as in terms of specific metrics, e.g. life expectancy and resource collection of the agents used in those experiments.

In this thesis, however, a slightly different methodology is employed to test the same concept, in that the reactive component of CernoCAMAL is pre-programmed with a probabilistic norm prior to runtime; meaning it is defined in the domain model. Experimental results confirm the performance improvement of the CernoCAMAL architecture (see 6.2.4 & 6.3.4).

At a high level conceptual view, CernoCAMAL goes through a number of phases as it operates:

- It observes the environment using its perceptual sensors.
- It forms focused beliefs and initiates BDI-association using the affect model.
- It performs the action based on the BDI-association combinations and degrees of belief and also domain model assumptions and restraints.
- It observes the environment again, once the action has been carried out.
- It feeds the consequences on the environment into the BDI-association and modifies the association values and degrees of belief based on this feedback message.
- It reasons probabilistically about the possible number of objects and their instances based on the feedback messages received from the reactive layer.

Obviously, if the observed state of Cerno's enclosure after the action conforms to its required desire state (i.e. if architecture's instantiated behaviour has succeeded to achieve the current goal), then the association mechanism increases the association value. If, however, the environmental state after the action does not conform to the required goal state, the association value is decreased. Put simply, the association value is constantly modified and updated based on whether the corresponding action or goal-directed behaviour failed or succeeded to achieve the agent's current goal. Therefore, association values fluctuate and are based on feedback from previous actions.

During every deliberative cycle, perceptual updates lead to belief formations, and then belief revision in the probabilistic BDI model which, then, gives rise to goal revision. The association-update then uses the revised (new) belief and goal sets to determine the relevant action(s) based on the domain model rules. The motivator-update enables goal revision and the selection of the next goal, based on goal importance and goal success and also current beliefs. This, in turn, drives motivator revision using the association construct, which in turn enables belief-desire-intention combinations to be ranked based on the likelihood of their success.

Figure 4.1 shows a schematic view of the CernoCAMAL architecture, along with its major components. Also, illustrated are the motivator coupling of BDI and affect models, enabling the linking of reactive and deliberative processes through affect, and also the mapping of affordances onto objects and actors in an external environment. CernoCAMAL's motivational construct as the core representational schema of the architecture links perception, beliefs, desires, intentions, and actions.

The central idea here is the use of motivational constructs as a domain and task workspace with which a cognitive agent can organise its perceptions, beliefs, motivations, and intentions. Supporting these constructs is the affect model and various affordances that provide a consistent and systematic control language for ordering propositions, selecting goals from the desire set, constructing a plan of action from the intention set, forming a focused belief with an updated degree of belief, and prioritising processes – pulling together all these operations in a formalized manner.

In this diagrammatic view of the CernoCAMAL architecture, the left column represents perception and associated activities, the right column actions and associated activities; the central column (cognition) comprises of mechanisms responsible for reasoning, planning, behaviours, deliberation, etc. The probabilistic BDI model underpins information flow at the deliberative layer which is based upon the experimentally justified cognitive model of CRIBB. Perceptions, extended beliefs, desires, and intentions are reflected in the semantic contents of the motivational constructs present within the architecture. These map onto behaviours at the reactive level and actions via the robot's motors (actuators or effectors) in the environment. The affect model associated with the motivational constructs allows adaptive behaviour preference which is based upon the a-CRIBB model.

Belief Set: B	Goal Set: D	Intention Set: I			
Focused Belief:	$b \in B$	(Descriptor,	Source, Time,	*DegBeL*	)
Selected Goal:	$g \in D$	(Descriptor,	SuccessCondition,	*Importance*	)
Association:	a	(b, g,	$i \in I$,	*Insistence*	)
Motivator:	m	(g, a, deterministic, reactive-cycles,		*Intensity*	)

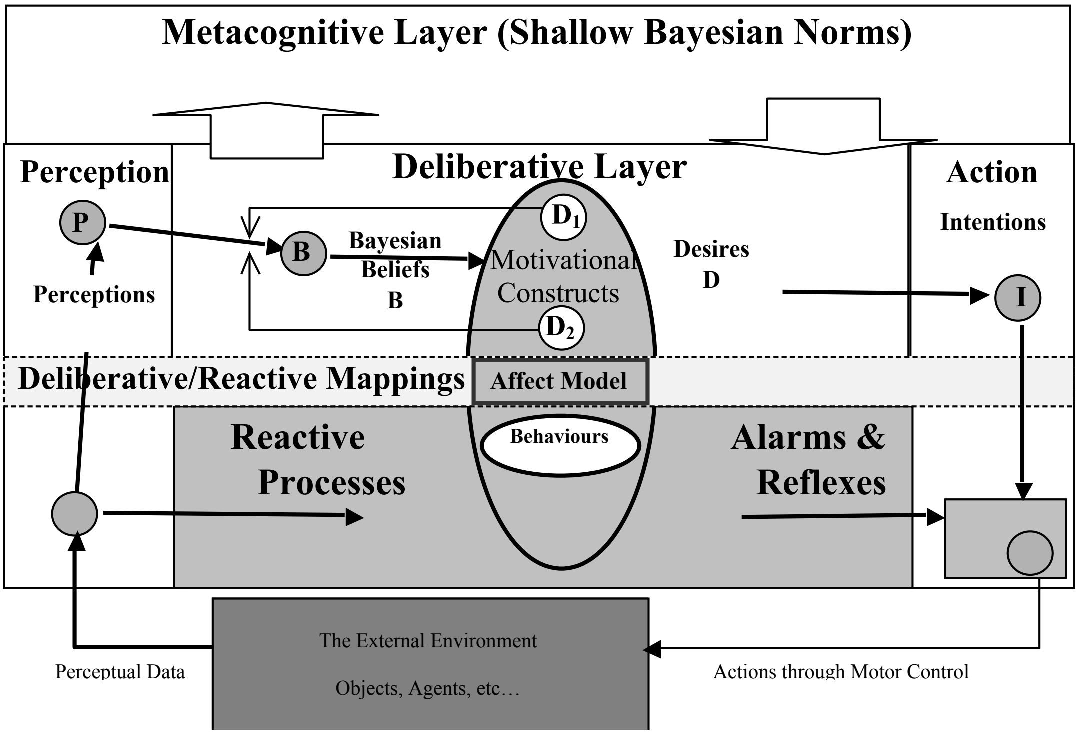

Figure 4-1: Overview of Architectural Operation

Before proceeding to experimentation for validation purposes, it is worthwhile to recap the principles and arguments behind this thesis: CAMAL is an example of a general class of integrative cognitive architectures for synthetic intelligence that is built around the concept of control states and, in particular, motivators. The primary conjecture is that the further development, design, and implementation of such cognitive architectures can proceed using a consistent and systematic control language across all aspects of reasoning and domain model management – something that the development of CernoCAMAL has taken advantage of. This control language is grounded in the use of *affect and affordances* with the aim to be consistent across different domains, tasks, and levels of processing. Together, they can be used to guide both internal and external decision-making and activities, and enable the cognitive agent to weigh its beliefs and processes, and also control the economics of its reasoning and processing.

4.7 Summary

This chapter began with a brief overview of relevant research conducted prior to CAMAL and CernoCAMAL, with regards to probabilistic thinking and reasoning. The fact that none of the CAMAL spin-off projects to date had addressed the cognitive capability of probabilistic reasoning was underlined, followed by emphasizing the main motivation of extending CAMAL to develop this fundamental cognitive ability. A brief introduction to the probability theory was given, followed by a review of the basics of the probability calculus and framework. The structure of CAMAL's belief statements was explored, from which the CernoCAMAL's Extended Belief Structure (EBS) was developed. The EBS was discussed and its advantages over CAMAL's inadequate belief statements were pointed out. The chapter then went on to introduce CernoCAMAL's Probabilistic Reasoner (CPR) and explaining how it reasoned probabilistically over the feedback generated by reactive sub-systems of CernoCAMAL. A comprehensive design criteria for the belief predicates constituting goal- and task-oriented feedback generated by reactive sub-systems was outlined. The memory facility used in the design of the CPR was addressed, followed by the norm facility incorporated in the structure of motivators. The chapter concluded with presenting an operational overview of CernoCAMAL.

5 Experimentation Testbeds

This chapter begins with a brief introduction to the concepts of testbeds and controlled experimentation. The experimental methodology used to evaluate various aspects of CernoCAMAL's cognitive performance is outlined, followed by a summary of the different types of experiments to be carried out in these two synthetic worlds. An ontological description of the two simulation testbeds that Cerno will be operating in for experimentation purposes is given. The features and implementation details for these two virtual environments are presented. The predicates and elements constituting the reactive feedback list, accompanied by their interpretations and actions to be taken by the CPR, are listed. The chapter concludes with reflecting on the way in which previous CAMAL spin-off projects' testbeds have been evaluated, and further point out the merits of the CernoCAMAL testbeds.

5.1 Introduction

The appropriate use of the methodological underpinnings of AI, such as testbeds and controlled experimentation, is imperative in evaluating a cognitive system or designing and validating a cognitive agent (Hanks, Pollack, Cohen 1993). It is also necessary to have meaningful benchmarks and metrics, and thereby allow the testing, evaluation, and comparison of the various abilities of competing cognitive systems. In the realm of Artificial Intelligence and Cognitive Robotics research, these standard problems will be those able to be carried out by animals, adults, and children of various ages, so that the level of progress of a cognitive agent can be tested against alternative cognitive agents.

A carefully-designed testbed is a challenging environment in which AI or Robotic programs can be studied. A testbed environment serves as a simplified version of a real-world environment in which the experimenter has access to particular aspects of the environment. Other aspects may be allowed to vary parametrically or even randomly. Controlled experimentation is performed on a testbed as a widely-used method of investigating cognitive systems and agents. The researcher systematically varies the features of a system or the environment in which the agent is embedded, and then measures the effects of those variations on some aspects of system performance.

All CAMAL spin-off projects including this CernoCAMAL research work have involved the use of testbeds and controlled experimentation.

5.2 Experimental Methodology

As set out in the first chapter, the CernoCAMAL research attempts to address the following specific research questions in the current cognitive architecture under investigation:

- Can CernoCAMAL reason probabilistically by exploiting the proposed EBS? Can the integration of the proposed EBS facilitate probabilistic reasoning and inference in CernoCAMAL?
- Can the BDI model run compatibly with the affect and motivational models, and affective and motivational valences used throughout the whole architecture? Can this ensure a consistent and systematic metric across all aspects of affect, reasoning, and domain model management?
- Can the probabilistic deliberation results of the CPR be used for computing changing degrees of belief given apriori, and subsequently using the BDI, affect, and motivational models to determine the agent's intentions, actions, or behaviours?
- Can the CernoCAMAL cognitive architecture be applied to virtual and physical cognitive agents using synthetic testbeds and mobile robots?

The key to designing and conducting suitable experiments for assessing the above criteria is to understand what needs to be measured and evaluated. It is then that the obtained results can be interpreted correctly as meaningful metrics and benchmarks to determine whether they validate the achievement of the research goals or perhaps otherwise.

In order to enable experimentation, the blackboard monitor has been designed to contain an `Experiment` feature. This feature facilitates the running of the cognitive architecture for a pre-defined number of times (i.e. deliberative cycles) which in turn allows statistics collection. A series of experiments, listed below, were set out to test the CernoCAMAL cognitive architecture for validating the research criteria outlined.

The obtained results from carrying out these different types of experiments in two synthetic testbeds (described next) are presented in Chapter Six.

- EBS and CPR Experiments
- Goal Achievement Success and Failure
- Population Adaptation
- Probabilistic Motivator Norms
- Comparative Experiments

5.3 Predator-Prey Tile World

This application area uses a graphical world created using SWI-Prolog Version 5.8.0 – an abstract tile world with operators that affect the world, incorporating a (white) Cerno agent, several edible spheres (blue objects), preys (red agents), and predators (green agents). A screenshot of this synthetic terrain along with its corresponding Prolog command window are shown in Figure 5.1 below.

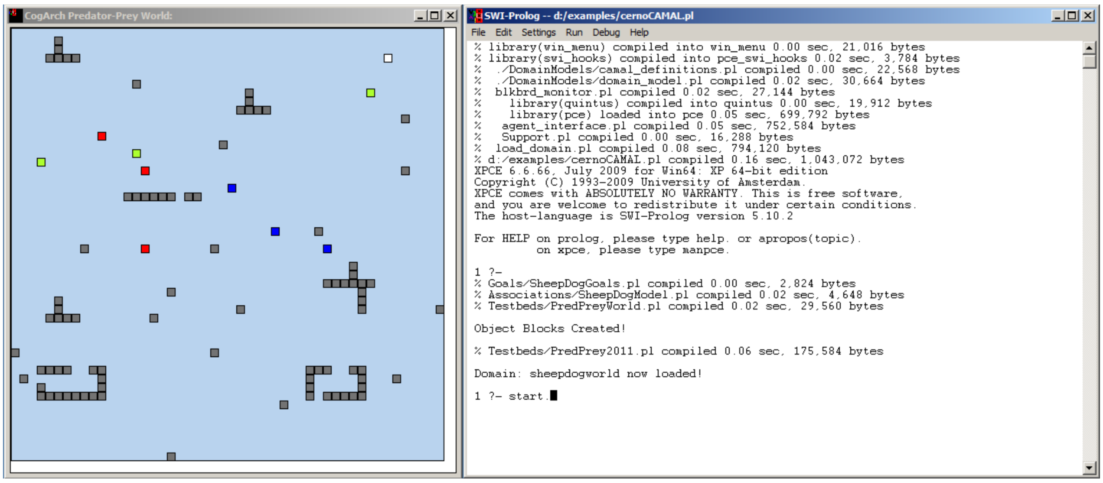

Figure 5-1: The Predator-Prey Simulation World (originally by Davis 2008)

Probability computation lends itself well to predator-prey scenario, since the computed probabilities could imply the extent or risk that a particular entity is a predator, etc.

Furthermore, probability provides a formalism to align danger risk, goal risk, task liability, etc, with the belief degrees and affect and motivational valences used throughout the architecture. Before fleshing out the specifics of the simulation domain though, it is crucial to present an *ontological analysis* of the domain model to clarify the structure of the knowledge, entities, and relationships among them.

Essentially, an *ontology* of a particular domain describes the kinds of entity involved in that domain and the relationships that can hold among different entities (Church, Hanks 1990). Ontological analysis clarifies the structure of knowledge. Given a specific domain such as predator-prey world, its ontology forms the heart of any system of knowledge representation for that domain. Without ontologies, there cannot be a *vocabulary* for representing knowledge; meaning ontologies enable knowledge sharing in a conceptual framework (Gruber 1993).

Figure 5.2 below illustrates the two possible entities that may exist in this domain:

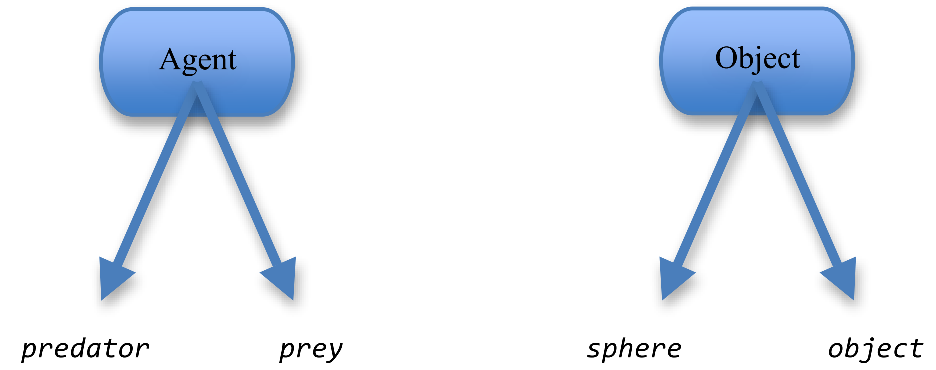

Figure 5-2: The Possible Entities of the Predator-Prey Simulation World

Here is an ontological description of the domain that Cerno will be operating in:

```
Ontology          :-   Domain Descriptor | Belief Clause |
                       Goal Clause | Intention Clause |
                       Association

Domain Descriptor :-   Object | Default Predicate Set |
                       Domain Predicate Set

Object            :-   Default Object | Domain Object
```

```
Default Object          :-    Self | Object | Identifier | Unknown

Domain Object           :-    sphere | pred | prey

Identifier              :-    ID ← gensym(Domain Object)

Default Predicate Set   :-    environment(EnvDescript)|
                              fail(Intention) |
                              time(T) | reactive_cycles(N) |
                              instance_of(Identifier, Object) |
                              degree_of_belief(assumption, DegBel) |
                              degree_of_belief(deduction, DegBel) |
                              degree_of_belief(perception, DegBel)

Domain Predicate Set    :-    Belief | Goal

Belief                  :-    hit(Object) | lost(Object) |
                              found(Object) | near(Object) |
                              know_of(Object) | instance_of(Object)
                              ate(Object) | herded(Object)
                              attacked(Object) | destroyed(Object) |
                              location(Object, Location) |
                              apriori_prob(Object, Affordance)

Goal                    :-    avoid(Collisions) | avoid(Object) |
                              find(Object) | eat(Object) |
                              herd(Object) | hit(Object) |
                              attack(Object)

Intention               :-    methodavoid | methodfind
                              methodhit | methodattack |
                              methodherd | methodeat

Association             :-    Belief | Goal | Intention | Affordance
```

```
Descriptor Set         :-    sparse | cluttered | dynamic | static

Object Set             :-    object | agent | sphere | prey | pred

Domain Super Class     :-    agent ← pred | prey
```

The testbed implementation of the predator-prey simulation world is organized around the theme that the agent is situated in a two-dimensional *500×500* grid. It is designed to support controlled experiments with cognitive agents (e.g. Cerno) situated in a dynamic and uncertain environment. Dynamic describes the fact that the simulation environment is changing all the time, as the agents are moving around. Uncertain is defined as unpredictable and describes the fact that the agents have no way of finding out which agent is going to do what next.

The world consists of the described grid on which can be placed a Cerno, several spheres, preys, predators, and obstacles (the experimenter can manipulate the numbers). The ability to modify and control these and other parameters and features of the grid world allows systematic exploration of the simulation world with various characteristics. The goal of such explorations is, essentially, to find relationships between world characteristics and corresponding characteristics of the embedded Cerno and its cognitive performance.

Each object occupies one cell of the grid. The simulation interface allows Cerno, predators, and preys to take one of four primitive actions: move left, right, up, and down, unless doing so would cause them to run into the world's boundaries or obstacles. The number of spheres, preys, and predators can be determined prior to the experiment, but obstacles appear randomly. The simulation world also has explicit sensing operators with configurable parameters, e.g. sonar and vision, with pre-defined ranges:

```
sense_limit( vision, 300 ).
sense_limit( sonar , 300 ).
sense_limit( near , 50 ).
sense_limit( herd , 45 ).
sense_limit( hit , 40 ).
sense_limit( attack, 35 ).
sense_limit( eat , 30 ).
```

There are five scenarios as to what Cerno could represent as a cognitive agent, in order to investigate probabilistic reasoning and behaviour selection:

1. **Scenario One**: Cerno avoids everything.
2. **Scenario Two**: Cerno finds and eats sphere.
3. **Scenario Three**: Cerno finds and herds prey.
4. **Scenario Four**: Cerno finds and attacks pred.
5. **Scenario Five**: This is a more sophisticated scenario, building on the first four. Here, Cerno acts as a virtual sheepdog, herding prey, feeding off spheres, and protecting against predators.

For each scenario, goals and associations are designed according to roles, such as being a herd dog, and what that may entail. Moreover, in each scenario 'prey' refers to a sheep that is both regarded as prey for a predator and to be herded by Cerno. Due to multiple various goals and goal preferences of scenario five, and also greater number of goals and associations, this scenario is selected for carrying out experiments. The first four scenarios are not of significant interest for experimentation, as they do not represent complex settings. It is in the fifth scenario that goals start to fail, etc.

5.4 ARIA MobileSim World

ARIA (**A**ctiveMedia **R**obotics **I**nterface for **A**pplications) is an object-oriented API (**A**pplications **P**rogramming **I**nterface) written in C++. It is a client-side software for access to and management of ActiveMedia mobile robots and robotics applications. Its flexibility makes ARIA an excellent foundation for high-level control of robots that effectively act as the server in a client-server environment. ARIA can be used in many different ways, from simple command-control of the robot for direct-drive navigation, to development of high-level intelligent actions.

MobileSim is a simulator developed by ActiveMedia Robotics Corporation (providers of the ARIA API) to simulate ActiveMedia mobile robots and their environments, which is useful for debugging and testing ARIA programs. It converts an ActiveMedia map file into an stage environment and then places simulated robots in that environment – in this case a P3DX. The MobileSim world consists of an stage environment, simulated robots, and some obstacles.

To operate the robots, the ARIA library is used. This is a C++ library that can be used with Microsoft Visual Studio on Windows operating system. Essentially, ARIA communicates with the robots over a *COM1* port (*RS-232* serial port) or it can communicate with the MobileSim simulator over a local TCP socket. Figure 5.3 shows a sample environment containing *3* (red) robots and *4* (green) objects.

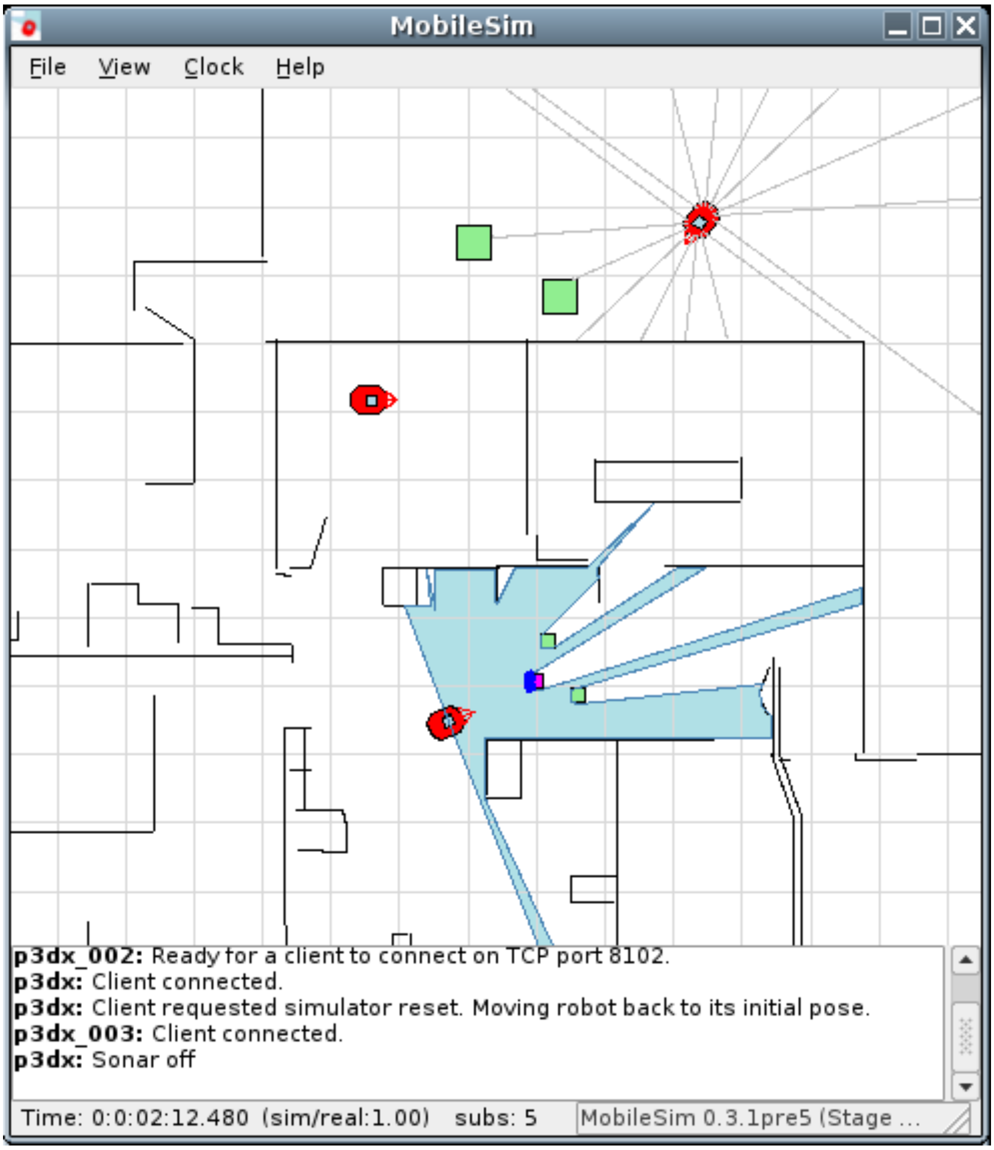

Figure 5-3: The ARIA MobileSim Simulation World (ActiveMedia 2010)

The CernoCAMAL framework was initially implemented as a situated cognitive agent in a synthetic predator-prey Tile World. It is, subsequently, interfaced with ARIA's MobileSim environment and applied to a virtual P3DX robot to implement a cognitive robotic agent. Figure 5.4 illustrates the two possible entities that may exist in this simulation domain:

Figure 5-4: The Possible Entities of the ARIA MobileSim World

Here is an ontological description of the domain that Cerno will be operating in:

```
Ontology            :-    Domain Descriptor | Belief Clause |
                          Goal Clause | Intention Clause |
                          Association

Domain Descriptor :-      Object | Default Predicate Set |
                          Domain Predicate Set

Object              :-    Default Object | Domain Object

Default Object      :-    Self | Object | Identifier | Unknown

Domain Object       :-    object | agent

Identifier          :-    ID ← gensym(Domain Object)

Default Predicate Set   :-    environment(EnvDescript)|
                              fail(Intention) |
                              time(T) | reactive_cycles(N) |
                              instance_of(Identifier, Object) |
                              degree_of_belief(assumption, DegBel) |
                              degree_of_belief(deduction, DegBel) |
                              degree_of_belief(perception, DegBel)
```

```
Domain Predicate Set    :-    Belief | Goal

Belief                  :-    hit(Object) | lost(Object) |
                              found(Object) | near(Object) |
                              know_of(Object) | instance_of(Object)
                              location(Object, Location) |
                              apriori_prob(Object, Affordance)

Goal                    :-    avoid(Collisions) |
                              avoid(Object) | find(Object)

Intention               :-    methodavoid | methodfind

Association             :-    Belief | Goal | Intention | Affordance

Descriptor Set          :-    sparse | cluttered | dynamic | static

Object Set              :-    object | agent

Domain Super Class      :-    agent ← robot
```

The MobileSim interface allows the CernoCAMAL architecture to be mounted on a P3DX robot that moves around (unless doing so would cause it to run into the world's boundaries or obstacles). Another robot – an Amigo – represents a predator in this terrain. There are also some obstacles as spheres and preys that appear randomly. The simulation world has explicit sensing operators with configurable parameters, e.g. sonar with pre-defined ranges. Figure 5.5 shows the angular positions that ARIA assigns to each of the sonar sensors on the robots (of course virtual ones here).

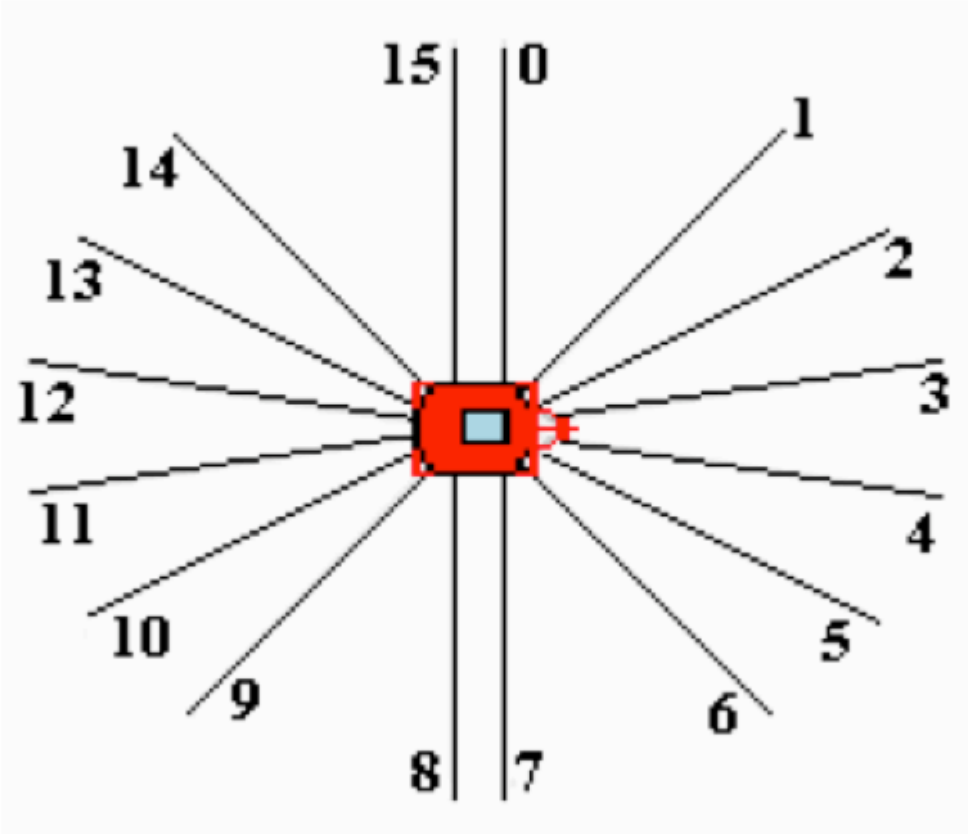

Figure 5-5: The Angular Positions of the Sonar Sensors (Whitbrook 2010)

It must be noted that presently sonar readings can only give a range to whatever one of the sonar sensors detects, from which the position coordinates can be calculated as ARIA does for range sensors. It is, however, not possible to detect the identity or colour of an object. This is one of the shortcomings of the MobileSim environment that consequently simplifies the experiments performed in this simulation environment.

5.5 CPR Functionality in Testbeds

The following are a comprehensive set of possible elements constituting the reactive feedback list, accompanied by their interpretations and actions to be taken by CernoCAMAL's CPR. These predicates are applicable to both simulation worlds, but mostly specific to the predator-prey terrain, e.g. `ate(prey)`. Understanding these mechanisms is a prerequisite to making sense of the experiments.

LOST

1. Lost an object in its general form, e.g. *lost(sphere)* → all instances of that object are lost + all *know_of* and *instance_of* are retracted + memory of all the lost instances is formed + a suitable message is displayed.
2. Lost an instance of an object that has never been found before, i.e. illegal argument for the predicate *lost* → a suitable message is displayed.
3. Lost an instance of an object that has actually been found before → that particular instance is lost + *know_of* and *instance_of* for that particular instance are retracted + memory of the lost instance is formed + a suitable message is displayed.

ATE

1. Ate an object in its general form, e.g. *ate(sphere)* → only one instance of that object is eaten (the most-recently found instance, i.e. *Recency Conflict Resolution*) + that particular instance is destroyed + *know_of* and *instance_of* for that particular instance are retracted + a suitable message is displayed.
2. Ate an instance of an object that has never been found before → i.e. illegal argument for the predicate *ate* → a suitable message is displayed.
3. Ate an instance of an object that has actually been found before → that particular instance is destroyed + *know_of* and *instance_of* for that particular instance are retracted + a suitable message is displayed.

FOUND

1. Found an instance that has already been found before → a suitable message is displayed.
2. Found an object (either known such as *sphere* or unknown such as *object*) in its general form, that there is a memory trail of a previously lost instance of it → that previously generated lost instance is re-created, instead of a new incremental one + a suitable message is displayed + *know_of* and *instance_of* for that particular instance are asserted.

3. Found an object (either known such as *sphere* or unknown such as *object*) in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be found → a new unique instance is generated + a suitable message is displayed + *know_of* and *instance_of* for the new unique instance are asserted.
4. Found an object (either known such as *sphere* or unknown such as *object*) in its general form for the *second* time (i.e. two *found*s in a single reactive feedback list) → the first *found* refers to a previously found instance + the second *found* generates a new unique instance + a suitable message is displayed + *know_of* and *instance_of* for the new unique instance are asserted.
5. Found an object (either known such as *sphere* or unknown such as *object*) in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list) → *found* refers to that previously found instance + a suitable message is displayed.
6. Found an illegal instance (an instance that cannot possibly be recognized as that instance!) → a suitable message is displayed.

NEAR

1. Near an instance that has already been found before → a suitable message is displayed.
2. Near an object (either known such as *sphere* or unknown such as *object*) in its general form, that there is a memory trail of a previously lost instance of it → that previously generated lost instance is re-created, instead of a new incremental one + a suitable message is displayed + *know_of* and *instance_of* for that particular instance are asserted.
3. Near an object (either known such as *sphere* or unknown such as *object*) in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be near to (to be found) → a new unique instance is generated + a suitable message is displayed + *know_of* and *instance_of* for the new unique instance are asserted.

4. Near an object (either known such as `sphere` or unknown such as `object`) in its general form for the *second* time (two `near`s in a single reactive feedback list) → the first `near` refers to a previously found instance + the second `near` generates a new unique instance + a suitable message is displayed + `know_of` and `instance_of` for the new unique instance are asserted.
5. Near an object (either known such as `sphere` or unknown such as `object`) in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list) → `near` refers to that previously near (found) instance + a suitable message is displayed.
6. Near an illegal instance (an instance that cannot possibly be recognized as that instance) → a suitable message is displayed.

`HIT`

1. Hit an instance that has already been found before → a suitable message is displayed.
2. Hit an object (either known such as `sphere` or unknown such as `object`) in its general form, that there is a memory trail of a previously lost instance of it → that previously generated lost instance is re-created, instead of a new incremental one + a suitable message is displayed + `know_of` and `instance_of` for that particular instance are asserted.
3. Hit an object (either known such as `sphere` or unknown such as `object`) in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be hit (to be found) → a new unique instance is generated + a suitable message is displayed + `know_of` and `instance_of` for the new unique instance are asserted
4. Hit an object (either known such as `sphere` or unknown such as `object`) in its general form for the *second* time (two `hit`s in a single reactive feedback list) → the first `hit` refers to a previously found instance + the second `hit` generates a new unique instance + a suitable message is displayed + `know_of` and `instance_of` for the new unique instance are asserted

5. Hit an object (either known such as `sphere` or unknown such as `object`) in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list) → `hit` refers to that previously hit (found) instance + a suitable message is displayed
6. Hit an illegal instance (an instance that cannot possibly be recognized as that instance) → a suitable message is displayed

`Attacked`

1. Attacked an instance that has already been found before → a suitable message is displayed.
2. Attacked an object (either known such as `sphere` or unknown such as `object`) in its general form, that there is a memory trail of a previously lost instance of it → that previously generated lost instance is re-created, instead of a new incremental one + a suitable message is displayed + `know_of` and `instance_of` for that particular instance are asserted.
3. Attacked an object (either known such as `sphere` or unknown such as `object`) in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be hit (to be found) → a new unique instance is generated + a suitable message is displayed + `know_of` and `instance_of` for the new unique instance are asserted
4. Attacked an object (either known such as `sphere` or unknown such as `object`) in its general form for the *second* time (two `attacked`s in a single reactive feedback list) → the first `hit` refers to a previously found instance + the second `attacked` generates a new unique instance + a suitable message is displayed + `know_of` and `instance_of` for the new unique instance are asserted
5. Attacked an object (either known such as `sphere` or unknown such as `object`) in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list) → `attacked` refers to that previously hit (found) instance + a suitable message is displayed
6. Attacked an illegal instance (an instance that cannot possibly be recognized as that instance) → a suitable message is displayed

5.6 Testbeds Evaluation

Hanks et al. (1993) in their thorough examination of testbeds [12] and controlled experimentation, identify a major purpose for the use of these methods: to provide meaningful metrics and benchmarks for comparing competing systems. They also argue that the scientific value of well-crafted testbeds and experimental methods is their power to highlight interesting aspects of the system performance. They further state that this value is realized only if the researcher can adequately explain why their system behaves the way it does.

As remarked earlier, all CAMAL spin-off projects including this CernoCAMAL research work have involved the use of testbeds and controlled experimentation. The foundation for CAMAL as a UTC is based on the on-going development of computational architectures for cognition, affect, and motivation that make use of a BDI model, an affect model, and a motivational blackboard. It would be instructive to reflect on the way in which previous CAMAL spin-off projects' testbeds have been evaluated, and further point out the advantages of the CernoCAMAL testbeds.

It is still open to argument, though, what the appropriate evaluation criteria for measuring any individual cognitive architecture testbed are, thus the success of any cognitive research project can be difficult to assess. However, the list below provides a guideline for approaches that measure the mentioned past projects' testbeds and experiments on CAMAL. It notes some essential features that testbeds should exhibit. It is then followed by a table, highlighting these measures and criteria against which the CAMAL past projects, including CernoCAMAL, can be evaluated.

The need for experimentation in testbeds is manifest: we want to understand why and how well our cognitive architecture (CernoCAMAL) works and how efficiently our cognitive agent (Cerno) operates in dynamic and unknown environments. Obviously, we will not always be able to do so using analytical methods! Experimentation is therefore key.

[12] Some noteworthy examples of Simulation Grid Worlds: The Parameterisable Tile World of Pollack and Ringuette (Pollack & Ringuette 1990), The Multi-Agent MICE Grid World (Montgomery & Durfee 1990), The Phoenix Fire fighting Testbed (Hart, Cohen, Greenberg, Westbrook 1990), The Truck World Simulator (Hanks et al. 1992), The NASA Tile World (Philips & Bresina 1991), The DND Tile World (Davis 2008).

- The testbed should have a clean interface, in order to maintain a clear distinction between the agents and objects and also the world in which they are all embedded.

- The testbed, although reasonably realistic [13] should be controlled and simplified.

- The testbed should be populated with agents having imperfect sensors and motors (actuators – effectors).

- The testbed should present a well-defined and reasonable model of passing time.

- The testbed and the carried out controlled experimentation should allow for problems and environmental conditions to be varied in a controlled fashion.

- The testbed should allow performance statistics to be gathered and then used for the evaluation of the system performance. Ideally, it should also be useful for the data to be formatted automatically for analysis.

- Measure of success should not solely be based on a simple achievement of a goal state. It should also take into account the possibility of partial goal satisfaction (Haddaway, Hanks 1993; Wellman, Doyle 1991).

- The testbed metrics should be both internal (separate module function) and external (behavioural – observed responses).

[13] Meaning it contains objects, blocks, etc, as in a real world terrain.

Evaluation Criteria	*Nunes 2001*	*Bourgne 2003*	*Lewis 2004*	*Venkatamuni 2008*	*Gwatkin 2009*	*Miri 2012*
Clean and Clear Interface	✓	✓	✓	✓	N/A	✔
Controlled and Simplified Environment	✓	✓	✓	✓	✓	✓
Realistically Imperfect Sensors and Motors	N/A	N/A	N/A	N/A	✓	✓
Feature Variation Flexibility (Parameterisability)	✓	✓	✓	✓	✓	✓
Partial Goal Satisfaction	×	×	×	×	×	✔
Statistics Collection	×	✓	✓	✓	✓	✓
Validating Test Cases	✓	✓	✓	✓	✓	✓

Table 5-1: Evaluation of CAMAL's Spin-Off Projects

A crucial observation to be made here is that experimentation mandates simplification, and that must not be mistaken with insufficiency or incomprehensive evaluation. The success of any cognitive architecture research project can be evaluated and reported by concentrating on only that specific area in which the experimentation is being carried out – in this thesis being the probabilistic reasoning capability, object and instant recognition, and degree of belief manipulation of CernoCAMAL. It may be necessary for a while to tolerate relatively simplified synthetic testbeds as we explore the cognitive abilities of particular cognitive architectures. Later, we can gradually add depth and realism to the simulation systems we build.

5.7 Summary

This chapter began with a brief introduction to the concepts of testbeds and controlled experimentation. The experimental methodology used to evaluate various aspects of CernoCAMAL's cognitive performance was outlined, followed by a summary of the different types of experiments to be carried out in these two synthetic worlds. An ontological description of the two simulation testbeds that Cerno will be operating in for experimentation purposes was given. The features and implementation details for these two virtual environments were presented. The predicates and elements constituting the reactive feedback list, accompanied by their interpretations and actions to be taken by the CPR, were listed. The chapter concluded with reflecting on the way in which previous CAMAL spin-off projects' testbeds had been evaluated, and further pointed out the merits of the CernoCAMAL testbeds.

6 Experiments and Results

This chapter begins with a brief introduction to a series of experiments with the CernoCAMAL cognitive architecture in the two synthetic testbeds described. The details of the experiments are outlined, starting with the most important one – how the proposed EBS enables probabilistic object / instance reasoning in the CPR. After discussing the experiments one by one, the obtained graphs are presented. This is followed by repeating the same experimentation process in the second testbed. A final experiment is proposed to use the most recent phase of CAMAL research (RoboCAMAL: Gwatkin 2009) that focused on translating the concepts of motivation, affect, and learning from a simulated agent to an embedded robot. The rationale behind this evaluation method is highlighted, followed by the details of the tests carried out. The chapter concludes with summarising the acquired results.

6.1 Introduction

This thesis presented one possible way to develop a probabilistic CAMAL that could be used to govern artificial minds probabilistically. The motivation and impetus for this investigation and subsequently extending CAMAL was the considerable evidence that probabilistic thinking and reasoning was linked to cognitive development and played a role in cognitive functions. This led us to believe that a probabilistic reasoning capability was an essential part of human intelligence. Thus, it should be a vital part of any system that attempted to emulate human intelligence computationally. A series of experiments were set out to test the CernoCAMAL cognitive architecture for validating its research goals and objectives, including:

- EBS and CPR Experiments (in CernoCAMAL)
- Goal Achievement Success and Failure (between CAMAL and CernoCAMAL)
- Population Adaptation (in CernoCAMAL)
- Probabilistic Motivator Norms (in CernoCAMAL)
- Comparative Experiments (between CernoCAMAL and RoboCAMAL)

6.2 Predator-Prey Experiments

This section presents the predator-prey experiments carried out using the CernoCAMAL cognitive agent in the predator-prey tile-world testbed. A number of experiments were performed over a number of cycles, and some internal variables and statistics were recorded. From the obtained results, the two cognitive architectures of CAMAL and CernoCAMAL can be compared and contrasted.

6.2.1 Validation of CPR in Predator-Prey Testbed

These tests are performed to ensure that the CPR can use the EBS and assumed degrees of belief, infer posterior probabilities correctly, assign them to the appropriate belief descriptors, and reason probabilistically about the number of objects and their instances that may be present in the environment. Cerno was allowed to operate in the sheep-dog virtual world (see 5.2) that consisted of a varying number of spheres, preys, and predators. Each experiment was run for *10* deliberative cycles. This was repeated with and without *obstacles*. The configuration of having *3* spheres, *3* preys, and *3* predators, with no unidentified objects, can be represented as:

```
belief ( apriori_prob ( object, 0.0 ), assumption, 1, 0.5 ).
belief ( apriori_prob ( sphere, 1/3 ), assumption, 1, 0.5 ).
belief ( apriori_prob ( pred,   1/3 ), assumption, 1, 0.5 ).
belief ( apriori_prob ( prey,   1/3 ), assumption, 1, 0.5 ).
```

The reactive component was pre-configured with the following goals:

```
find ( sphere )        eat ( sphere )
find ( prey )          herd ( prey )
find ( pred )          attack ( pred )
```

This initial configuration assumes that if there is one object, then for Cerno it is equally likely to be a sphere, a prey, or a predator, hence the probability of any domain object being present is *1/3* or *0.3* with a *0.5* degree of belief since the statements are assumptive.

CernoCAMAL, upon receiving goal- and task-oriented reactive feedback interprets them and reasons about the number of objects and their instances that may be present in the environment. It is imperative to understand what exactly every predicate indicates and how it is operated on. In particular, note the instances that are destroyed (eaten) and cannot be re-generated when stumbled upon, and the instances that are lost but can be re-generated when stumbled upon. Table 6.1 over the next few pages shows how Cerno interprets the constituent elements of a reactive feedback list, along with its reasoning and the actions it takes.

Environment Description/Occurrence	**Cerno's Interpretations/Actions**
Lost an instance of an object that has actually been found before, e.g. `lost(sphere1).`	That particular instance is lost.
Lost an object in its general form, e.g. `lost(sphere).`	All instances of that object are lost.
Ate an instance of an object that has actually been found before, e.g. `ate(prey2).`	That particular instance is eaten and, therefore, destroyed.
Ate an object in its general form, e.g. `ate(sphere).`	Only one instance of that object is eaten (the most-recently found instance, i.e. Recency Conflict Resolution) + that particular instance is destroyed.
Found an object in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be found.	A new unique instance is generated.
Found an object in its general form, that there is memory trail of a previously lost instance of it.	That previously generated lost instance is re-created, instead of a new incremental one.

Found an object in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list).	`found` refers to that previously found instance.
Found an object in its general form for the *second* time (i.e. two `found`s in a single reactive feedback list).	the first `found` refers to a previously found instance + the second `found` generates a new unique instance.
Near an object in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be near to.	A new unique instance is generated.
Near an object in its general form, that there is memory trail of a previously lost instance of it.	That previously generated lost instance is re-created, instead of a new incremental one.
Near an object in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list).	`near` refers to that previously near (found) instance.
Near an object in its general form for the *second* time (two `near`s in a single reactive feedback list).	The first `near` refers to a previously found instance + the second `near` generates a new unique instance.
Hit an object in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be hit.	A new unique instance is generated.
Hit an object in its general form, that there is memory trail of a previously lost instance of it.	That previously generated lost instance is re-created, instead of a new incremental one.
Hit an object in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list).	`hit` refers to that previously hit (found) instance.

Hit an object in its general form for the *second* time (two `hit`s in a single reactive feedback list).	The first `hit` refers to a previously found instance + the second `hit` generates a new unique instance.
Attacked an object in its general form, while the instance list of that object is empty, i.e. that object is definitely the first to be attacked.	A new unique instance is generated.
Attacked an object in its general form, that there is memory trail of a previously lost instance of it.	That previously generated lost instance is re-created, instead of a new incremental one.
Attacked an object in its general form, while it can refer to a previously found instance, rather than generating a new unique instance (rather than extending the instance list).	`attacked` refers to that previously hit (found) instance
Attacked an object in its general form for the *second* time (two `attacked`s in a single reactive feedback list)	the first `attacked` refers to a previously found instance + the second `attacked` generates a new unique instance

Table 6-1: Cerno's Reasoning over Reactive Feedback Lists

The initial assignment of degrees of belief follows the Belief Preference Model values (below) and basics of probability calculus detailed in section 4.3 including the main rule of probability that states: *The probability of an event is defined as the ratio of the number of outcomes of that event divided by the total number of possible outcomes.* For example [pred1, prey1, sphere1] indicates the following set of probabilities (P(pred)=1/3, P(prey)=1/3, P(sphere)=1/3) and [pred1, prey1, sphere1, sphere2] indicates the following set of probabilities (P(pred)=1/4, P(prey)=1/4, P(sphere)=2/4) as evident in the sample run of the program over the next few pages.

Beliefs Preference Model:

```
belief_preference( perception, assumption ).

belief_preference( perception, deduction ).

belief_preference( deduction, assumption ).
```

A sample run of the program, demonstrating that Cerno operates as expected, is presented below:

```
[ found(pred), found(prey), found(sphere) ].

found(pred)       -->   pred1

found(prey)       -->   prey1

found(sphere)     -->   sphere1

==> [ pred1, prey1, sphere1 ]

belief( apriori_prob(object, 0.0),  deduction, 1, 0.75 ).

belief( apriori_prob(sphere, 0.3),  deduction, 1, 0.75 ).

belief( apriori_prob(pred,   0.3),  deduction, 1, 0.75 ).

belief( apriori_prob(prey,   0.3),  deduction, 1, 0.75 ).

------------------------------------------------------------------

[ found(sphere), hit(prey), found(sphere) ].

found(sphere)     -->   refers to sphere1

hit(prey)         -->   refers to prey1

found(sphere)     -->   sphere2

==> [ pred1, prey1, sphere1, sphere2 ]

belief( apriori_prob(object, 0.0),  deduction, 2, 0.75 ).

belief( apriori_prob(sphere, 0.5),  deduction, 2, 0.75 ).

belief( apriori_prob(pred, 0.25),   deduction, 2, 0.75 ).

belief( apriori_prob(prey, 0.25),   deduction, 2, 0.75 ).

------------------------------------------------------------------

[ found(sphere), found(prey), attacked(pred) ].

found(sphere)     -->   refers to either sphere1 or sphere2

found(prey)       -->   refers to prey1

attacked(pred)    -->   refers to pred1

==> [ pred1, prey1, sphere1, sphere2 ]
```

```
belief( apriori_prob(object, 0.0),  deduction, 3, 0.75 ).

belief( apriori_prob(sphere, 0.5),  deduction, 3, 0.75 ).

belief( apriori_prob(pred, 0.25),   deduction, 3, 0.75 ).

belief( apriori_prob(prey, 0.25),   deduction, 3, 0.75 ).

-------------------------------------------------------------------

[ found(prey), found(sphere), near(pred) ].

found(prey)       -->   refers to prey1

found(sphere)     -->   refers to either sphere1 or sphere2

near(pred)        -->   refers to pred1

==> [ pred1, prey1, sphere1, sphere2 ]

belief( apriori_prob(object, 0.0),  deduction, 4, 0.75 ).

belief( apriori_prob(sphere, 0.5),  deduction, 4, 0.75 ).

belief( apriori_prob(pred, 0.25),   deduction, 4, 0.75 ).

belief( apriori_prob(prey, 0.25),   deduction, 4, 0.75 ).

-------------------------------------------------------------------

[ found(prey), lost(pred1), attacked(pred) ].

found(prey)       -->   refers to prey1

lost(pred1)       -->   refers to pred1

attacked (pred)   -->   refers to previously-lost pred1

==> [ pred1, prey1, sphere1, sphere2 ]

belief( apriori_prob(object, 0.0),  deduction, 5, 0.75 ).

belief( apriori_prob(sphere, 0.5),  deduction, 5, 0.75 ).

belief( apriori_prob(pred, 0.25),   deduction, 5, 0.75 ).

belief( apriori_prob(prey, 0.25),   deduction, 5, 0.75 ).

-------------------------------------------------------------------
```

```
[ ate(prey), found(pred), found(prey), near(sphere) ].

ate(prey)         -->   prey1 is destroyed

found(pred)       -->   refers to pred1

found(prey)       -->   prey2  (prey1 cannot come back to life)

near(sphere)      -->   refers to either sphere1 or sphere2

==> [ pred1, prey2, sphere1, sphere2 ]

belief( apriori_prob(object, 0.0),  deduction, 6, 0.75 ).

belief( apriori_prob(sphere, 0.5),  deduction, 6, 0.75 ).

belief( apriori_prob(pred, 0.25),   deduction, 6, 0.75 ).

belief( apriori_prob(prey, 0.25),   deduction, 6, 0.75 ).

-------------------------------------------------------------------

[ ate(sphere), found(pred), hit(prey), found(sphere) ].

ate(sphere)       -->   sphere2 is destroyed  (recency rule)

found(pred)       -->   refers to pred1

hit(prey)         -->   refers to prey2

found(sphere)     -->   refers to sphere1

==> [ pred1, prey2, sphere1 ]

belief( apriori_prob(object, 0.0),  deduction, 7, 0.75 ).

belief( apriori_prob(sphere, 0.3),  deduction, 7, 0.75 ).

belief( apriori_prob(pred, 0.3),    deduction, 7, 0.75 ).

belief( apriori_prob(prey, 0.3),    deduction, 7, 0.75 ).

-------------------------------------------------------------------

[ attacked(pred), found(pred), found(prey), found(sphere) ].

attacked(pred)    -->   refers to pred1

found(pred)       -->   pred2

found(prey)       -->   refers to prey2

found(sphere)     -->   refers to sphere1
```

```
==> [ pred1, pred2, prey2, sphere1 ]
belief( apriori_prob(object, 0.0),  deduction, 8, 0.75 ).
belief( apriori_prob(sphere, 0.25), deduction, 8, 0.75 ).
belief( apriori_prob(pred, 0.5),    deduction, 8, 0.75 ).
belief( apriori_prob(prey, 0.25),   deduction, 8, 0.75 ).
-------------------------------------------------------------------
[ lost(pred), found(prey), near(sphere), attacked(pred) ].
lost(pred)        -->   refers to both pred1 and pred2
found(prey)       -->   refers to prey2
near(sphere)      -->   refers to sphere1
attacked (pred)   -->   refers to previously-lost pred1
==> [ pred1, prey2, sphere1 ]
belief( apriori_prob(object, 0.0),  deduction, 9, 0.75 ).
belief( apriori_prob(sphere, 0.3),  deduction, 9, 0.75 ).
belief( apriori_prob(pred, 0.3),    deduction, 9, 0.75 ).
belief( apriori_prob(prey, 0.3),    deduction, 9, 0.75 ).
-------------------------------------------------------------------
[ found(pred), found(prey), hit(sphere), near(sphere) ].
found(pred)       -->   refers to previously-lost pred2
found(prey)       -->   refers to prey2
hit(sphere)       -->   refers to sphere1
near(sphere)      -->   sphere3  (sphere2 cannot come back to life)
==> [ pred1, pred2, prey2, sphere1, sphere3 ]
belief( apriori_prob(object, 0.0),  deduction, 10, 0.75 ).
belief( apriori_prob(sphere, 0.4),  deduction, 10, 0.75 ).
belief( apriori_prob(pred, 0.4),    deduction, 10, 0.75 ).
belief( apriori_prob(prey, 0.2),    deduction, 10, 0.75 ).
```

6.2.2 Goal Achievement Success and Failure

A succession of experiments was carried out to evaluate CernoCAMAL's overall performance, in terms of goal success and failure, and consequently task effectiveness. The objective here was to assess the efficacy of the CernoCAMAL architecture over the original CAMAL in a tangible manner, by using success and failure counts. In other words, efficacy is measured in quantitative terms here, as: greater number of successes and lower number of failures. A general sample set of obtained experimental results are presented in figures 6.1 and 6.2 to enable the comparison of the two cognitive architectures. In the predator-prey tile world, the term *success* refers to one of the following:

1. The set goal (defined in the domain model) is achieved. The new beliefs explicitly state that. This situation is known as `Explicit Goal Success`.
2. The set goal (defined in the domain model) is achieved. The new beliefs state that the negation of the goal negation has been achieved. This situation is known as `Double Negation Success`.
3. The intention (defined in the association construct) is accomplished. The new beliefs state that. This situation is known as `Explicit Intention Success`.
4. The avoid-collisions intention (defined in the association construct) is accomplished. The new beliefs state that. This situation is known as `Avoid-Collisions Success`.

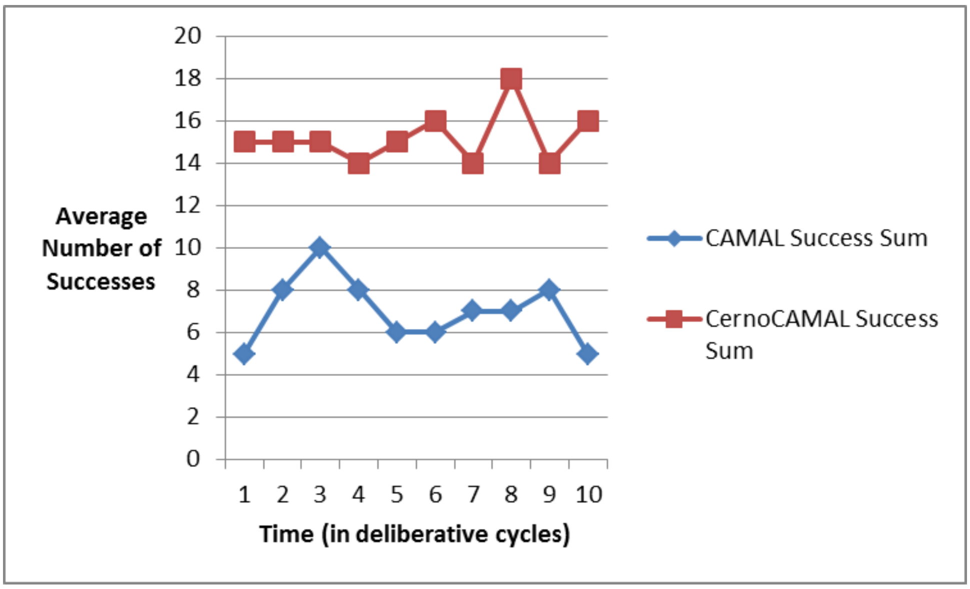

Figure 6-1: Overall Increase in the Number of Successes

These results serve to demonstrate a clear increase in the number of successes in time that occurred, directly supporting the overall advantage of the CernoCAMAL architecture over CAMAL in terms of success counts. Similarly, the term *failure* refers to one of the following:

1. The set goal (defined in the domain model) is not achieved. The new beliefs explicitly state that. This situation is known as `Explicit Goal Failure`.
2. The set goal (defined in the domain model) is not achieved. The new beliefs state that the negation of the goal has been achieved. This situation is known as `Goal Negation Failure`.
3. The set goal (defined in the domain model) is achieved, but on the wrong object. The new beliefs state that. This situation is known as `Wrong-Object Goal Failure`.
4. The intention (defined in the association construct) is not accomplished. The new beliefs state that. This situation is known as `Explicit Intention Failure`.
5. The avoid-collisions intention (defined in the association construct) is not accomplished. The new beliefs state that. This situation is known as `Avoid-Collisions Failure`.

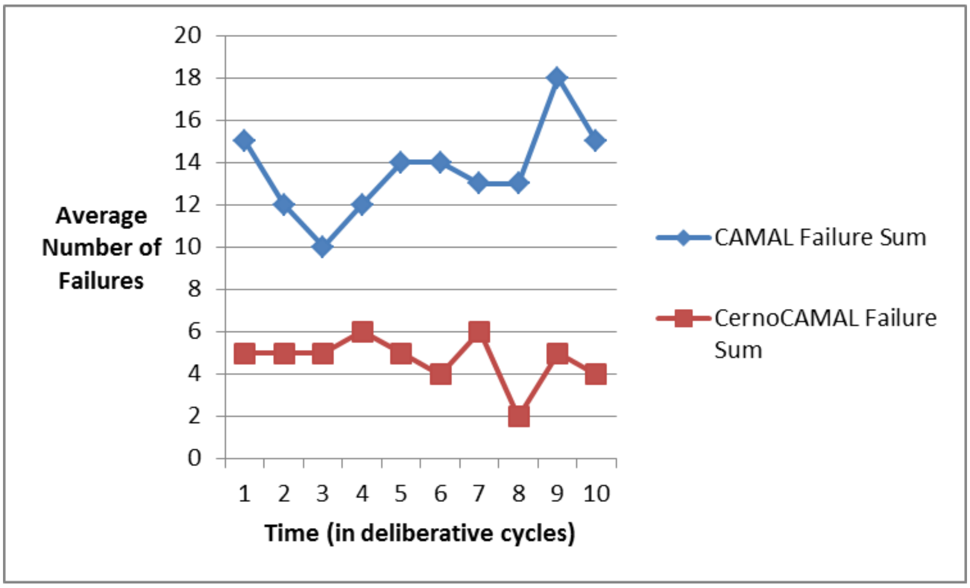

Figure 6-2: Overall Reduction in the Number of Failures

These results serve to demonstrate a clear decrease in the number of failures in time that occurred, directly supporting the overall advantage of the CernoCAMAL architecture over CAMAL in terms of failure counts.

Summarily, the term *task effectiveness* describes if overall the tasks (goals and intentions collectively) were successfully completed. This is inferred based on the increase in the number of success counts and the reduction in the number of failure counts. The two graphs clearly show a decrease in the number of failures in time along with an increase in the number of successes in time that occurred. This outcome indicates the overall advantage of the CernoCAMAL architecture over CAMAL in terms of success / failure counts, and also task effectiveness.

6.2.3 Population Adaptation Experiments

These experiments investigate CernoCAMAL's ability to adapt in a dynamic, uncertain environment. Adaptation is a term commonly used in AI to connote assimilation and adjustment of a cognitive agent to its surroundings, particularly if there is uncertainty inherent to the environment the agent is embedded in, e.g. changes taking place in an unpredictable fashion. Cerno is a cognitive agent equipped with a means of reasoning probabilistically about objects and their instances in its enclosure. It is, therefore, reasonable to conjecture that over time the findings using Cerno about a probable number of objects and their instances should be close to the actual numbers pre-determined by the experimenter, hence the term Population Adaptation.

Furthermore, if the controlled experimentation was paused and some modification to the number of specific objects and their instances were made, upon resuming experimentation, the findings using Cerno about a probable number of objects and their instances should again be close to the actual numbers pre-set by the experimenter. This particular interpretation of adaptability is what we refer to in this thesis. For this purpose, the `Experiment` feature of the blackboard monitor was deployed. The number of domain model objects was incremented until there was *5* of each, giving a total of *15* objects in the simulation world (anything more would have been too cluttered). Upon completion of the tests, the inferred numbers for each object and their instances were recorded.

Figure 6.3 illustrates the obtained graph based on the experimental results. It can clearly be seen how close the findings using Cerno were to the actual number of objects and their instances in its environment, with no drastic error in working out the population number. These results serve to demonstrate CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about the number of objects and their instances.

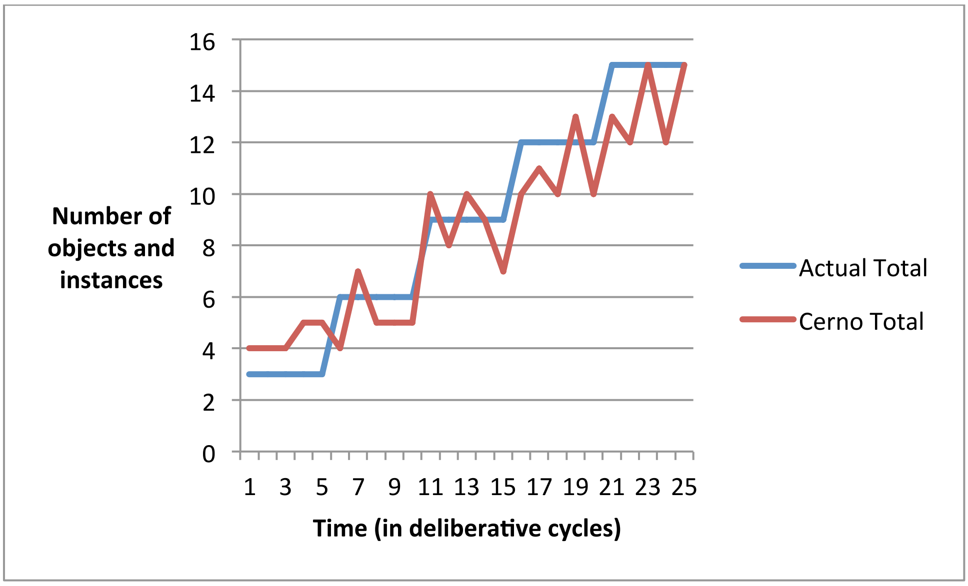

Figure 6-3: First Adaptability Experiment

For the second part, to perform the add/remove assessments based on the highlighted specific perspective on adaptation, the `Adapt` feature was designed. This feature enabled the changes to take place at the testbed level. They were made to Cerno's surroundings by either *removing* known instances of objects or *adding* general classes of objects. This process facilitated the running of the cognitive architecture for a pre-defined number of times (deliberative cycles), pausing it, adding or removing a number of objects or instances, and then resuming Cerno's operation. The number of domain model objects was incremented until there was *5* of each, giving a total of **15** objects in the simulation world (anything more would have been too cluttered). Upon completion of the tests, the inferred numbers for each object and its instances were recorded. Figure 6.4 illustrates the obtained graph based on the experimental results.

It can clearly be seen how close the findings using Cerno were to the actual number of objects and instances in its environment. This further validates CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about the number of objects and their instances. It, essentially, demonstrates how CernoCAMAL could be adaptive to changes in its enclosure – a capability that CAMAL does not possess.

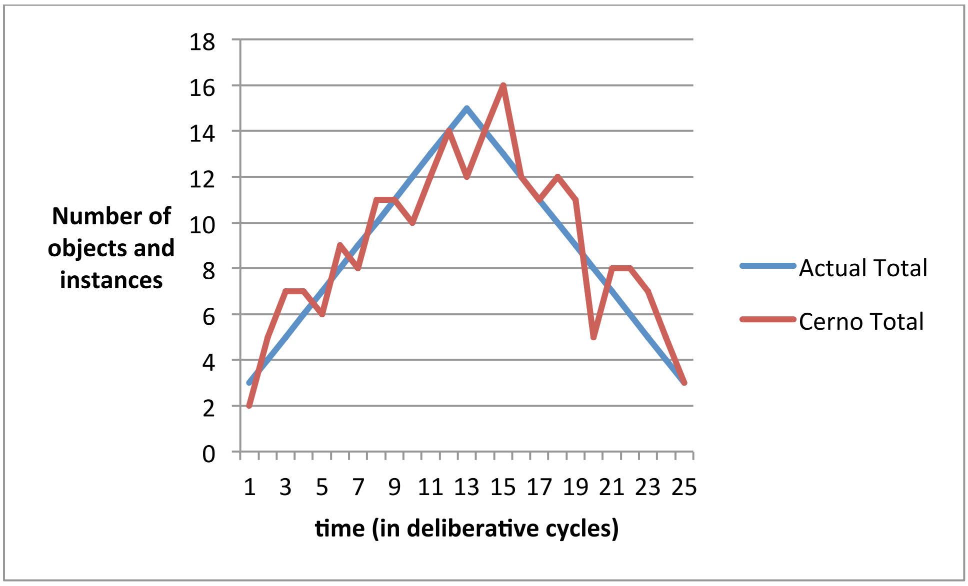

Figure 6-4: Second Adaptability Experiment

It is important to note that these demonstrations and experiments show no drastic fluctuations between the inferred number of objects and their instances and the actual numbers pre-determined by the experimenter. This implies a consistent accuracy in how close the findings using Cerno were to the actual number of objects and instances in its environment, lending support to validate CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about the number of objects and their instances.

6.2.4 Probabilistic Motivator Norm Experiments

This succession of experiments investigates whether the inclusion of a probabilistic motivator norm yields a higher overall performance, success count, task effectiveness, and goal achievement. In a previous CAMAL spin-off work (Venkatamuni 2008) the concept of metacognition was investigated as a powerful catalyst for control of motivational-BDI architectures, with respect to reasoning, planning, decision-making, and learning. It was concluded, based on extensive experimental results, that using a metacognitive (reflective) layer would improve the performance of the CAMAL architecture in general, as well as in terms of specific metrics, e.g. life expectancy and resource collection.

In this work, however, a slightly different methodology is employed to test the same concept, in that the reactive component of CernoCAMAL does not actually deliberate to determine which norm should be used in the motivational blackboard. Instead, the probabilistic norm is pre-programmed prior to runtime; meaning it is defined in the domain model (hand-coded). This manual incorporation is highlighted in the final chapter as a potential future improvement. Figures 6.5 and 6.6 show the obtained graphs based on experimental results. It can clearly be seen that there is a decrease in the number of failure counts in time, plus an increase in the number of success counts in time that occurred.

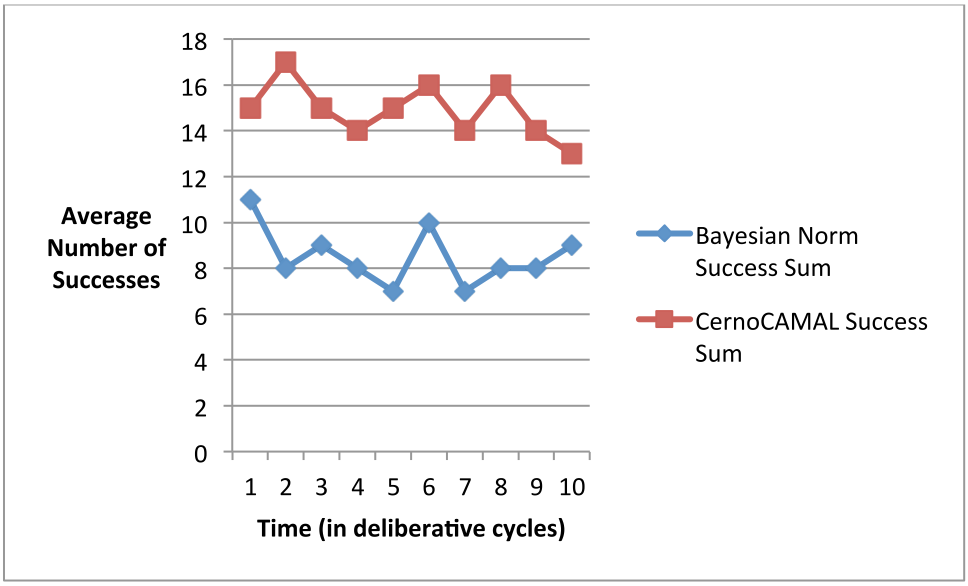

Figure 6-5: Overall Increase in the Number of Successes

These results along with the second set of results below serve to demonstrate the overall advantage of the integration of a shallow probabilistic norm that takes into account the inferred degrees of belief.

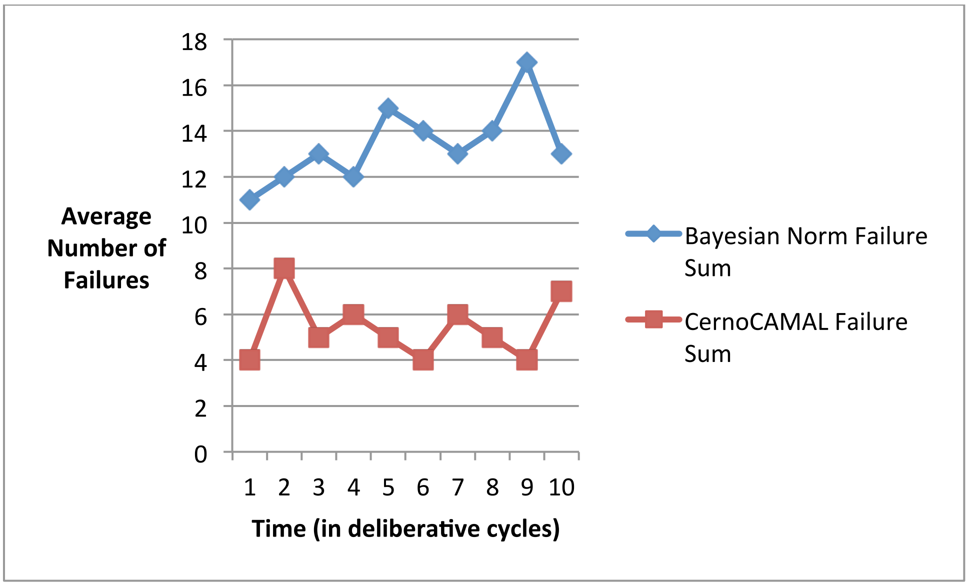

Figure 6-6: Overall Reduction in the Number of Failures

6.3 MobileSim Experiments

This section presents the ARIA robotic experiments carried out using the CernoCAMAL cognitive agent (Cerno) in the MobileSim testbed. A number of experiments were performed over a number of cycles, and some internal variables and statistics were recorded. From the obtained results, the two cognitive architectures of CAMAL and CernoCAMAL can be compared and contrasted.

6.3.1 Validation of CPR in ARIA MobileSim Testbed

Similar CPR tests are performed in the MobileSim environment to ensure that the CernoCAMAL's Probabilistic reasoner can use the EBS and assumed degrees of belief, infer posterior probabilities correctly, assign them to the appropriate belief descriptors, and reason probabilistically about the number of objects and their instances that may be present in the environment. Cerno was allowed to operate in the MobileSim virtual world that consisted of a varying number of (virtual) robots, objects, and obstacles.

CernoCAMAL runs on a P3DX robot, whereas an Amigo robot wonders around as an agent. There are also `5` objects of different sizes present in the MobileSim environment. The aim for Cerno here is to distinguish the robot (agent) from objects (and obstacles and walls). Cerno is a cognitive agent equipped with a means of reasoning probabilistically about objects and their instances in its enclosure. It is, therefore, expected that over time the findings using Cerno in the MobileSim testbed about a probable number of objects and robots are close to the actual numbers known to the experimenter. Note that it is not attempted here to emulate a similar setting (e.g. same number of objects) to the predator-prey simulation environment. The point of exploiting the ARIA MobileSim testbed is to provide a simulation environment different to that of predator-prey, in both features and terrain and also the robotic nature of it. Each experiment was run for `5` minutes.

The configuration of having `5` objects (MobileSim's default, randomly-appearing objects) and `1` robot (agent) can be represented as:

```
belief ( apriori_prob ( object, 1/2 ), assumption, 1, 0.5 ).

belief ( apriori_prob ( robot,  1/2 ), assumption, 1, 0.5 ).
```

The reactive component was pre-configured with the following goals:

```
find ( object )          avoid ( object )

find ( robot )           avoid ( robot )
```

This initial configuration assumes that if there is one item, then for Cerno it is equally likely to be a default object or a robot, hence the probability of any domain object being present is `1/2` or `0.5` with a `0.5` degree of belief since the statements are assumptive. CernoCAMAL, upon receiving goal- and task-oriented reactive feedback interprets them and reasons about the number of objects and robots that may be present in the environment. A sample run of the program, demonstrating that Cerno operates as expected, is presented below. The code snippets represent the running of the program over `5` minutes, that demonstrate Cerno operates as expected. For clearer illustration, `belief( apriori_prob )` clauses have been removed.

```
-------------------------------------------------------------------

[ found(object), found(robot), found(object) ].

found(object)     -->   object1

found(robot)      -->   robot1

found(object)     -->   object2

==> [ object1, object2, robot1 ]

-------------------------------------------------------------------

[ found(robot), hit(object), found(object) ].

found(robot)      -->   refers to robot1

hit(object)       -->   refers to either object1 or object2

found(object)     -->   object3

==> [ object1, object2, object3, robot1 ]

-------------------------------------------------------------------

[ found(object), found(robot), hit(object) ].

found(object)     -->   refers to either object1 or 2 or 3

found(robot)      -->   refers to robot1

hit(object)       -->   refers to object4

==> [ object1, object2, object3, object4, robot1 ]

-------------------------------------------------------------------

[ found(robot), lost(object), near(object) ].

found(robot)      -->   refers to robot1

lost(object)      -->   refers to object1 and 2 and 3 and 4

near(object)      -->   refers to previously-lost object1

==> [ object1, robot1 ]

-------------------------------------------------------------------
```

```
[ near(robot), found(object), near(object) ].

near(robot)       -->   refers to robot1

found(object)     -->   refers to previously-lost object2

near(object)      -->   refers to previously-lost object3

==> [ object1, object2, object3, robot1 ]

------------------------------------------------------------------

[ near(robot), lost(object), near(object) ].

near(robot)       -->   refers to robot1

lost(object)      -->   refers to object1 and 2 and 3

near(object)      -->   refers to object4

==> [ object4, robot1 ]

------------------------------------------------------------------

[ near(robot), lost(robot), near(object) ].

near(robot)       -->   refers to robot1

lost(robot)       -->   refers to robot1

near(object)      -->   refers to previously-lost object1

==> [ object1, object4 ]

------------------------------------------------------------------

[ near(robot), hit(robot), near(object) ].

near(robot)       -->   refers to previously-lost robot1

hit(robot)        -->   refers to robot1

near(object)      -->   refers to previously-lost object2

==> [ object1, object2, object4, robot1 ]

------------------------------------------------------------------
```

```
[ hit(robot), found(object), near(object) ].

hit(robot)        -->   refers to robot1

found(object)     -->   refers to either object1 or 2 or 4

near(object)      -->   refers to previously-lost object3

==> [ object1, object2, object3, object4, robot1 ]

-----------------------------------------------------------------

[ lost(robot), lost(object), near(object) ].

lost(robot)       -->   refers to robot1

lost(object)      -->   refers to object1 and 2 and 3 and 4

near(object)      -->   refers to previously-lost object4

==> [ object4 ]

-----------------------------------------------------------------

[ hit(robot), found(object), hit(object)].

hit(robot)        -->   refers to previously-lost robot1

found(object)     -->   refers to previously-lost object1

near(object)      -->   refers to previously-lost object2

==> [ object1, object2, object4, robot1 ]

-----------------------------------------------------------------

[ hit(object), found(object), hit(robot) ].

hit(object)       -->   refers to either object1 or 2 or 4

found(object)     -->   refers to previously-lost object3

hit(robot)        -->   refers to robot1

==> [ object1, object2, object3, object4, robot1 ]

-----------------------------------------------------------------
```

6.3.2 Goal Achievement Success and Failure

A succession of experiments were carried out to evaluate CernoCAMAL's overall performance, in terms of goal success and failure and consequently task effectiveness. The objective here was to assess the efficacy of the CernoCAMAL architecture over the original CAMAL in a tangible manner, by using success and failure counts. In other words, efficacy is measured in quantitative terms here, as the greater number of successes and lower number of failures. A general sample set of obtained experimental results are presented in Figures 6.7 and 6.8 to enable the comparison of the two cognitive architectures.

In the ARIA MobileSim world, the term *success* refers to one of the following:

1. The set goal (defined in the domain model) is achieved. The new beliefs explicitly state that. This situation is known as `Explicit Goal Success`.

2. The set goal (defined in the domain model) is achieved. The new beliefs state that the negation of the goal negation has been achieved. This situation is known as `Double Negation Success`.

3. The intention (defined in the association construct) is accomplished. The new beliefs state that. This situation is known as `Explicit Intention Success`.

4. The avoid-collisions intention (defined in the association construct) is accomplished. The new beliefs state that. This situation is known as `Avoid-Collisions Success`. [14]

[14] It is noteworthy that since MobileSim has a built-in avoid-collisions mechanism, the number of `Avoid-Collisions Failure` dropped to zero in all of the experiments.

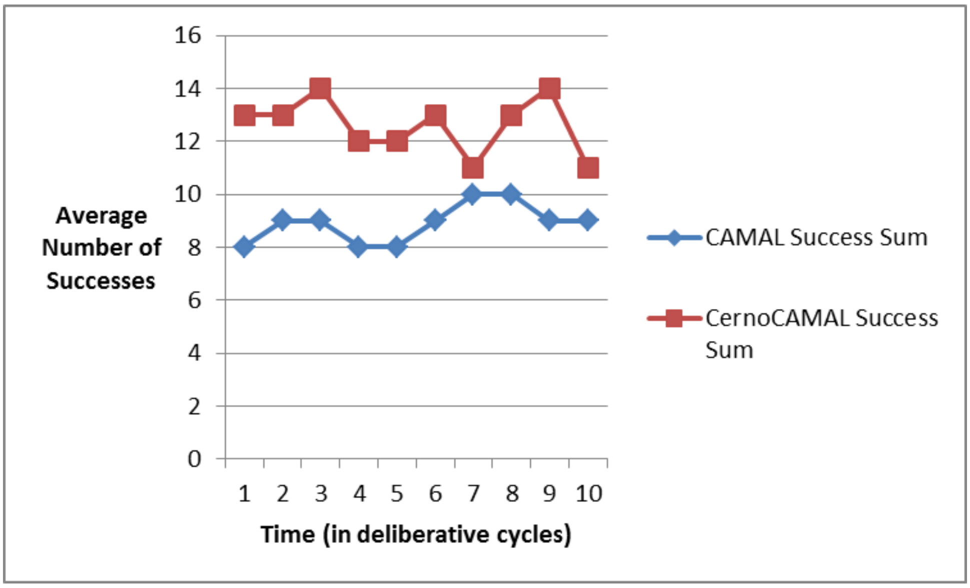

Figure 6-7: Overall Increase in the Number of Successes

These results serve to demonstrate a clear increase in the number of successes in time that occurred, directly supporting the overall advantage of the CernoCAMAL architecture over CAMAL in terms of success counts.

Similarly, the term *failure* refers to one of the following:

1. The set goal (defined in the domain model) is not achieved. The new beliefs explicitly state that. This situation is known as `Explicit Goal Failure`.

2. The set goal (defined in the domain model) is not achieved. The new beliefs state that the negation of the goal has been achieved. This situation is known as `Goal Negation Failure`.

3. The set goal (defined in the domain model) is achieved, but on the wrong object. The new beliefs state that. This situation is known as `Wrong-Object Goal Failure`.

4. The intention (defined in the association construct) is not accomplished. The new beliefs state that. This situation is known as `Explicit Intention Failure`.

5. The avoid-collisions intention (defined in the association construct) is not accomplished. The new beliefs state that. This situation is known as `Avoid-Collisions Failure`.

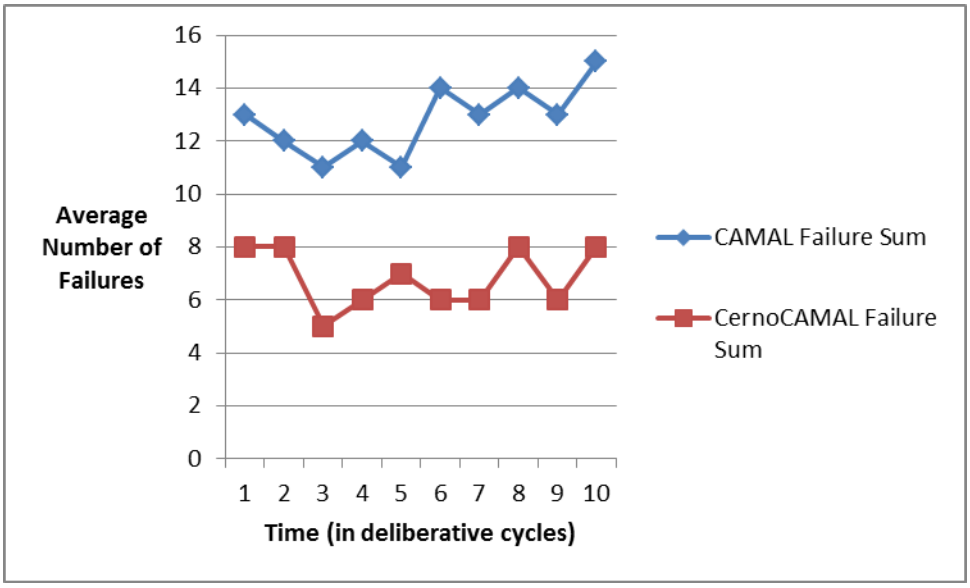

Figure 6-8: Overall Reduction in the Number of Failures

These results serve to demonstrate a clear decrease in the number of failures in time that occurred, directly supporting the overall advantage of the CernoCAMAL architecture over CAMAL in terms of failure counts.

Summarily, the term *task effectiveness* describes if overall the tasks (goals and intentions collectively) were successfully completed. This is inferred based on the increase in the number of successes and the reduction in the number of failures. The two graphs clearly show a decrease in the number of failures in time along with an increase in the number of successes in time that occurred. This outcome demonstrates the overall advantage of the CernoCAMAL architecture over CAMAL in terms of success / failure counts and also task effectiveness.

6.3.3 Population Adaptation Experiments

These experiments investigate CernoCAMAL's ability to adapt in the MobileSim dynamic, uncertain environment. It is dynamic as there is a moving robot (besides the P3DX that is running the CernoCAMAL cognitive architecture) and it is uncertain as there is built-in noise and added random error in the simulation testbed. Adaptation is used in the exact same sense as before (see 6.2.3). Therefore, the findings using Cerno about a probable number of obstacles and robots should be close to the actual numbers pre-determined by the experimenter.

Furthermore, if the controlled experimentation was paused and some modification to the number of specific objects or robots were made, upon resuming experimentation, the findings using Cerno about a probable number of objects and their instances should, again, be close to the actual numbers pre-set by the experimenter.

Similar to predator-prey experiments, the `Experiment` feature of the blackboard monitor was deployed. The number of domain model objects was incremented until there was *5* green objects and one red robot (see figure 5.3). Upon completion of the tests, the inferred numbers for each object and its instances were recorded. Figure 6.9 illustrates the obtained graph based on the experimental results. It can clearly be seen how close the findings using Cerno were to the actual number of objects and robots in its environment.

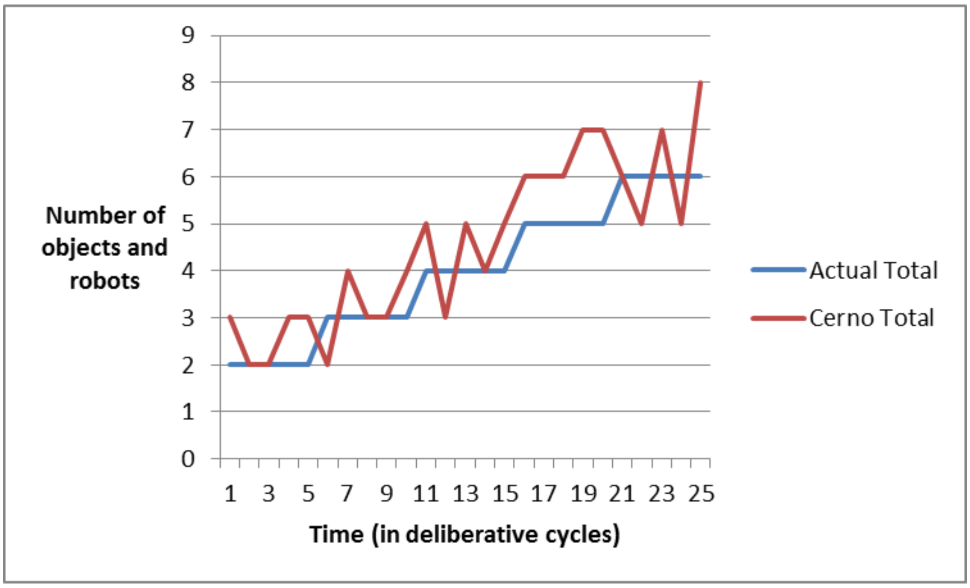

Figure 6-9: First Adaptability Experiment

It can clearly be seen how close the findings using Cerno were to the actual number of objects and instances in its environment. This further validates CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about the number of objects and their instances. It, essentially, demonstrates how CernoCAMAL could be adaptive to changes in its enclosure – a capability that CAMAL does not possess.

For the second part, to perform the add/remove assessments based on the highlighted specific perspective on adaptation, the `Adapt` feature was employed, similar to Section 6.2.3. This feature enabled the changes to take place at the testbed level. They were made to Cerno's surroundings by either removing or adding green objects and red robots. This process facilitated the running of the cognitive architecture for a pre-defined number of times (deliberative cycles), pausing it, adding or removing a number of objects or instances, and then resuming Cerno's operation. Upon completion of the tests, the inferred numbers for each object and its instances were recorded. Figure 6.10 illustrates the obtained graph based on the experimental results.

It can clearly be seen how close the findings using Cerno were to the actual number of objects and robots in its environment. This further validates CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about objects and robots (static and dynamic items). It, essentially, demonstrates how CernoCAMAL could be adaptive to changes in its enclosure – a capability that CAMAL did not possess.

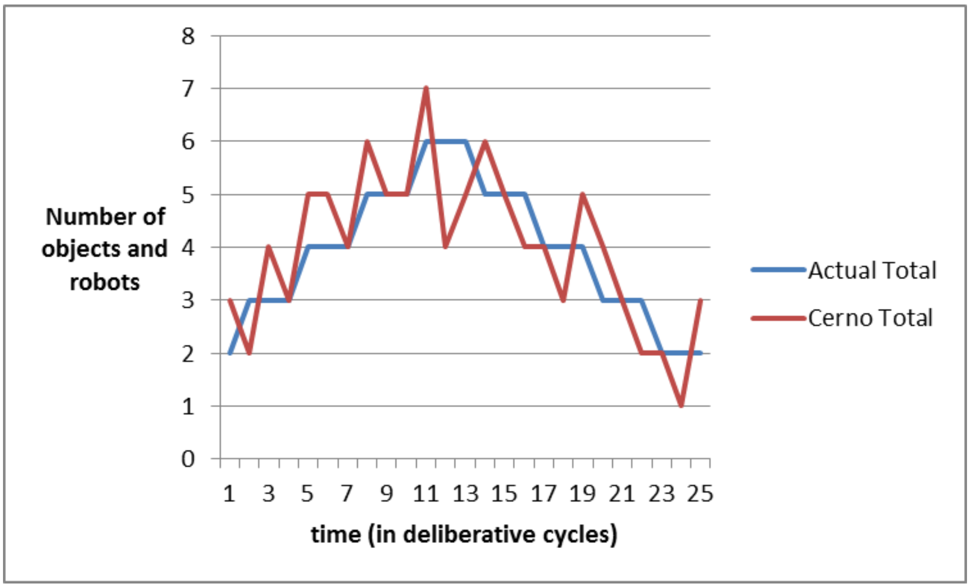

Figure 6-10: Second Adaptability Experiment

6.3.4 Probabilistic Motivator Norm Experiments

As mentioned in Section 6.2.4, a previous CAMAL spin-off work (Venkatamuni 2008) investigated the concept of metacognition as a powerful catalyst for control of Motivational-BDI architectures, with respect to reasoning, planning, decision-making, and learning. Extensive experimental results confirmed that using a reflective (metacognitive) layer would improve the performance of the CAMAL architecture in general, as well as in terms of specific metrics, e.g. Life Expectancy and Resource Collection. The probabilistic norm is pre-programmed prior to runtime; meaning it is defined in the domain model. Figures 6.11 and 6.12 show the obtained graphs based on the experimental results. It can clearly be seen that there is a decrease in the number of failures in time, as well as an increase in the number of successes in time that occurred. This outcome demonstrates the overall advantage of the integration of a probabilistic norm that takes into account the inferred degrees of belief.

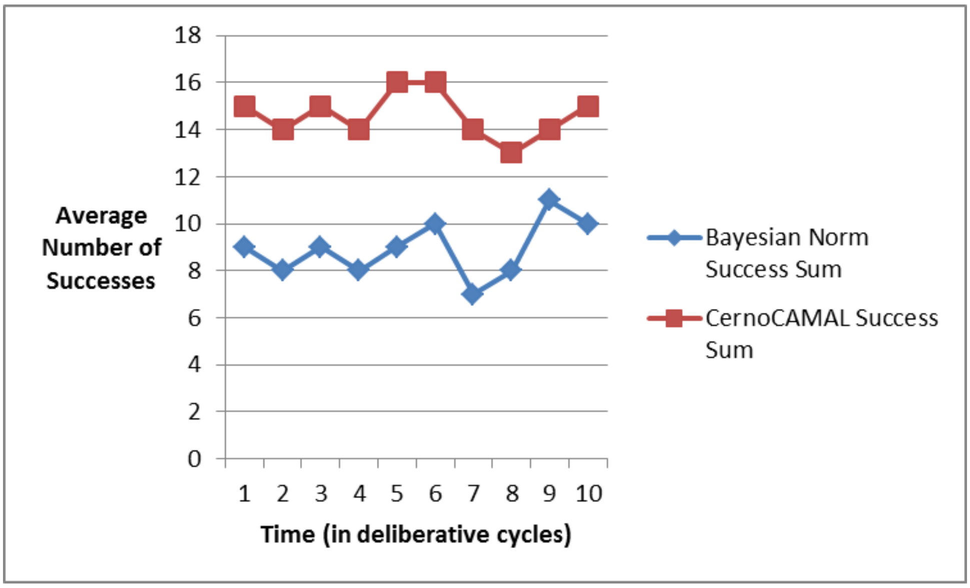

Figure 6-11: Overall Increase in the Number of Successes

These results along with the second set of results on the next page serve to demonstrate the overall advantage of the integration of a shallow probabilistic norm that takes into account the inferred degrees of belief.

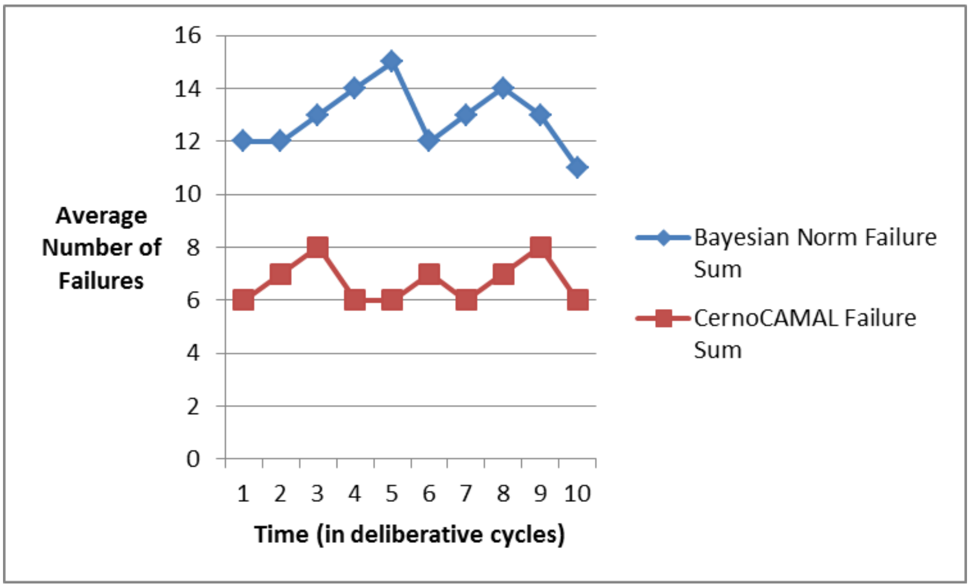

Figure 6-12: Overall Reduction in the Number of Failures

6.4 CernoCAMAL vs. RoboCAMAL

The RoboCAMAL research project (Gwatkin 2009) specifically investigated the anchoring problem in a mobile robot running a simplified version of the CAMAL cognitive architecture (see 3.9.1). The anchoring problem is the problem of linking perceptual data about objects and events to symbolic representations of those objects and events. In other words, anchoring is the establishment and maintenance of a correspondence from sensory data to propositions denoting objects identified from within the sensory data, and actions upon those objects (Coradeschi and Saffiotti 1999, 2003; Shapiro and Ismail 2003). RoboCAMAL is an autonomous mobile robot (AmigoBot – ActiveMedia Robotics) that inhabits a bounded maze, which can include specific known objects e.g. blue ball and further mobile robots.

RoboCAMAL also investigated whether it was possible to learn and adapt in a physical environment. It used two different sensor modalities (sonar, vision, or both) and four different reactive architectures (priority, suppression, aggregate, winner) producing a total of twelve possible reactive sub-architectures to achieve any goal.

The CernoCAMAL research project specifically investigated how CAMAL could be extended to reason probabilistically about domain model objects through perception, and how probability formalism could be integrated into its BDI model to coalesce a number of mechanisms and processes. CernoCAMAL incorporated probabilistic reasoning capability into CAMAL and can, therefore, be used to control the actions of a cognitive mobile robot too. However, due to the different research goals and objectives, experiments performed with RoboCAMAL are not directly mapped onto CernoCAMAL. Only two significant experiments carried out by RoboCAMAL have been identified to be simulated on CernoCAMAL to ascertain whether an improvement or rectification has been accomplished in the process of integrating probabilistic reasoning ability in CAMAL.

The rationale behind this selection was the fact that the first experiment highlighted a *shortcoming* in RoboCAMAL and is, therefore, a sensible point of reference and comparison between the two architectures. Subsequently, the second experiment highlighted a *strength* in RoboCAMAL and is, therefore, an appropriate point to ensure that RoboCAMAL was not compromised by the inclusion of belief affordances. The obtained experimental results are summarized into two graphs that clearly illustrate the argued points.

6.4.1 Challenge for CernoCAMAL

A significant partially-successful learning experiment was identified that was performed using RoboCAMAL, to discover that two domain model objects were mistaken: blue ball and black robot. RoboCAMAL incorrectly identified the blue ball as the black robot due to the large number of black pixels generated by its primitive vision system in response to the blue ball. A similar experiment can, thus, be considered as a method of comparison that shifts the emphasis of this work towards a tangible aspect of the contributions, since RoboCAMAL provides a solid point of reference to an existing published system. The experiment aims to ascertain whether the extension of the belief structure and incorporation of probabilistic reasoning capability using degrees of belief have improved one of RoboCAMAL's drawbacks: misidentification of these two domain model objects.

Of course an obvious way of improving this shortcoming is by adopting a more sophisticated camera and coding a new vision system, but this is beyond the scope of this work. Since the CernoCAMAL project has access to the RoboCAMAL's platform and hardware however, it would be instructive to reflect whether CernoCAMAL architecture could compensate for RoboCAMAL's inadequacy. For this experiment, CernoCAMAL was provided with the following environmental beliefs:

environment(sparse)   &   environment(static)

plus all the possible actions it can take (intentions), to determine whether it can achieve its goal correctly; i.e. without misidentifying the blue ball with the black robot. To carry out this experiment, CernoCAMAL needs to be able to generate a list of associations. To do this, CernoCAMAL was pre-programmed with the goal:

goal(X)

Here, X is the object of the goal, with Y and Z representing the other possible objects. These are highlighted in the Object(s) column of Table 6.2. 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. No associations were pre-programmed. CernoCAMAL was run in six different environments for five minutes each. The experiment was repeated three times for each environment. Table 6.2 shows the six possible environment combinations used for the experiment.

Environment	Object(s)
1	Blue Ball
2	Black Robot
3	Red Robot
4	Blue Ball + Black Robot
5	Blue Ball + Red Robot
6	Blue Ball + Black Robot + Red Robot

Table 6-2: Cerno's Possible Environment Combinations

Each experiment produced a number of associations. In RoboCAMAL, this failed in that a significant number of associations appeared to find the black robot when it was not actually present! Also, some of the associations themselves were wrong. For instance, when the environment only contained one blue ball, it should have been considered sparse. However, the belief `environment(cluttered)` was present in the associations.

This result, basically, highlighted the difficulty when using real sensors. The only way RoboCAMAL could have constructed the wrong belief `found(black_robot)` was if the vision system had identified a black robot. This meant that the vision system had incorrectly identified the object as a black robot when in fact only a blue ball was present.

This also explained the presence of the belief `environment(cluttered)`. If RoboCAMAL believed it had found both the blue ball and the black robot, then it would deduce that the environment is cluttered (as more than one object is present). It was clear that the cause of this failure was the primitive vision system incorrectly identifying a blue ball as a black robot.

In CernoCAMAL, this did not fail in the sense of misclassifying the two objects frequently. There were a number of wrong associations that indicated the two domain model objects were indeed mistaken, but in the large context of the experiment, this number was negligible. The summary results are shown in the graph of Figure 6.13. The minuscule number of wrong associations (`2.3` %) are plotted against the total number of generated associations, to show that the percentage of failures was negligible. The different environments are marked using numbers one to six.

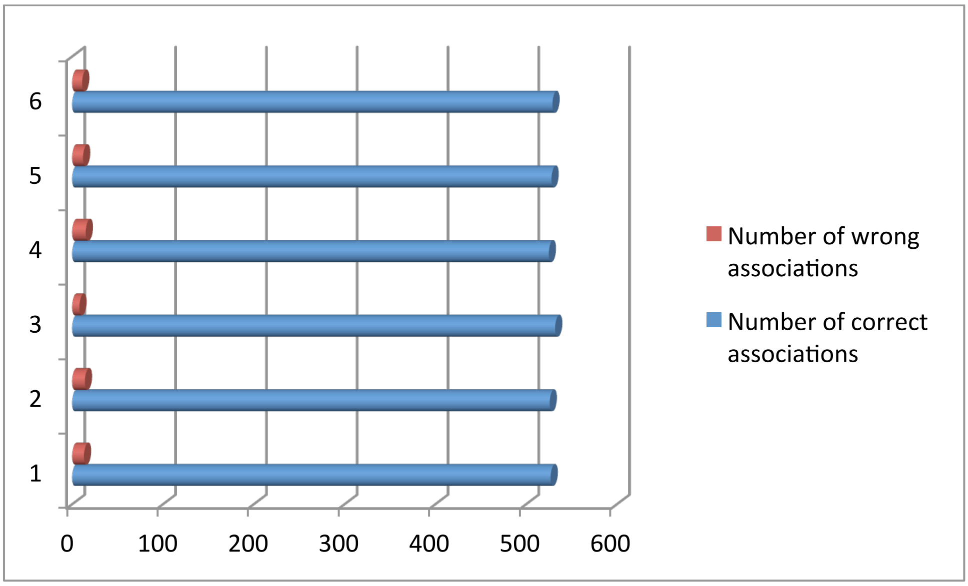

Figure 6-13: Negligible number of wrong associations

6.4.2 Cerno-on-Robo Challenge

The other significant successful experiment performed using RoboCAMAL was adaptation experiment. An important point to note here is that in the context of RoboCAMAL, adaptation referred to its ability to modify its goals to reflect changes in its environment. The obtained experimental results showed that RoboCAMAL had the ability to adapt to a variable environment, and attempted the goals it believed achievable at the right time. Similar to the previous set of experiments, it would be instructive to reflect whether CernoCAMAL architecture might have compromised this RoboCAMAL's capability.

For this experiment, CernoCAMAL architecture was instantiated with three goals:

hit(blue_ball) & hit(red_robot) & hit(black_robot)

plus the correct associations were given to the architecture at start up, to determine whether it can modify its goals to reflect changes in its environment. CernoCAMAL was then allowed to run three minutes in a variable environment. The environment contained the six possible combinations used for the previous experiment, listed in Table 6.2. These combinations were changed at intervals of one minute.

Each experiment produced a number of associations. In RoboCAMAL, this succeeded in that most of the generated associations reflected the changes made at one minute intervals. In CernoCAMAL this, too, succeeded based on the huge percentages of correct-to-incorrect associations that showed CernoCAMAL had modified its goals to reflect the changes made to its maze. There were a number of wrong associations that indicated adaptation took at times up to a whole minute, but in the large context of the experiment, this number was negligible (*1.95* %). The summary results are shown in the graph of Figure 6.14.

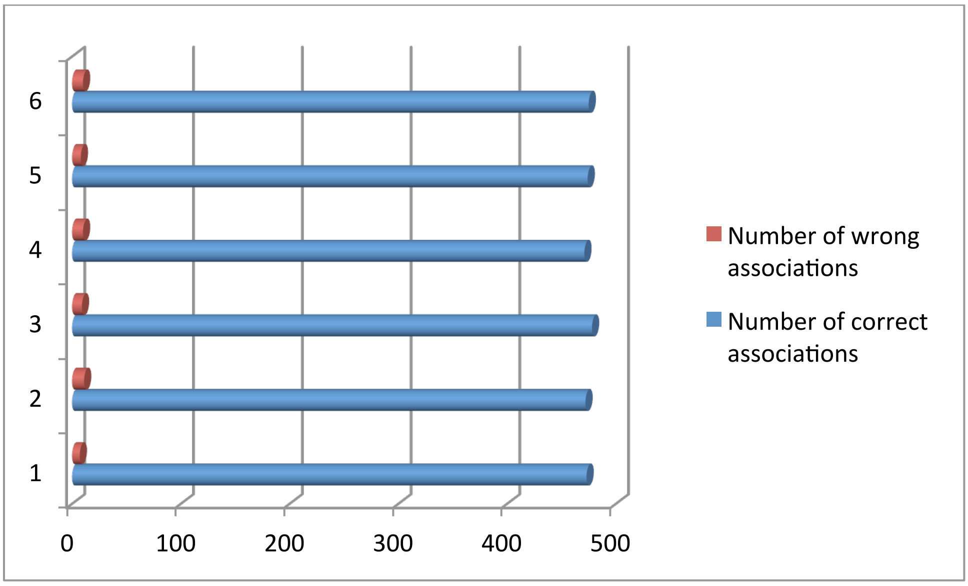

Figure 6-14: Negligible number of wrong associations

6.5 Summary

This chapter began with a brief introduction to a series of experiments with the CernoCAMAL cognitive architecture in the two synthetic testbeds described. The details of the experiments were outlined, starting with the most important one – how the proposed EBS enabled probabilistic object / instance reasoning in the CPR. After discussing the experiments one by one, the obtained graphs were presented. This was followed by repeating the same experimentation process in the second testbed. A final experiment was proposed to use the most recent phase of CAMAL research (RoboCAMAL: Gwatkin 2009). The rationale behind this evaluation method was highlighted, followed by the details of the tests carried out. The chapter concluded with summarising the acquired results.

7 Critical Analysis and Discussion

This chapter begins with reiterating the outcomes of experiments, followed by discussion and analysis of the results. The key capabilities and functions that cognitive architectures should support are presented. Some dimensions along which one should assess and evaluate cognitive architectures are, then, outlined. The chapter concludes with noting some open issues in the area and proposing some directions that future research should take to address them.

7.1 Experimental Results Appraisal

The previous chapter described the various experiments performed with CernoCAMAL along with their obtained results. It began by describing the experiments carried out in the predator-prey testbed, including the comprehensive testing of the EBS and CPR, goal achievement success and failure, population adaptation, and probabilistic motivator norm experiments. This was followed by repeating the same experimentation process in the second testbed – ARIA MobileSim. The chapter concluded with some experiments performed using the RoboCAMAL platform and hardware, as a simple method of comparison that shifts the emphasis of this work towards a tangible aspect of the contributions, since RoboCAMAL provides a solid point of reference to an existing published system.

The experiments that tested the functioning of the EBS and CPR showed that at the deliberative level the integration of the EBS and inclusion of the CPR enabled probabilistic reasoning over the goal- and task-oriented feedback generated by reactive sub-systems. Put simply, they enabled probabilistic reasoning about domain model objects through perception, as well as integrating probabilistic formalism into the BDI model.

The goal achievement success and failure experiments showed that there was a clear decrease in the number of failures in time, as well as a clear increase in the number of successes in time. This outcome demonstrated the overall advantage of the CernoCAMAL architecture over CAMAL in terms of success and failure counts, and also task effectiveness.

The population adaptation experiments showed that the findings using Cerno were very close to the actual number of objects and instances in its environment. This further validated CernoCAMAL's ability to adapt in a dynamic, uncertain environment, with regards to reasoning probabilistically about objects and their instances. It, essentially, demonstrated how CernoCAMAL could adapt to changes in its enclosure, based on the specific definition of 'adaptability' provided earlier.

The probabilistic motivator norm experiments showed that there was a clear decrease in the number of failures in time, as well as a clear increase in the number of successes in time. This outcome demonstrated the overall advantage of the integration of a probabilistic norm that takes into account the inferred degrees of belief.

These results also implied that the revision and re-implementation of the belief-updating, belief-resolution, motivator-selection, and motivator-updating mechanisms worked as expected. In conclusion, the results showed that CernoCAMAL performed as expected. They also indicated the compatibility of the probabilistic BDI model with the affect and motivational models and also affective and motivational valences used throughout CernoCAMAL, ensuring a consistent and systematic metric across all aspects of affect, reasoning, and domain model management.

In a nutshell, the succession of experiments in simulation and robotic testbeds demonstrated improvements and increased efficacy in CernoCAMAL's overall cognitive performance, as well as specific probabilistic reasoning capabilities of the CPR. In applying the CernoCAMAL architecture to RoboCAMAL platform, the minuscule number of wrong associations showed that the percentage of failures was negligible, thus RoboCAMAL's performance improved by the application of CernoCAMAL architecture. Obviously, the 'negligibility' concept is open to interpretation as a question of how little is considered negligible. In the scope of this work, given the number of overall experiments, the ratio of wrong-to-correct associations is considered negligible (2.3 % and 1.95 %).

These results, along with their interpretations, address and validate the research goals and objectives that were set out in the beginning of this thesis. The next section addresses some general evaluation and assessment criteria in order to consider CernoCAMAL in a wider picture.

7.2 Evaluation and Assessment Criteria

It is now time to consider CernoCAMAL in a wider picture, and address some of the research issues and challenges for cognitive architectures in general; in particular what cognitive architectures are actually for and how they should be evaluated and assessed. This is important, because the area of cognitive architectures research supports one of the pivotal goals of Cognitive Science and Artificial Intelligence: the structure, creation, and understanding of synthetic intelligence and artificial cognition for autonomous cognitive agents and mobile robots.

Cognitive architectures refer to the design and organization of the mind and cognition. They show various components and the control flow through those components. They also specify the underlying mechanisms and processes of a cognitive system. Essentially, they provide a working model of some set of cognitive phenomena. They must, therefore, account for the existence and explanation of cognitive behaviours and capacities.

The inclination towards developing cognitive architectures is rooted in the fact that they aim for a reasonable breadth of coverage across a diverse set of tasks and domains. More importantly, they offer accounts of intelligent behaviour at the systems level, rather than at the level of component methods designed for specialized tasks (Langley, Laird, Rogers 2009). Instead of carrying out research that addresses only one issue at a time, we should attempt to unify many findings into a single theoretical framework (i.e. a computational cognitive architecture) and then proceed to test and refine that theory and framework. As highlighted in the Literature Review Chapter, there has been a substantial body of research on cognitive architectures (see 3.9). Clearly, based on the definition of a UTC as a single set of mechanisms and processes for all cognitive behaviour, a fully transparent and functioning model of all cognition would be fantastic, if such a thing is even possible! Realistically, however, in making steps towards that ultimate goal, we need to define, evaluate, and assess those aspects of cognition that are central and pivotal to cognitive architectures. There are varied properties and key capabilities, as a means of evaluating and assessing them, that a cognitive architecture can and should support, including:

7.2.1 Knowledge and Information Accessibility

Langley et al. (2009) in their thorough discussion of cognitive architectures and the research issues surrounding them, identify a central issue that confronts the designer of a cognitive architecture: how to let a cognitive agent access different sources of knowledge and information. In CernoCAMAL this is facilitated by the use of a blackboard that acts like a global workspace for holding the relevant information about the agent's environment, attributes, properties, beliefs, desires, current state, previous states, etc. This blackboard is potentially accessible by all the processes of the agent. Furthermore, the knowledge and information held on it can be divided into several distinct areas: Extended beliefs that the agent can have about its environment, desires that the agent can have about the objects in its environment, intentions that the agent can have to achieve its goals, associations that are used to manage the probabilistic BDI and affect models, and a motivator that contains the result of the operation and execution of the various knowledge sources of the blackboard. The motivator construct of the CernoCAMAL blackboard is such an important part that the blackboard is usually referred to as the 'motivational' blackboard. It not only holds the relevant information and knowledge and allows the various cognitive processes access to the information they require to carry out their tasks, but also controls and coordinates the flow of information through the cognitive architecture.

7.2.2 Generality and Integration

Cognitive architectures as UTCs are intended to support general intelligent behaviour exhibited by cognitive beings, such as humans and animals. Thus, generality is a key aspect for evaluating a candidate cognitive architecture. Such generality factor could be evaluated by constructing testbeds that are designed for a diverse set of tasks, and then testing its behaviour in those environments. The more environments in which the cognitive architecture supports intelligent behaviour, the greater its generality. The CernoCAMAL framework was initially implemented as a situated cognitive agent in a synthetic predator-prey Tile World. It was subsequently interfaced with ARIA's MobileSim environment and applied to a virtual P3DX robot to implement a cognitive robotic agent. The architecture was even mounted on RoboCAMAL hardware to ascertain whether it would improve some of the drawbacks of RoboCAMAL. A succession of experiments in both synthetic simulation and robotic testbeds were carried out to demonstrate improvements in CernoCAMAL's overall cognitive performance, as well as specific probabilistic reasoning capabilities. The results, along with their interpretations, addressed and validated the research goals and objectives that were set out in the beginning of this thesis.

7.2.3 Belief and Degree-of-Belief Reasoning and Updating

Forming beliefs, updating them, and reasoning with them are central cognitive activities that let an agent augment its knowledge base of beliefs along with their plausibility. There can also be drawn (inferred) conclusions from other beliefs or assumptions that the agent already holds. To support such capabilities, a cognitive architecture must first be able to represent beliefs and relationships among them. Subsequently, it must be able to reason about them and eventually update them. A common formalism for encoding belief representation and the relationships between the stated beliefs is first-order logic [15]. In CernoCAMAL, Prolog has facilitated the symbolic expression of and operations on belief statements, including the degree-of-belief incorporation and reasoning. Given that the deliberative component is written in Prolog, CernoCAMAL's belief updating and reasoning are computationally efficient. These belief updating and reasoning operations play an important role, not only when inferring new beliefs but also when deciding whether to maintain existing ones or discard them. Such belief reasoning and maintenance mechanisms are especially important for dynamic and uncertain environments in which situations may change in unexpected ways.

7.2.4 Desire and Goal Reasoning and Updating

In addition to beliefs and belief maintenance operations, a cognitive architecture must also be able to represent desires and goals. These are detailed in the cognitive agent's domain model and possibly generated. In CernoCAMAL, Prolog has facilitated the expression of and operations on goal statements, including the affective value of goal importance (urgency). The goal importance resolution includes reducing this affective value if a goal fails and increasing it upon success.

[15] Other notations include production rules, neural networks, and Bayesian networks.

7.2.5 Intention, Action, and Behaviour

A cognitive architecture must be able to execute intentions, actions, or behaviours in the environment (simulation or real world). Intentions are plans of actions that can be inferred from beliefs and desires of an agent, or explicitly provided (pre-programmed in the domain model). They are akin to plans of actions or possibly reactive sub-architectures that are based on the agent's desires, and have access to the agent's beliefs. If the environment changes and the agent's beliefs are updated, then the agent's intentions should be modified accordingly to prevent the agent from failing to achieve its goals. For example, a mobile robot should have means and skills for navigating from one place to another, or possibly for manipulating its surroundings with actuators and effectors (motors). These may be encoded in the robot's domain model in terms of primitive or component actions, but they may also specify more complex actions or procedures. CernoCAMAL's BDI schema and domain model contain the intentions, actions, and behaviours that the agent is capable of carrying out. These intentions are used in the association constructs, and are acted upon to achieve the agent's desires.

7.2.6 Object Recognition and Classification

A cognitive agent must be able to recognize objects – whether fixtures e.g. walls and obstacle, or dynamic like other robots. It must also be able to recognize situations and events, e.g. finding or hitting an object. Recognition is closely related to classification of objects and events, which involves the assignment of objects and situations to known concepts or categories. For example, in the case of RoboCAMAL (Davis, Gwatkin 2010) that the architecture was used to control a mobile robot, blue-coloured balls were often misclassified as black, due to the similarity between the number of black pixels in their captured images. This was improved by the incorporation of degrees of beliefs in the structure of belief predicates. To support this cognitive capability, a cognitive architecture must provide some way of identifying objects and events. This is obviously testbed-dependant. In the predator-prey testbed, the simulation world has explicit sensing operators with configurable parameters, e.g. sonar and vision, with pre-defined ranges (see 5.2). In the MobileSim testbed, too, the simulation world has explicit sensing operators (see 5.3).

7.2.7 Decision Making, Learning, and Adaptation

To operate in an environment (physical terrain or simulation world) a cognitive agent requires the ability to make decisions and select among available actions. Such decisions are often associated with the recognition of an event or situation, and most cognitive architectures combine the two mechanisms in a perception-deliberation-action cycle (recognize-think-act cycle) that underlies most of cognitive behaviour. To decide what to do next and support decision making, a cognitive architecture must provide some way of representing alternative intentions, actions, or behaviours, and then offer some process for selecting among these alternatives. The first, obviously, determines whether a given intention, action, or behaviour is allowable. The second selects among these allowable alternatives, often by computing an affective numeric score (affordance) and choosing one or more with higher affordances. In CernoCAMAL, the probabilistic BDI, affect, and motivational models are used to enable the decision-making process. The inclusion of degree-of-belief in the structure of its belief predicates (its BDI model) also enables the architecture to select a focused belief set that reflects its current activities, as highlighted by actions, objects, and agents referenced in a current motivator. The motivator, then, enables goal revision and the selection of the next goal, based on goal importance and current beliefs and goal success. The deliberative processing of these constructs allows the selection of an appropriate action related to specific objects and tasks. This, in turn, drives motivator revision using the association construct, which in turn enables belief-desire-intention combinations to be ranked based on the likelihood of their success (insistence – association values).

Ideally, a cognitive architecture should also incorporate some way to improve its decision making through learning and adapting. For example, in the case of RoboCAMAL the architecture used associations to enable some form of reinforcement learning about the effects of its actions upon its environment (Davis, Gwatkin 2010). Another project (Venkatamuni 2008) benefited from a capability that Sloman (2001) refers to as meta management mechanisms. In this CAMAL spin-off work, the concept of metacognition was investigated as a powerful catalyst for control of affective-motivational-BDI architectures, with respect to reasoning, planning, decision-making, and learning. It was concluded that using a metacognitive (reflective) layer would improve the performance of the CAMAL architecture.

As illustrated in the experiments, the same concept helped improve CernoCAMAL's cognitive performance as well, in terms of specific metrics related to this work, despite the fact that there was actually no 'reflection' in choosing a probabilistic metacognitive norm (see 6.2.4 & 6.3.4).

7.2.8 Perception and Perceptual Processing

A cognitive agent may sense the world through different modalities, just as humans use sight, hearing, touch, etc. A cognitive architecture must, therefore, provide some way of identifying objects and events, which is again obviously testbed-dependant. In CernoCAMAL, perceptual data from the testbeds' sensors (reactive layer) are passed to the deliberative component. These perceptual messages are actually posted to the motivational blackboard and reasoned about by the CPR. The belief-update module 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. Put differently, sensory information is mapped onto belief structures. Belief affordances (degrees of belief) define the degree to which the belief statement is believed to be true. The insistence measure (association value) allows the control of external behaviour through the building of associations that link beliefs, goals, and intentions. In a nutshell, the rationality of CernoCAMAL's BDI model and CPR reasoner is modulated with affective mechanisms, allowing belief and apriori updating to structured and controlled environments through the use of its EBS.

In addition to fulfilling all the above evaluation criteria and assessment measures, this work presents a vigorous affect- and affordance-based core for mind and cognition, that consequently addresses belief uncertainty using degrees of belief in the structure of belief predicates. It is also in line with the Gibsonian affect and affordance theory, as well as Davis's theory of affect (Gibson 1979; Davis, Lewis 2003, 2004) since degrees of belief could be considered belief affordances. The next section highlights some of the open research issues that are still to be addressed in this area.

7.3 Open Research Issues and Some Recent Developments

Despite some significant commonalities, the current state of cognitive architectures is exploding with different theories and frameworks, being developed with different goals in mind. The future of cognitive architectures, however, is mainly determined by how the major and immediate challenges for this area of research are dealt with. Section 2.1 stated that nearly thirty years ago, Donald Norman (1980) had set an agenda of important topics for Cognitive Science. He had argued that there were at least twelve issues that should comprise this agenda: consciousness; perception; memory; language; thought; belief; emotion; interaction; performance; learning; skill; and development. This list has played an important role in determining where researchers have focused their work in AI. In addition, Section 2.5 pointed out some of the challenges that were faced by many research groups across the globe undertaking research using autonomous cognitive agents and mobile robots. These are the research issues that are still being actively investigated:

- How to describe enough of the properties of a cognitive agent, its abilities, and its environment, to allow it to make high-level decisions on how to act or what to do next.
- How to store, manipulate, express, and retrieve the knowledge and information about the environment and the agent itself into its cognitive architecture.
- How to include concepts such as common-sense, free-will, and the like.

These, as well as Norman's agenda, are the major themes for this area of research. Other open research issues such as representing beliefs and goals of a cognitive agent have also played an immense role in the shaping of cognitive architectures research, with numerous accounts in the literature (e.g. Simon 1967; Sloman 1987; Newell 1990). Their frameworks and architectures for computational models of cognition have allowed the specification of beliefs and goals to guide the cognitive agent's actions or behaviours. One such architecture based on their framework, ICARUS (Langley, Choi 2006; Choi 2011) as a significant recent development in this field, operates in an explicit goal-oriented fashion in which it uses multiple, reactive, top-level goals. ICARUS recognizes the need for further research on *uncertainty of beliefs* and highlights this area as a potential future work. CernoCAMAL has addressed this, in line with its stated research questions and objectives.

As previously discussed (see Chapters Three & Four), Sloman (1987, 2002) had proposed 'motivators' to resolve conflicts among goals. In contrast to ICARUS, which uses degree of relevance to prioritize goals, Sloman proposed three different measures – urgency, insistence, and intensity – that should affect how a system manages goals. Some cognitive architectures such as CLARION (Sun et al. 2001; Sun 2007; Sun 2009) also provide nomination and retraction of goals, as well as drive and goal mechanisms that correspond to a psychological account of goal nomination. CernoCAMAL not only implements insistence (association value), urgency (goal importance), and intensity (motivator value), but also possesses goal generation and retraction based on how long a goal has been tried and how many times it has succeeded or failed.

Another outstanding recent work (Dittes, Goerick 2011) presents a formalism suitable for both flexible description of hierarchical architecture concepts, as well as functional design of the resulting system integration process. Their formalism uses the CogAff framework as an existing schema for embedding and relating integrated functionalities. Some researchers have also recognized the extent to which the internal and external states of a cognitive agent are influenced by affect and motivation (e.g. Norman, Shallice 1986). Norman and Shallice (1986) presented a detailed model for control of behaviour that included the environmental stimuli, motivational factors, and an attentional system to govern the activation of goals and the selection of action schemas. Simon (1967) proposed goal-terminating and interruption mechanisms that enabled an essentially serial information processor to deal with unpredictable situations in real time. His termination mechanism stops further actions when a goal is achieved, which ICARUS incorporates as one of its basic features (Langley, Choi 2006; Choi 2011). In 2002 Gray and Braver claimed that emotions could prioritize conflicting alternatives and trade-offs. They further argued that the need for integration of emotion in cognitive control aids adaptation to the environment. Except for the interruption mechanism, CernoCAMAL's affect and motivational models provide a comprehensive framework to incorporate emotion and motivation processes, as fully dissected in chapters three and four.

There also are some cognitive architectures that address motivations and goals in different ways to CernoCAMAL. One such system is CLARION (Sun 2007) which incorporates implicitly represented drives and explicitly specified goals. Two s in the architecture interact to nominate goals. A motivational sub-system maintains an implicit, value-based network that relates the state of the world and the strength of an agent's drives. A metacognitive sub-system uses a multiple vote approach to determine the current goal. Each internal drive proposes multiple goals in the order of their assigned numeric preference. The sub-system chooses the goal that receives most votes across all the drives and passes it to the execution module.

During the same year, another architecture was developed that incorporated an explicit goal nomination mechanism (Broersen, Dastani, Hulstijn, and van der Torre 2002) dubbed BOID. This framework explains goals as a result of interactions between beliefs and desires, but the architecture also includes obligations and intentions. An agent in this framework computes beliefs from observations of the world, while desires and obligations that are consistent with these beliefs trigger goals. The system treats previously-generated goals as intentions that it uses to generate successive goals.

7.4 Summary

This chapter began with reiterating the outcomes of experiments, followed by discussion and analysis of the results. The key capabilities and functions that cognitive architectures should support were presented. Some dimensions along which one should assess and evaluate cognitive architectures were then outlined. The chapter concluded with noting some open issues in the area and proposing some directions that future research should take to address them.

8 Summary and Conclusion

This chapter begins with a reminder of the primary aims of the CernoCAMAL research project. The motivation and impetus for this work that was put forth in earlier chapters is re-iterated, followed by highlighting the goals and objectives of this thesis. A summary of the work done on the cognitive architecture under investigation is presented, followed by drawing some conclusions on the outcomes and obtained experimental results. The main contributions of this research work are then revisited, in light of the overall collection of experimental results. The chapter concludes with a few potential future avenues for improving CernoCAMAL further.

8.1 Thesis Summary

This thesis set out to demonstrate one possible way to extend CAMAL and enable it to reason probabilistically about domain model objects through perception. The primary aims of the CernoCAMAL research project were, therefore, to investigate how CAMAL could be extended to reason probabilistically about domain model objects through perception, and how probability formalism could be integrated into its architecture to coalesce a number of mechanisms. The thesis also demonstrated, through experimentation and analysing the obtained experimental results, an improvement and increased efficacy in CernoCAMAL's overall cognitive performance, as well as specific probabilistic reasoning capabilities of its CPR. It was argued that these improvements, particularly in terms of goal success, goal failure, and task effectiveness, occurred as a consequence of incorporating degrees of belief as a means of probabilistic reasoning and inference in CernoCAMAL.

The motivation and impetus for this investigation was the considerable evidence that probabilistic thinking and reasoning was linked to cognitive development and played a role in cognitive functions, such as decision making and learning. This led us to believe that a probabilistic reasoning capability was an essential part of human intelligence. Thus, it should be a vital part of any system that attempted to emulate human intelligence computationally. In other words, probabilistic reasoning is an essential aspect of the process of cognition and, therefore, must be considered in any adequate description of it, such as a computational cognitive architecture.

8.2 Conclusions and Outcomes

Cognition is better viewed as solving probabilistic, rather than logical, inference problems; meaning cognition is better understood in terms of probability theory, rather than in terms of logic (Oaksford, Chater 2007, 2009). The probabilistic approach to cognition has, therefore, become an established approach in recent decades – something that this body of work took advantage of.

This thesis presented a BDI affective-motivational cognitive architecture that could be used to govern artificial minds probabilistically. There are many views on what constitutes a cognitive architecture or the place of affect and motivation in a cognitive architecture though. CernoCAMAL pursued a perspective informed by affective and motivational control states, rationalized by a cognitive model of probabilistic reasoning using degrees of belief. Any thesis that deals with cognition and cognitive architectures needs some explanation as to its *scope* and *focus*. The scope and focus of the current cognitive architecture under investigation was to extend the original overarching cognitive architecture of CAMAL to enable it to reason probabilistically about domain model objects through perception, and also integrate probability formalism into its BDI model to coalesce a number of mechanisms, in line with the Gibsonian affect and affordance theory, as well as Davis's theory of affect (Gibson 1979; Davis, Lewis 2003, 2004).

The succession of experiments in simulation and robotic testbeds established improvements in CernoCAMAL's cognitive performance and probabilistic inference over the original CAMAL. In applying the CernoCAMAL cognitive architecture to RoboCAMAL platform, the minuscule number of wrong associations showed that the percentage of failures was negligible, thus RoboCAMAL performance improved by the application of CernoCAMAL architecture. CernoCAMAL effectively presents a vigorous affect- and affordance-based core for mind, as the BDI model is now valenced via affective values and affordances, allowing the entire BDI schema to run using numeric affective values to prioritize choices over the current belief set. Since affect is used across the entire cognitive architecture as a decision metric, affective values and affordances can be thought of as a currency. The BDI model that lacked an affective decision metric consistent with the affordances used in the affect and motivational models, is now grounded consistently in the use of affect.

8.3 Research Questions Re-visited

There clearly existed a need to address and incorporate probabilistic reasoning and inference in CAMAL. The primary aim of the CernoCAMAL research project was to tackle this need with the formal probability theory. This research therefore attempted to address the following specific research questions in the current cognitive architecture under investigation:

- Can CernoCAMAL reason probabilistically by exploiting the proposed EBS? Can the integration of the proposed EBS facilitate probabilistic reasoning and inference in CernoCAMAL?

  Yes. In light of the overall collection of experimental results, CernoCAMAL can reason probabilistically by exploiting the proposed EBS. In other words, the integration of the proposed EBS facilitated probabilistic reasoning and inference in CernoCAMAL. This was specifically validated and confirmed, as correct operation of the CernoCAMAL's CPR in terms of probabilistic object and instance reasoning was tested comprehensively.

- Can the BDI model run compatibly with the affect and motivational models, and affective and motivational valences used throughout the whole architecture? Can this ensure a consistent and systematic metric across all aspects of affect, reasoning, and domain model management?

  Yes. The correct results and expected operations of the processes indicated the compatibility of the BDI model with the affect and motivational valences used throughout the architecture. This provides a consistent and systematic control language for ordering propositions, selecting goals, constructing a plan of action, forming a focused belief with an updated degree of belief, and prioritising processes. It ensures a consistent and systematic metric across all aspects of affect, reasoning, and domain model management.

- Can the probabilistic deliberation results of the CPR be used for computing changing degrees of belief given apriori, and subsequently using the BDI, affect, and motivational models to determine the agent's intentions, actions, and behaviours?

Yes. The performed CPR tests confirmed and validated that the CernoCAMAL's probabilistic reasoner can deliberate using the EBS and assumed degrees of belief, infer posterior probabilities correctly, assign them to the appropriate belief descriptors, and reason probabilistically about the number of objects and their instances that may be present in the environment.

- Can the CernoCAMAL cognitive architecture be applied to virtual and physical cognitive agents using synthetic testbeds and mobile robots?

  Yes. The succession of experiments in simulation and robotic testbeds, by successfully applying the CernoCAMAL cognitive architecture to virtual and physical cognitive agents, showed improvements and increased efficacy in CernoCAMAL's overall cognitive performance, as well as specific achievements in light of the overall collection of experiments.

8.4 Thesis Contributions and Claims Re-visited

This thesis has made several contributions, which are effectively the extensions and augmentations to the CAMAL architecture. They include the following:

- The integration of the EBS that associates a probability value with every belief statement, in order to represent the degrees of belief numerically.

  The correct results and expected operations of the processes indicated the correct integration of the EBS. The association of probability values with belief predicates represented beliefs numerically, resulting in the compatibility of the probabilistic BDI model with the affect and motivational models and affective and motivational valences and affordances used throughout the architecture.

- The inclusion of the CPR that reasons probabilistically over the goal- and task-oriented feedback generated by reactive sub-systems.

  In light of the overall collection of experimental results, the CPR was included successfully, resulting in CernoCAMAL being able to reason probabilistically over the goal- and task-oriented feedback generated by reactive sub-systems by exploiting the proposed EBS.

- The compatibility of the probabilistic BDI model with the affect and motivational models and affective and motivational valences and affordances used throughout CernoCAMAL.

  The probabilistic BDI model is now valenced via affective values and affordances, allowing the entire BDI schema to run using numeric affective values to prioritize choices over the current belief set. The BDI model that lacked an affective decision metric consistent with the affordances used in the affect and motivational models, is now grounded consistently in the use of affect and therefore compatible with the affect and motivational models and affective and motivational valences used throughout CernoCAMAL, ensuring a consistent and systematic metric across all aspects of affect, reasoning, and domain model management.

The integration of the EBS and CPR necessitated the revision and re-implementation of the belief-updating and belief-resolution mechanisms. It also necessitated the revision and re-implementation of the motivator-selection and motivator-updating mechanisms. Their correct operations was validated and confirmed by CernoCAMAL's various evaluations performed in experimentation testbeds. Also, the succession of experiments in simulation and robotic testbeds, by successfully applying the CernoCAMAL cognitive architecture to virtual and physical cognitive agents, showed improvements and increased efficacy in CernoCAMAL's overall cognitive performance, as well as specific achievements in light of the overall collection of experiments.

8.5 Potential Future Directions

The current CernoCAMAL research has now taken a number of new directions, as researchers pursue their own agenda. These new directions take the original design associated with the overarching CAMAL architecture, together with the concept of *an underlying affect and affordance mechanism that can be used to compare process priority and rank goals and weigh intentions* but re-frame the research according to specific interests or needs (Davis 2010).

Using a more sophisticated mobile robot such as a P3DX (ActiveMedia 2010) along with a new vision system and camera could be the next step, resulting in a more accurate object identification and consequently a deeper perceptual anchoring model. This new perceptual model could combine the input from the improved sensors of the new robot with apriori information included in the domain model. The new robot would be more adaptable and capable of working in unknown and uncertain environments. This is loosely related to the UK Computing Research Committee's Grand Challenge Number Five: *Architecture of Brain and Mind* (GC5 2011).

GC5 is a multidisciplinary attempt to understand and integrate natural intelligence and high-level cognitive processes at various levels of abstraction. The aim is to demonstrate the results of our improved understanding in a succession of increasingly sophisticated working robots. Previously, Professor Aaron Sloman and recently Professor Leslie Smith have been coordinating the efforts and research in this area.

Another next step could be regarding the manual incorporation of shallow probabilistic metacognitive norms (see 4.5.2 & 6.2.4 & 6.3.4). Currently, CernoCAMAL does not deliberate to determine which norm should be used in the motivational blackboard. This means that the probabilistic norms have to be pre-programmed prior to start-up; i.e. hand-coded and defined in the domain model. This manual incorporation could be improved upon by constructing more norms and reasoning to choose one that has already yielded greater success and task effectiveness in the past.

In addition to the above two specific ways of improving the CernoCAMAL architecture, there are still plenty of open issues in cognitive architectures research that deserve attention and effort from researchers in the area, despite the many advances that have occurred during almost four decades of research and work. An outstanding issue is that each existing cognitive architecture exhibits many of the capacities described in this thesis and elsewhere, but few support all of them. However, a cognitive architecture as a UTC was defined as *a single set of mechanisms and processes for all cognitive behaviour.* The research community should perhaps devote more resources to trying to coalesce and unify the existing capacities and capabilities into one universal and comprehensive framework of mind and cognition.

Moreover, methods for the evaluation and assessment of cognitive architectures and their cognitive abilities could be broadened to include more realistic terrains. Metrics like those used in this thesis for experimentation purposes are necessary, but not sufficient to provide an accurate way of comparing and contrasting competing architectures and cognitive systems. Despite evaluating various cognitive performances in different testbeds, more complex environments must be created, both physical and simulated, that exercise these cognitive capabilities and provide realistic opportunities for measurement (Langley, Messina 2004). Experimental comparisons among competing architectures can play an important role in measuring key variables in unbiased and informative ways.

On the positive side, we now have over four decades worth of experience and development with constructing and using a variety of cognitive architectures for a wide range of problems and terrains!

9 References, Bibliography, and Further Reading

ActiveMedia. (2010) *ARIA Reference Manual.* ActiveMedia Robotics.

ActiveMedia. (2010) *MobileSim* http://robots.mobilerobots.com/wiki/MobileSim [Accessed 18th … 25th January 2010].

Agre, P.E. & **Chapman, D.** (1987) PENGI: An Implementation of a Theory of Activity. In *Proceedings of the 6th National Conference on Artificial Intelligence,* 268-272.

Anderson J.R. (1976) *Language, Memory, and Thought.* Hillsdale, NJ: Lawrence Earlbaum Associates.

Anderson J.R. (1990) *The Adaptive Character of Thought.* Hillsdale, NJ: Lawrence Earlbaum Associates.

Anderson J.R. (2007) *How can the human mind exist in the physical universe.* New York: Oxford University Press.

Anderson J.R. & **Lebiere, C.** (1998) *The Atomic Components of Thought.* Hillsdale, NJ: Lawrence Earlbaum Associates.

Anderson J.R. & **Matessa, M.** (1998) In Oaksford, M. & Chater, N. (Eds.), *Rational Models of Cognition.* Oxford University Press.

Anderson J.R. & **Matessa, M.** (1998) In Oaksford, M. & Chater, N. (Eds.), *The Rational Analysis of Categorization and the ACT-R Architecture.* Oxford University Press.

Anderson J.R.; **Bothell, D.**; **Byrne, M.D.**; **Douglass, S.**; **Lebiere, C.** & **Qin, Y**. (2004) An Integrated Theory of the Mind. *Psychological Review,* 111: 1036-1060.

Arbib, M.A. (1975) Artificial Intelligence and Brain Theory: Unities and Diversities. *Biomedical Engineering,* 3: 238-274.

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

Arkin, **R**.**C**. (2005) Moving Up the Food Chain: Motivation and Emotion in Behaviour-based Robots, In *Who Needs Emotions? The Brain Meets the Robot.* Oxford University Press.

Aube, M. (2005) Beyond Needs: Emotions and the Commitments Requirements. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Baars, B.J. (1997) *In the Theatre of Consciousness: The Workspace of the Mind.* Oxford University Press.

Barsalou, L.W. (1999) Perceptual Symbol System. *Behavioural and Brain Sciences,* 22: 577-660.

Bartsch, K. & **Wellman, H.** (1989) Young Children's Attribution of Action to Beliefs and Desires. *Child Development,* 60: 946-964.

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

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

Beck, R. (2000) *Motivation: Theories and Principles.* Pearson Education.

Bevacqua, E.; **de Sevin, E.**; **Pelachaud, C.**; **McRorie, M.**; & **Sneddon, I**. (2010) Building Credible Agents: Behaviour Influenced by Personality and Emotional Traits. In *Proceedings of the International Conference on Kansei Engineering and Emotion Research (KEER'10).*

Blumenthal, A.L. (1977) *The Process of Cognition.* Prentice Hall.

Bourgne, G. (2003) *Affect-Based Multi-Agent Architecture for a 5-aside Football Simulation*. M.Sc. Thesis, Department of Computer Science, University of Hull, UK.

Bower, G. (1994) Some Relations Between Emotions and Memory. In Ekman, P. & Davidson, R.J. (Ends). *Nature of Emotion.* Oxford University Press.

Bratman, M. (1987) *Intentions, Plans, and Practical Reason.* Harvard University Press.

Broersen J.; **Dastani, M.**; **Hulstijn, J.**; & **van der Torre, L**. (2002) Goal Generation in the BOID Architecture. *Cognitive Science Quarterly,* 2: 428-447.

Brooks, R.A. (1986) A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation,* 2(1): 14-23.

Brooks, R.A. (1989) A Robot That Walks: Emergent Behaviour from a Carefully-Evolved Network. *Neural Computation,* 1(2): 253-262.

Brooks, R.A. (1990) Elephants Don't Play Chess. In Maes, P. (Ed.), *Designing Autonomous Agents.* MIT Press, Massachusetts.

Brooks, R.A. (1991) Intelligence without Representation. *Artificial Intelligence,* 47(2): 139-159.

Brooks, R. A. (1991) Intelligence without Reason. In *Proceedings of the International Joint Conference on Artificial Intelligence,* 1: 569-595.

Brooks, R.A. (1991) New Approaches to Robotics. *Science,* 253: 1227-1232.

Brooks, R.A. (1997) From Earwigs to Humans. *Robotics and Autonomous Systems,* 20: 291-304.

Brooks, R.A. (1999) *Cambrian Intelligence: The Early History of the New AI.* MIT Press, Massachusetts.

Brooks, R.A. & **Lynn, A.S.** (1994) Building Brains for Bodies. MIT AI Lab, Cambridge, Massachusetts USA, *Autonomous Robots,* 1(1): 7-25.

Camras, L.A. (1992) Expressive Development and Basic Emotions. *Cognition and Emotion,* 6: 269-283.

Cassimatis, N.L.; **Trafton, J.**; **Bugajska, M.** & **Schultz, A**. (2004) Integrating Cognition, Perception, and Action through Mental Simulation in Robots. *Journal of Robotics and Autonomous Systems,* 49: 13-23.

Chater, N.; **Tenenbaum, J.B.** & **Yuille, A.** (2006) Probabilistic Models of Cognition: Conceptual Foundations. *Trends in Cognitive Sciences,* 10(7): 287-291.

Choi, D. (2011) Reactive Goal Management in a Cognitive Architecture. *Cognitive Systems Research (Special Issue on Complex Cognition),* 12(3-4): 293-308.

Church, K.W. & **Hanks, P.** (1990) Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics,* 16(1): 22-29.

Clancy, W. (1997) *Situated Cognition: On Human Knowledge and Representations.* Cambridge University Press.

Clocksin, W.F. (2004) Memory and Emotion in the Cognitive Architecture. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Clark, A. (1989) *Micro Cognition: Philosophy, Cognitive Science, and Parallel Distributed Processing.* MIT Press, Massachusetts.

Clark, A. (1998) Embodiment and the Philosophy of Mind. In O'Hear, A. (Ed.), *Current Issues in Philosophy of Mind: Royal Institute of Philosophy Supplement 43,* Cambridge University Press.

Clore, G.L. (1994) Why Emotions are Never Unconscious. In Ekman, P. & Davidson, R.J. (Ends). *Nature of Emotion.* Oxford University Press.

Coradeschi, S. & **Saffiotti, A.** (1999) Anchoring Symbols to Vision Data by Fuzzy Logic. In Hunter, A. & Parsons, S. (Eds.), *Qualitative and Quantitative Approaches to Reasoning with Uncertainty.* Springer-Verlag.

Coradeschi, S. & **Saffiotti, A.** (2003) An Introduction to the Anchoring Problem. *Robotics and Autonomous Systems,* 43: 85-96.

Corkill, D.D. (1991) Blackboard Systems. *AI Expert,* 6(9): 40-47.

Corkill, D.D. ; **Gallagher, K.Q.** & **Murray, K.E.** (1986) A Generic Blackboard Development System. In *Proceedings of the National Conference on Artificial Intelligence,* 1008-1014.

Damasio, A.R. (1994) *Descartes' Error.* Avon Books.

Darwin, C.R. (1872) *The Expression of Emotions in Man and Animals.* London, Murray.

Davis, D.N. (1996) Reactive and Motivational Agents: Towards a Collective Minder. In Muller, J. P. & Wooldridge, M. J. & Jennings, N. R. (Eds.), *Intelligent Agents III: Agent Theories, Architectures, and Languages.* Springer Verlag.

Davis, D.N. (1998) Synthetic Agents: Synthetic Minds. *Systems, Man, and Cybernetics,* 3: 2658-2663.

Davis, D.N. (2000) *Minds have personalities: Emotion is the core.* Department of Computer Science, University of Hull, UK.

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

Davis, D.N. (2002) Computational Architectures for Intelligence and Motivation. In *Proceedings of the 17th IEEE International Symposium on Systems and Intelligent Control.*

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

Davis, D.N. (2004) Why Do Anything? Emotions, Affect, and the Fitness Function Underlying Behaviour and Thought. In *Proceedings of the AISB Symposium on Affective Computing.*

Davis, D.N. (Ed.) (2005) *Visions of Mind: Architectures for Cognition and Affect.* Information Science Publication.

Davis, D.N. (2008) Linking Perception and Action through Motivation and Affect. *Journal of Experimental and Theoretical Artificial Intelligence,* 20(1): 37-60.

Davis, D.N. (2010) Cognitive Architectures for Affect and Motivation. *Cognitive Computation,* 2: 199-216.

Davis D.N. & **Gwatkin, J.** (2010) RoboCAMAL: A BDI Motivational Robot. *Journal of Behavioural Robotics,* 1(2): 116-129.

Davis, D.N. & **Lewis, S.C.** (2003) Computational Models of Emotion for Autonomy and Reasoning. *Informatica,* 27(2): 159-165.

Davis, D.N. & **Lewis, S.C.** (2003) *Computational Modelling of Emotion and Reasoning.* Department of Computer Science, University of Hull, UK.

Davis, D.N. & **Lewis, S.C.** (2004) Affect and Affordance: Architectures without Emotion. In *Proceedings of AAAI 04 Spring Symposium,* Stanford, USA.

Davis, D.N. ; **Sloman, A.** & **Poli, R.** (1995) *Simulating Agents and their Environments.* AISB Quarterly.

Davis, D.N. & **Venkatamuni V.M.** (2007) *Metacognition in a Society of Minds.* Department of Computer Science, University of Hull, UK.

Davis, D.N. & **Venkatamuni V.M.** (2007) *Metacognition, Agents, Animats, and the Society of Minds.* Department of Computer Science, University of Hull, UK.

Descorps-Declère, S.; **Ziébelin, D.**; **Rechenmann, F.**; & **Viari, A.** (2006) Genepi: A Blackboard Framework for Genome Annotation. BMC Bioinformatics, 7: 450.

Dittes, B. & **Goerick, C.** (2011) A Language for Formal Design of Embedded Intelligence Research Systems. *Robotics and Autonomous Systems,* 59 (3-4): 181-193.

Duffy, E. (1941) An Explanation of Emotional Phenomena without the Use of the Concept Emotion. *The Journal of General Psychology,* 25: 283-293.

Ekman, P. (1992) Are There Basic Emotions? *Psychological Review,* 99(3): 550-553.

Ekman, P. (1992) An Argument for Basic Emotions. *Cognition and Emotion,* 6(3/4): 169-200.

Ekman, P. (1993) Facial Expression and Emotion. *American Psychologist (April),* 48(4): 384-392.

Ekman, P. (1994) Moods, Emotions and Traits. In Ekman, P. & Davidson, R.J. (Eds.). *Nature of Emotion.* Oxford University Press.

Ekman, P. (1999) Basic Emotions. In Dalgleish, T. & Power, M. (Eds.), *Handbook of Cognition and Emotion.* John Wiley and Sons.

Ekman, P. & **Davidson, R.J.** (1994) *The Nature of Emotion.* Oxford University Press.

Ekman, P. & **Friesen, W.** (1977) *Facial Action Coding System.* Consulting Psychologists Press.

Ekman, P. & **O'Sullivan, M.** (1991) Who can catch a liar? *American Psychologist,* 46(9): 913-920.

Edmunds, B. (2003) Implementing Free Will. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

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

Elliott, C. (1994) Research Problems in the Use of a Shallow Artificial Intelligence Model of Personality and Emotion. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI 94).* Seattle, Washington, USA.

Fischbein, E. (1975) *The Intuitive Sources of Probabilistic Thinking in Children.* Dordrecht: Reidel.

Fischbein, E. & **Schnarch, D.** (1997) The Evolution with Age of Probabilistic, Intuitively-Based Misconceptions. *Journal of Research in Mathematics Education,* 28(1): 96-105.

Fisher, M. ; **Bordini, R.H.** ; **Hirsch, B.** & **Torroni, P.** (2007) Computational Logics and Agents: A Roadmap of Current Technologies & Future Trends. *Computational Intelligence,* 23(1): 61–91.

Fodor, J.A. (1976) *The Language of Thought.* Hassocks Press, Sussex.

Fodor, J.A. (1983) *The Modularity of Mind.* MIT Press, Massachusetts.

Fodor, J.A. (1987) *Psychosemantics.* MIT Press, Massachusetts.

Fodor, J.A. & **Pylyshyn, Z.W.** (1988) Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition,* 8: 305-336.

Franklin, S. (1995) *Artificial Minds.* MIT Press, Massachusetts.

Franklin, S. (2000) A Consciousness-Based Architecture for a Functioning Mind. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Franklin, S. & **Graesser, A.** (1996) Is it an agent or just a program: A Taxonomy for Autonomous Agents. In Muller, J. P. & Wooldridge, M. J. & Jennings, N. R. (Eds.), *Intelligent Agents III: Agent Theories, Architectures, and Languages.* Springer-Verlag.

Gallese, V. & **Lakoff, G.** (2005) The Brain's Concepts: The Role of the Sensory-Motor System in Conceptual Knowledge. *Cognitive Neuropsychology,* 22(3/4): 455-479.

GC5 (2011) UK Computing Research Committee's Grand Challenge No. 5: Architecture of Brain and Mind, University of Birmingham website, http://www.cs.stir.ac.uk/gc5

Georgeff, M.P. & **Rao, A.S.** (1991) Modeling Rational Agents within a BDI Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning,* 473-484, San Francisco, CA, Morgan Kaufmann.

Georgeff, M.P. & **Rao, A.S.** (1995) BDI Agents: From Theory to Practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS),* 312-319, Washington, USA, IEEE Press.

Georgeff, M.P. & **Rao, A.S.** (1995) The Semantics of Intention Maintenance for Rational Agents. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* 1: 704-710.

Georgeff, M.P.; **Pell, B.**; **Pollack, M.E.**; **Tambe, M.** & **Wooldridge, M.J.** (1999) The Belief-Desire-Intention Model of Agency. In Muller, J. P. & Singh, M. & Rao, A. (Eds.), *Intelligent Agents V: Agent Theories, Architectures, and Languages.* Springer-Verlag.

Gibson, J.J. (1979) *The Ecological Approach to Visual Perception.* Houghton Mifflin.

Gopnik, A. (1993) How We Know Our Minds: The Illusion of the First Person Knowledge of Intentionality. *Behavioural and Brain Sciences,* 16: 1-14.

Gordon, E. & **Logan, B.** (2005) Managing Goals and Resources in Dynamic Environments. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Gratch, J. & **Marsella, S.** (2004) A Domain-Independent Framework for Modelling Emotion. *Journal of Cognitive Systems Research,* 5(4): 269-306.

Gray, J.R. & **Braver, T.S.** (2002) Integration of Emotion and Cognitive Control: A Neuro-Computational Hypothesis for Dynamic Goal Resolution. In Oaksford, M. & Moore, S.C. (Eds.), *Emotional Cognition: From Brain to Behaviour, p. 289-316.*

Gray, G.L.; **McKee, T.E.**; & **Mock, T.J.** (1991) The Future Impact of Expert Systems and Decision Support Systems in Auditing. *Advances in Accounting,* 9, 249–273.

Green, D. (1983) A Survey of Probability Concepts in 3000 Pupils aged 11-16 years. In *Proceedings of the 1st International Conference on Teaching Statistics,* 2: 766-783.

Gruber, T.R. (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition,* 6(2): 199-221.

Gwatkin, J. (2009) *RoboCAMAL: Anchoring in a BDI Motivational Cognitive Robot.* Ph.D. Thesis, Department of Computer Science, University of Hull, UK.

Gwatkin, J. & **Davis D.N.** (2007) *Motivated Control of Multiple Reactive Architectures.* Department of Computer Science, University of Hull, UK.

Haddaway, P. & **Hanks, S.** (1993) Utility Models for Goal-Directed Decision-Theoretic Planners. *Computational Intelligence,* 14(3): 392-429.

Hanks, S.; **Pollack M.E.** & **Cohen P.R.** (1993) Benchmarks, testbeds, Controlled Experimentation, and the Design of Agent Architectures. *AI Magazine,* 14(4): 17-42.

Harnad, S. (1990) The Symbol Grounding Problem. *Physica D,* 42: 335-346.

Haugeland, J. (1985) *Artificial Intelligence: The Very Idea.* MIT Press, Massachusetts.

Hayes-Roth, B. (1995) An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence,* 72(12): 329-365.

Haykin, S.S. (1994) *Neural Networks: A Comprehensive Foundation.* Macmillan College, New York.

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

Izard, C. (1993) Four Systems for Emotion Activation: Cognitive and Non-Cognitive Processes. *Psychological Review,* 100(1): 68-90.

Jaynes, E.T. (1995) *Probability Theory: The Logic of Science.* Springer-Verlag.

Jiang, H.; **Vidal, J.M.**; & **Huhns, M.N.** (2007) eBDI: An Architecture for Emotional Agents. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems,* 1-3, New York, USA: ACM.

Kaebling, L.P. (1989) An Architecture for Intelligent Reactive Systems. In Allen, J. (Ed.), *Readings in Planning.* Morgan Kaufmann.

Kaebling, **L.P**. & **Rosenschein**, **S.J**. (1994) Action and planning in embedded agents. In Maes, P. (Ed.), *Designing Autonomous Agents,* MIT Press, Massachusetts.

Kaplan, C.A. & **Simon, H.A.** (1989). Foundations of Cognitive Science. In Posner, M.I. (Ed.), *Foundations of Cognitive Science,* MIT Press, Massachusetts.

Knill, D.C. & **Richards, W.** (1996) *Perception as Bayesian Inference.* Cambridge University Press.

Koedinger K.R. ; **Anderson, J.R.** ; **Hadley, W.H.** & **Mark, M**. (1997) Intelligent Tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education,* 8: 30-43.

Konolige, **K**. ; **Myers**, **K**. ; **Ruspini**, **E**. & **Saffiotti**, **A**. (1997) The Saphira Architecture: A Design for Autonomy. *Journal of Experimental & Theoretical Artificial Intelligence,* 9: 215-235.

Kortenkamp, **D**. ; **Bonasso**, **R.P**. & **Murphy**, **R**. (1998) *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems.* MIT Press, Massachusetts.

Laird, J.E. (1991) Preface for Special Section on Integrated Cognitive Architectures. *SIGART Bulletin,* 2: 12-123.

Laird, J.E.; **Newell, A.** & **Rosenbloom, P. S.** (1987) SOAR: An Architecture for General Intelligence. *Artificial Intelligence,* 33: 1-64.

Laird, J.E.; **Rosenbloom, P. S.** & **Newell, A.** (1986) Chunking in SOAR: The Autonomy of a General Learning Mechanism. *Machine Learning,* 1: 11-46.

Langley, P. & **Messina, E.** (2004) Experimental Studies of Integrated Cognitive Systems. In *Proceedings of The Performance Metrics for Intelligent Systems Workshop,* Gaithersburg, MD.

Langley, P. & **Choi, D.** (2006a) A Unified Cognitive Architecture for Physical Agents. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence,* Boston, AAAI Press.

Langley, P. & **Choi, D.** (2006b) Learning Recursive Control Programs from Problem Solving. *Journal of Machine Learning Research,* 7: 493-518.

Langley, P.; **Laird, J.E.**; & **Rogers, S.** (2009) Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research,* 10: 141-160.

Lee, C.H.; **Kim, K.**; **Breazeal, C.** & **Picard, R.W.** (2008) ShyBot: Friend-Stranger Interaction for Children Living with Autism. Work-In-Progress in the *Extended Abstract of CHI 2008,* April 5-10, Florence, Italy.

Lewis, R.L. (2001) Cognitive Theory – SOAR. In Smelser, N.J. & Baltes, P.B. (Eds.), *International Encyclopaedia of the Social and Behavioural Sciences,* 2178-2183, Amsterdam, Pergamon.

Lewis, S.C. (2004) *Computational Models of Emotion and Affect.* Ph.D. Thesis, Department of Computer Science, University of Hull, UK.

Lewis, S.C. & **Davis D.N.** (2004) *Computational Modelling of Emotion and Reasoning.* Department of Computer Science, University of Hull, UK.

Magerko, B. ; **Laird, J.E.** ; **Assanie, M.** ; **Kerfoot, A.** & **Stokes, D.** (2004) AI Characters and Directors for Interactive Computer Games. In *Proceedings of the 16th Innovative Applications of Artificial Intelligence Conference,* San Jose, CA: AAAI Press, 877-884.

McCarthy, J. & **Hayes, P.J.** (1969) Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence,* 4: 463-502.

Middleton, D. & **Edwards, D.** (1990) *Collective Remembering.* Sage.

Minsky, M.L. (1985) *The Society of Mind.* Simon & Schuster.

Nader, k. (2003) Memory Traces Unbound. *Trends in Neurosciences,* 26(2): 65-72.

Neisser, U. & **Fivush, R.** (1994) *The Remembering Self: Construction and Accuracy in the Self-Narrative.* Cambridge University Press.

Newell, A. (1980) Physical Symbol Systems. *Cognitive Science,* 4: 135-183.

Newell, A. (1982) The Knowledge Level. *Artificial Intelligence,* 18: 87-127.

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

Newell, A. & **Simon, H.A.** (1972) *Human Problem Solving.* Prentice Hall.

Newell, A. & **Simon, H.A.** (1976) Computer Science as Empirical Enquiry: Symbols and Search. *Communications of the Association for Computing Machinery,* 19(3): 113-126.

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

Norman, D.A. (1980) Twelve Issues for Cognitive Science. *Cognitive Science,* 4: 1-33.

Norman, D.A. & **Shallice, T.** (1986) Attention to Action: Willed and Automatic Control of Behaviour. In Davidson, R.J. ; Schwartz, G.E. & Shapiro, D. (Eds.), *Consciousness and Self-Regulation (pp. 1-18)* New York: Plenum.

Nunes, H. (2001) *Investigation of Motivation in Agents using the Simulation of 5-aside Football*. M.Sc. Thesis, Department of Computer Science, University of Hull, UK.

Oaksford, M. & **Chater, N.** (1999) Ten Years of the Rational Analysis of Cognition. *Trends in Cognitive Sciences,* 3(2): 57-65.

Oaksford, M. & **Chater, N.** (2007) *Bayesian Rationality: The Probabilistic Approach to Human Reasoning.* Oxford University Press.

Oaksford, M. & **Chater, N.** (2009) Précis of *Bayesian Rationality: The Probabilistic Approach to Human Reasoning*. *Behavioural and Brain Sciences,* 32: 69-120.

Oatley, K. & **Johnson-Laird, P.N.** (1987) Towards a Cognitive Theory of Emotions. *Emotion and Cognition,* 1: 29-50.

Peard, R.F. (1995) Student Decision Making in a Game of Chance and Misconceptions in Probabilistic Reasoning. In *Proceedings of the 18th Annual Conference of the Mathematics Education Research Group of Australasia,* 469-475.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems – Networks of Plausible Inference.* Morgan Kaufmann.

Pearl, J. (2003) *Causality: Models, Reasoning, and Inference.* Cambridge University Press.

Pfeifer, R. & **Scheier, C.** (1999) *Understanding Intelligence.* MIT Press, Massachusetts.

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

Picard, R. (2000) *Affective Computing: MIT Media Lab.* [Online] http://affect.media.mit.edu [Accessed February 2008 & September 2010].

Piaget, J. (1928) *Judgment and Reasoning in the Child.* London: Routledge & Kegan Paul.

Piaget, J. & **Inhelder, B.** (1951) *The Origin of the Idea of Chance in Children.* London: Routledge & Kegan Paul.

Pylyshyn, Z.W. (1987) *Robot's Dilemma: The Frame Problem in Artificial Intelligence.* Greenwood Publishing.

Pylyshyn, Z.W. (1989) Computing in Cognitive Science. In Posner, M. (Ed.), *Foundations of Cognitive Science,* MIT Press, Massachusetts.

Reilly, W.S. & **Bates, J.** (1993) Emotion as Part of a Broad Agent Architecture. In *Workshop on Architectures Underlying Motivation & Emotion.* Birmingham University, UK.

Reilly, S. N.; **Bates, J.** OZ Project Official Website: http://www.cs.cmu.edu/afs/cs/project/oz/web/oz.html [Accessed 21st February 2008].

Ritter, F.E. & **Young, R.M.** (2001) Embodied Models as Simulated Users. *International Journal of Human-Computer Studies,* 55: 1-14.

Robinson, P. & **El-Kaliouby, R.** (2009) Computation of Emotions in Man and Machine. *Philosophical Transactions of the Royal Society,* 364: 3441-3447.

Rosenbloom, P.S. (1993) *The SOAR Papers: Research on Integrated Intelligence (Artificial Intelligence).* MIT Press, Massachusetts.

Sansonnet, J.P. & **Bouchet, F.** (2011) Integrating Psychological Behaviours in the Rational Process of Conversational Assistant Agents, In *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference,* Florida, USA.

Selfridge, O.G. (1959) Pandemonium: A Paradigm for Learning. In *Symposium on the Mechanization of Thought Processes,* London: HM Stationary Office.

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

Shapiro, D. & **Ismail, H.O.** (2003) Anchoring in a Grounded Layered Architecture with Integrated Reasoning. *Robotics and Autonomous Systems,* 43: 97-108.

Shapiro, D. & **Langley, P.** (2004) *Symposium on Learning and Motivation in Cognitive Architectures.* Institute for the Study of Learning and Expertise, Palo Alto, CA.

Shaughnessy, J.M. (1981) Misconceptions of Probability: From Systematic Errors To Systematic Errors and Decisions. *Year-Book: Teaching Statistics and Probability,* 90-100.

Simon, H.A. (1967) Motivational and Emotional Controls of Cognition. *Psychological Review,* 74(1): 29-39.

Simon, H.A. (1996) *The Sciences of the Artificial.* MIT Press, Massachusetts.

Simon, H.A. (1999) Production Systems. In Wilson, R. & Keil, F. (Eds.), *The MIT Encyclopaedia of the Cognitive Sciences.* MIT Press, Massachusetts.

Singh, P. & **Minsky, M.** (2002) An Architecture of Diversity for Common sense Reasoning. *IBM Systems Journal.* 41(3): 530-539.

Sloman, A. (1987) Motives, Mechanisms and Emotions. *Cognition & Emotion,* 1: 217-234.

Sloman, A. (1993) Prospects for AI as the General Science of Intelligence. In Sloman, A.; Hogg, D.; Humphreys, G.; Ramsay, A.; Partridge, D. (Eds.), *Prospects for Artificial Intelligence,* IOS Press, Amsterdam, 1-10.

Sloman, A. (1993) The Mind as a Control System. In Hookway, C. & Peterson, D. (Eds.) *Philosophy and the Cognitive Sciences,* 69-110, Cambridge University Press.

Sloman, A. (1994) Explorations in Design Space. In *Proceedings of ECAI,*
11th European Conference on Artificial Intelligence, 578-582.

Sloman, A. (1995) Exploring Design Space and Niche Space.
In *Proceedings 5th Scandinavian Conference on AI,* IOS Press, Amsterdam.

Sloman, A. (1996) What sort of architecture is required for a human-like agent? In Ling, C. & Sun, R. (Eds.) *Proceedings Cognitive Modelling Workshop at the Conference of the American Association for Artificial Intelligence,* Portland, Oregon.

Sloman, A. (1999) Cognitive Architectures. In Wilson, R. & Keil, F. (Eds.), *The MIT Encyclopaedia of the Cognitive Sciences.* MIT Press, Massachusetts.

Sloman, A. (2001) Beyond Shallow Models of Emotion. *Cognitive Processing,* 2: 177-198.

Sloman, A. (2001) *Varieties of Affect and the CogAff Architecture Schema.* School of Computer Science, University of Birmingham, UK.

Sloman, A. (2002) How many separately evolved emotional beasties live within us. In *Emotions in Humans and Artefacts,* 35-114, MIT Press, Massachusetts.

Sloman, A. & **Croucher, M.** (1987) Why Robots Will Have Emotions. In *Proceedings of the 7th International Conference on Artificial Intelligence,* 197-202.

Sloman, A. & **Logan, B.** (1999) Building Cognitively Rich Agents Using the SIM_Agent Toolkit. *Communications of the Association for Computing Machinery,* 42(3): 71-77.

Sloman, A. & **Logan, B.** (1999) Evolvable Architectures for Human-Like Minds. In Hatano, G. & Okada, N. & Tanabe, H. (Eds.), *Affective Minds.* Elsevier: Amsterdam.

Sloman, A.; **Beaudoin, L.** & **Wright, I.** (1995) Computational Modelling of Motive-Management Processes. In Frijda, N. (Ed.), In *Proceedings of the Conference of the International Society for Research in Emotions.* ISRE.

Sripada, C.S. & **Stich, S.** (2005) *Evolution, Culture, and the Irrationality of the Emotions.* Oxford University Press.

Sun, R. (Ed.) (2005) *Cognition and Multi-Agent Interaction: Extending Cognitive Modelling to Social Simulation.* Cambridge University Press.

Sun, R.; **Merrill, E.** & **Peterson, T.** (2001) From Implicit Skills to Explicit Knowledge: A Bottom-Up Model of Skill Learning. *Cognitive Science,* 25: 203-244.

Sun, R. (2007) The Motivational and Metacognitive Control in CLARION. In Gray, W. (Ed.), *Modelling Integrated Cognitive Systems.* Oxford University Press.

Sun, R. (2009) Motivational Representations within a Computational Cognitive Architecture. *Cognitive Computation,* 1: 91-103.

Tambe, M. ; **Johnson, W.L.** ; **Johns, R.M.** ; **Koss, F.** ; **Laird, J.E.** & **Rosenbloom, P.S.** (1995) Intelligent Agents for Interactive Simulation Environments. *AI Magazine,* 16: 15-39.

Taylor, J.G. (2005) A Review of Cognitive Processing in the Brain. In *Proceedings of the 15th International Conference on Artificial Neural Networks,* 3696: 97-102.

Tenenbaum, J.B. & **Mozer, M.C.** (2000) Bayesian Approaches to Cognitive Modelling. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society,* USA.

Truran, J. (1996) Children's Misconceptions about the Independence of Random Generators. In *Proceedings of the 20th International Conference for the Psychology of Mathematics Education,* 4: 331-338.

Truran, K. (1996) Children's Use of a Representative Heuristic. In *Proceedings of the 20th International Conference for the Psychology of Mathematics Education,* 4: 339-346.

Vallverdu, J. & **Casacuberta, D.** (2009) Handbook of Research on Synthetic Emotions and Sociable Robotics. *USA: Idea Group Inc. (IGI).*

VanLehn, K. (Ed.) (1991) *Architectures for Intelligence.* Hillsdale, NJ: Lawrence Erlbaum.

Venkatamuni V.M. (2008) *A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture.* Ph.D. Thesis, Computer Science Dpt., University of Hull, UK.

Wahl, S. & **Spada, H.** (2000) Children's Reasoning about Intentions, Beliefs, and Behaviour. *Cognitive Science Quarterly,* 1: 5-34.

Way, J. (2003) The Development of Young Children's Notions of Probability. In *Proceedings of the 3rd Conference of the European Society for Research in Mathematics Education,* Thematic Group Five.

Wellman, M.P. & **Doyle, J.** (1991) Preferential Semantics for Goals. In *Proceedings of the 9th National Conference on Artificial Intelligence,* 698-703.

Westen, D. (1996) *Psychology: Mind, Brain, and Culture.* John Wiley and Sons Inc.

Whitbrook, A. (2010) *Programming Mobile Robots with Aria & Player.* Springer-Verlag.

Wilson, M. (2002) Six Views of Embodied Cognition. *Psychonomic Bulletin and Review,* 9(4): 625-636.

Wilson, R.A. & **Keil, F.C.** (1999). The MIT Encyclopaedia of the Cognitive Sciences, MIT Press, Massachusetts.

Wimmer, H. & **Perner, J.** (1983) Beliefs about Beliefs: Representation and Constraining Function of Wrong Beliefs in Young Children's Understanding of Deception. *Cognition,* 53: 45-57.

Adamatzky, A. (2005) Para chemistry of Mind: Case Studies of Doxastic and Affective Mixtures. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Agre, P.E. & **Chapman, D.** (1990) What are plans for? In Maes, P. (Ed.), *Designing Autonomous Agents.* MIT Press, Massachusetts.

Alpher, V.S. (1991) Affect, Epistemology, and the Perceptual-Ecological Perspective: Interpersonal Processes and Affordances in Psychotherapy. *Journal of Contemporary Psychotherapy,* 21(2): 101-113.

Ammeraal, L. (2000) *C++ for Programmers.* New York: John Wiley.

Anderson J.R. (1993) *Rules of the Mind.* Hillsdale: Lawrence Earlbaum.

Anderson J.R. (1996) ACT: A Simple Theory of Complex Cognition. *American Psychologist,* 51(4): 355-365.

Anderson, M.L. & **Perlis, D.** (2003) *Symbol Systems.* Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, USA.

Arbib, M.A. (2005) *Who Needs Emotions? The Brain Meets the Robot.* Oxford University Press.

Arkin, R.C. (1989) Navigational Path Planning for a Vision-Based Mobile Robot. *Robotica,* 7: 49-63.

Arkin, R.C. (1989) Motor Schema-Based Mobile Robot Navigation. *International Journal of Robotics Research,* 8(4): 92-112.

Arkin, R.C. (1990) Integrating Behavioural, Perceptual, and World Knowledge in Reactive Navigation. *Robotics and Autonomous Systems,* 6: 105-122.

BACS Project. (1981) Bayesian Approach to Cognitive Systems. http://www.bacs.ethz.ch [Accessed 4th … 29th January 2010].

Baldwin, J.F. (1981) Fuzzy Logic and Fuzzy Reasoning. In Mamdani, E.H. (Ed.), *Fuzzy Reasoning and Its Applications.* Academic Press.

Ballard, L.D. (2008) *Affect-Based Multi-Agent Architecture for a 5-aside Football Simulation*. M.Sc. Thesis, Department of Electrical Engineering, Utah State University, Logan, Utah, USA.

Barber, P.J. & **Legge, D.** (1976) *Perception and Information.* Methuen, London.

Barsalou, L.W. (2008) Grounded Cognition. *Annual Review of Psychology,* 59: 617-645.

Bechtel, W. & **Abrahamsen, A.** (1991) *Connectionism and the Mind: An Introduction to Parallel Processing in Networks.* Blackwell.

Bernhardt, R. & **Albright, S.L.** (1993) *Robot Calibration.* Chapman & Hall.

Bishop, C.M. (1995) *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Bishop, C.M. (2006) *Pattern Recognition and Machine Learning.* Springer-Verlag.

Blakemore, S.J. & **Frith, U.** (2005) *The Learning Brain.* Blackwell.

Boden, M.A. (1970) Intentionality and Physical Systems. *Philosophy of Science,* 37(2): 200-214.

Boden, M.A. (1977) *Artificial Intelligence and Natural Man.* Harvester Press, Hassocks.

Boden, M.A. (1981) *Minds and Mechanisms: Philosophical Psychology and Computational Models.* Harvester Press, Brighton.

Boden, M.A. (1989) *Artificial Intelligence in Psychology: Interdisciplinary Essays.* MIT Press, Massachusetts.

Boden, M.A. (1990) *The Philosophy of Artificial Intelligence.* Oxford University Press.

Boden, M.A. (1996) *The Philosophy of Artificial Life.* Oxford University Press.

Boden, M.A. (2004) *The Creative Mind: Myths and Mechanisms.* Routledge, London.

Boden, M.A. (2006) *Mind as Machine: A History of Cognitive Science.* Clarendon Press, Oxford.

Brafman, R.I.; **Latombe, J.C.**; **Moses, Y.** & **Shoham, Y**. (1997)
Applications of a Logic of Knowledge to Motion Planning under Uncertainty. *Journal of the ACM,* 44(5): 633-668.

Bratko, I. (2001) *Prolog Programming for Artificial Intelligence.* Addison-Wesley, New York.

Bringsjord, S. (1996) *Cognition Is Not Computation: The Argument from Irreversibility.* Department of Philosophy, Psychology, and Cognitive Science, Rensselaer Polytechnic Institute, Troy, USA.

Brooks, R.A. & **Viola, P.A.** (1990) Network-Based Autonomous Robot Motor Control: From Hormones to Learning. In Eckmiller, R. (Ed.), *Advanced Neural Computers.* Elsevier Science Publishers B. V., North Holland.

Brooks, R.A.; **Breazeal, C.**; **Marjanovic, M.**; **Scassellati, B.** & **Williamson, M.** (1999) The Cog Project: Building a Humanoid Robot. In Nehaniv, C. (Ed.), *Computation for Metaphors, Analogy, and Agents,* 52-87, Springer-Verlag.

Brustoloni, J.C. (1991) Autonomous Agents: Characterization and Requirements. Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University.

Bryson, J.J. (2004) Modular Representations of Cognitive Phenomena in Artificial Intelligence, Psychology, and Neuroscience. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Callan, R. (1998) *The Essence of Neural Networks.* Prentice Hall Europe.

Callear, D. (2003) *Prolog Programming for Student – With Expert Systems & Artificial Intelligence Topics.* Thomson, London.

Canamero, D. (1997) Modelling Motivations and Emotions as a Basis for Intelligent Behaviour. In *Proceedings of the 1st International Symposium on Autonomous Agents,* 148-155.

Canamero, L. (2005) Emotion Understanding from the Perspective of Autonomous Robots Research. *Neural Networks,* 18: 445-455.

Canamero, L. & **Gaussier, P.** (2005) Emotion Understanding: Robots as Tools and Models. In Nadel, J. & Muir, D. (Eds.), *Emotional Development: Recent Research Advances.* Oxford University Press.

Cassandra, A.R; **Kaelbling, L.P** & **Kurien, J.A.** (1996) Acting Under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems.*

Cawsey, A. (1998) *The Essence of Artificial Intelligence.* Prentice Hall Europe.

Chalkiadakis, G. & **Boutilier, C.** (2004) Bayesian Reinforcement Learning for Coalition Formation under Uncertainty. *3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04).* 3: 1090-1097.

Chalmers, D.J. (1995a) Facing Up to the Problem of Consciousness. *Journal of Consciousness Studies,* 2(3): 200-219.

Chalmers, D.J. (1995b) The Puzzle of Conscious Experience. *Scientific American,* 62-68.

Chella, A.; **Frixione, M.** & **Gaglio, S.** (2008) A Cognitive Architecture for Robot Self-Consciousness. *Artificial Intelligence in Medicine,* 147-154.

Chitsaz, H.R. (2006) *Geodesic Problems for Mobile Robots.* Ph.D. Thesis. University of Illinois at Urbana-Champaign, USA.

Choi, D. (2010) *Coordinated Execution and Goal Management in a Reactive Cognitive Architecture.* Ph.D. Thesis, Stanford University, USA.

Clancy, W. (1995) A boy scout, Toto, and a bird: How situated cognition is different from situated robotics. In Steels, L. & Brooks, R. A. (Eds.), *The Artificial Life Route to Artificial Intelligence: Building Situated Embodied Agents.* Hillsdale, Earlbaum.

Clark, A. (1990) Connectionism, Competence, and Explanation. *The British Journal for the Philosophy of Science,* 41(2): 195-222.

Clark, A. (1997) *Being There: Putting Brain, Body, and World Together Again.* MIT Press, Massachusetts.

Clark, A. (2001) *Mindware.* MIT Press, Massachusetts.

Clocksin, W.F. (1997) *Clause and Effect – Prolog Programming for the Working Programmer.* Springer-Verlag.

Clocksin, W.F. (1987) *Programming in Prolog – Using the ISO Standards.* Springer-Verlag.

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

Cohen, P.R. & **Levesque, H.** (1990b) Rational Interaction as the Basis for Communication. In Cohen, P.R.; Morgan, J. & Pollack, M. (Eds.), *Intentions in Communication, 221-256.* MIT Press, Massachusetts.

Connell, J.H. (1989) *A Colony Architecture for an Artificial Creature.* MIT Ph.D. Thesis, Department of Computer Science & Electrical Engineering, MIT AI Lab, USA.

Cooper, R.P. & **Fox, J.** (1998) COGET: A Visual Design Environment for Cognitive Modelling. *Behaviour Research Methods, Instruments, and Computers,* 30: 553-564.

Cooper, R.P.; **Fox, J.**; **Farringdon, J.** & **Shallice, T.** (1996) A Systematic Methodology for Cognitive Modelling. *Artificial Intelligence,* 85(1-2): 3-44.

Cooper, R.P. (2002) *Modelling High-Level Cognitive Processes.* Lawrence Erlbaum Associates. (with contributions from Yule, P.G.; Fox, J. & Glasspool, D.W.)

Corrêa F.M. & **Coelho, H.** (2004) A Collective Mental States Framework for Multi-Agents Based Modelling. *Petrópolis – LNCC.*

Cutting, J.E. (1982) Two Ecological Perspectives: Gibson vs. Shaw and Turvey. *The American Journal of Psychology,* 95(2): 199-222.

Das, S.K. (2008) *Foundations of Decision Making Agents – Logic, Probability, and Modality.* World Scientific, Imperial College Press.

Darwiche, A. & **Provan, G.** (1997) Query DAGs: A Practical Paradigm for Implementing Belief-Network Inference, *Journal of Artificial Intelligence Research,* 6: 147-176.

de Carvahlo Ferreira, N.; **Fisher, M.**; & **VanDerHoek, W.** (2008) Specifying and Reasoning about Uncertain Agents. *International Journal of Approximate Reasoning,* 49(1): 35-51.

Dekker, A.H. (2004) Possible Worlds, Belief, and Modal Logic: A Tutorial.

Delcher, A.L.; **Grove, A.J.**; **Kasif, S.** & **Pearl, J.** (1996) Logarithmic-Time Updates and Queries in Probabilistic Networks. *Journal of Artificial Intelligence Research,* 4: 37-59.

Dennette, D.C. (1984a) Cognitive Wheels: The Frame Problem of AI. In Hookway, C. (Ed.), *Minds, Machines, and Evolution: Philosophical Studies.* Cambridge University Press.

Dennette, D.C. (1978) *Brainstorms.* MIT Press, Massachusetts.

Dennette, D.C. (1984b) *Elbow Room.* MIT Press, Massachusetts.

Dennette, D.C. (1987) Cognitive Wheels. In Pylyshyn, Z. W. (Ed.), *Robot's Dilemma: The Frame Problem in Artificial Intelligence.* Greenwood Publishing.

Dennette, D.C. (1993) *Consciousness Explained.* Penguin.

De Raedt, L.; **Kersting, K.**; **Kimmig, A.**; **Revoredo, K.**; **Toivonen, H.** (2008) Compressing Probabilistic Prolog programs, *Machine Learning*, 70(2-3): 151-168.

Descartes, R. (1993) *Meditations on First Philosophy.* Routledge.

Drescher, G.L. (1991) *Made-Up Minds: A Constructivist Approach to Artificial Intelligence.* MIT Press, Massachusetts.

Dreyfus, H.L. & **Dreyfus, S.E.** (1989) Making a Mind versus Modelling the Brain: Artificial Intelligence back at a Branch-Point. *The Artificial Intelligence Debate: False Starts, Real Foundations.* MIT Press, Massachusetts.

Duncan, D.; **Brna, P.** & **Morss, L.** (1994) A Bayesian Approach to Diagnosing Problems with Prolog Control Flow. In *Proceedings of the 4th International Conference on User Modelling,* Cape Cod, Aug '94.

Ehlert, P.A.M. (2001) *Intelligent Driving Agents*. M.Sc. Thesis, Faculty of Information Technology and Systems, Delft University of Technology, The Netherlands.

Eliasmith, C. (2000) Is the Brain Analogue or Digital?. *Cognitive Science Quarterly,* 1(2): 147-170.

Fagg, A.H. & **Arbib, M.A.** (1998) Modelling Parietal Pre-Motor Interaction in Primate Control of Grasping. *Neural Networks,* 11(7-8): 1277-1303.

Feigenbaum, E.A. & **Simon, H.A.** (1984) EPAM-like Models of Recognition and Learning. *Cognitive Science,* 8: 305-336.

Fitzpatrick, P.; **Metta, G.**; **Natale, L.**; **Roa, S.** & **Sandini, G.** (2003) Learning about Objects through Actions – Initial Steps Towards Artificial Cognition. In *Proceedings of IEEE International Conference for Robotics and Automation (ICRA),* 3: 3140-3145.

Flasinski, M. (1997) "Every Man in His Notions" or Alchemists' Discussion on Artificial Intelligence. *Foundations of Science,* 2(1): 107-121.

Fodor, J.A. (1992) A Theory of the Child's Theory of Mind. *Cognition,* 44: 283-296.

Fox, J. (2003) Images of Mind: In Memory of Donald Broadbent and Allen Newell. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Francis, G. & **Spacek, L.** (2006) *Linux Robot with Omni Directional Vision.* Department of Computer Science, University of Essex, UK.

Frankish, K. (2008) Partial Belief and Flat-Out Belief. In Huber, F. & Schmidt-Petri, C. (Eds.), *Degrees of Belief.* Springer-Verlag.

Franklin, S. (1997) Autonomous Agents as Embodied AI. *Cybernetics and Systems,* 28(6): 499-520.

Frijda, N. (1986) *The Emotions.* Cambridge University Press.

Frijda, N. (1995) Emotions in Robots. *Comparative Approaches to Cognitive Science,* MIT Press, pp. 501-516.

Gallant, S.I. (1990) Perceptron-Based Learning Algorithms. *Neural Networks,* 1(2): 179-191.

Gallese, V.; **Fadiga, L.**; **Fogassi, L.** & **Rizzolatti, G.** (1996) Action Recognition in the Pre-Motor Cortex. *Brain,* 119: 593-609.

Gardner, H. (1993) *Frames of Mind: The Theory of Multiple Intelligences.* London: Heineman.

Gershenson, C. (2002) *On the Notion of Cognition.* Centrum Leo Apostel, Vrije Universiteit Brussel, Belgium.

Ghahramani, Z. (1998) Learning Dynamic Bayesian Networks. In Giles, C.L. & Gori, M. (Eds.), *Adaptive Processes of Sequences and Data Structures.* Lecture Notes in Artificial Intelligence, 168-197, Springer-Verlag.

Glasspool, D.W. (2000) The Integration and Control of Behaviour: Insights from Neuroscience and AI. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Gobet, F. & **Lane, P.C.R.** (2005) The CHREST Architecture of Cognition: Listening to Empirical Data. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Godfrey-Smith, P. Model-Based Science and the Representational Theory of Mind. *Philosophy Program, RSSS,* Australian National University & Philosophy Department, Harvard University.

Goleman, D. (1996) *Emotional Intelligence: Why It Can Matter More Than IQ.* Bloomsbury, London.

Gopnik, A. (2003) The Theory-Theory as an Alternative to the Innateness Hypothesis. In Anthony, L. & Hornstein, N. (Eds.), *Chomsky and His Critics.* Blackwell.

Gordon, E. & **Logan, B.** (2004) Managing Goals and Resources in Dynamic Environments. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Griffiths, T.L.; **Kemp, C.** & **Tenenbaum, J.B.** (2008) In Sun, R. (Ed.), *Cambridge Handbook of Computational Cognitive Modelling.* Cambridge University Press.

Haikonen, P.O. (2004) Artificial Minds and Conscious Machines. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Haugeland, J. (1987) An Overview of the Frame Problem. In Pylyshyn, Z. W. (Ed.), *Robot's Dilemma: The Frame Problem in Artificial Intelligence.* Greenwood Publishing.

Heckerman, D. (1995) A Tutorial on Learning with Bayesian Networks. *Microsoft Research, Technical Report:* http://research.microsoft.com/apps/pubs/default.aspx?id=69588 [Accessed 27th & 28th July 2009].

Hommel, B. (2007) Feature Integration across Perception and Action: Event Files Affect Response Choice. *Psychological Research,* 71: 42-63.

Horgan, T.E. & **Tienson, J.** (1989) Representations without Rules. *Philosophical Topics,* 17(1): 147-174.

Horgan, T.E. & **Tienson, J.** (1996) *Connectionism and the Philosophy of Psychology.* MIT Press, Massachusetts.

Huang, J; **Pan, H**; & **Wan, Y.** (2005). An Algorithm for Cooperative Learning of Bayesian Network Structure from Data. *Lecturer Notes in Computer Science,* 3168: 86-94.

Hui, B. & Boutilier, C. (2006) Who Is Asking For Help? A Bayesian Approach to Intelligent Assistance. In *Proceedings of the 11th International Conference on Intelligent User Interfaces* (Sydney, Australia). 186-193.

Hussain, A. (2009) Cognitive Computation: An Introduction. *Cognitive Computation,* 1: 1-3.

Ishizuka, M. & **Kanai, N.** *(1985) Prolog-ELF Incorporating Fuzzy Logic. International Joint Conference on Artificial Intelligence,* 701-703.

Ishizuka, M. & **Kanai, N.** *(1985) Prolog-ELP Incorporating Fuzzy Logic. New Generation Computing,* 3(4): 479-486.

Jaakkola, T.S. & **Jordan, M.I.** (1999) Variational Probabilistic Inference and the QMR-DT Network. *Journal of Artificial Intelligence Research,* 10: 291-322.

Jackson, J.V. (1987) Idea for a Mind. *Siggart Newsletter,* 181: 23-26.

Jaynes, E.T. (1979) Where do we stand on maximum entropy? In Levine, R.D. & Tribus, M. (Eds.), *Maximum Entropy Formalism.* MIT Press, Massachusetts.

Jaynes, E.T. (1982) On the Rationale of Maximum Entropy Methods. In *Proceedings of the IEEE,* 70(9): 939-952.

Jeannerod, M. (1997) *The Cognitive Neuroscience of Action.* Blackwell.

Jensen, F.V. (1996) *Introduction to Bayesian Networks.* Springer-Verlag.

Jiang, H. (2007) *From Rational to Emotional Agents.* Ph.D. Thesis. University of South Carolina, USA.

Jin, Y. & **Thielscher, M.** (2007) Iterated Belief Revision revised. *Artificial Intelligence,* 171: 1-18.

Johnson, C.G. (2003) Does a Functioning Mind Need a Functioning Body. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Jordan, M.; **Ghahramani, Z.**; **Jaakkola, T.S.** & **Saul, L.K.** (1999) An Introduction to Variational Methods for Graphical Models. *Machine Learning,* 37: 183-233.

Karunananda, A.S. (2002) Using an Eastern Philosophy for Providing a Theoretical Basis for Some Heuristics Used in Artificial Neural Networks. *Malaysian Journal of Computer Science,* 15(2): 28-33.

Kaupp, T. (2008) *Probabilistic Human-Robot Information Fusion.* Ph.D. Thesis, School of Aerospace, Mechanical, and Mechatronic Engineering, University of Sydney, Australia.

Kersting, K.; **Raedt, L.D.** & **Kramer, S.** (2000) Interpreting Bayesian Logic Programs. In Getoor, L. & Jensen, D. (Eds.), *Working Notes of the AAAI-2000 Workshop Learning Statistical Models from Rational Data (SRL-00), Austin, Texas, USA,* AAAI Press.

Korb, K.B. & **Nicholson, A.E.** (2003) *Bayesian Artificial Intelligence.* Chapman & Hall / CRC Computer Science & Data Analysis series.

Kristensen, S. (1997) Sensor Planning with Bayesian Decision Theory. *Robotics and Autonomous Systems,* 19: 273-286.

Lakoff, G. (1987) *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* University of Chicago Press.

Lazkano, E.; **Sierra, B.**; **Astigarraga, A.** & **Martinez-Otzeta, J.M.** (2007) On the Use of Bayesian Networks to Develop Behaviours for Mobile Robots. *Robotics and Autonomous Systems,* 55(3): 253-265.

Lauritzen, S.L. (1996) Graphical Models. *Maximum Entropy Formalism.* Oxford University Press.

Lauritzen, S.L. & **Spiegelhalter, D.** (1988) Local Computations with Probabilities on Graphical Structures and their Applications to Expert Systems. *Journal of the Royal Statistical Society,* 50: 157-224.

Lebeltel, O.; **Bessiere, P.**; **Diard, J.** & **Mazer, E.** (2004) Bayesian Robot Programming. *Autonomous Robots,* 16(1): 49-79.

Lewis, R.L. (1997) Specifying Architectures for Language Processing: Process, Control, and Memory in Parsing and Interpretation. In Crocker, M. & Pickering, M. & Clifton, C. (Eds.), *Architectures and Mechanisms for Language Processing,* Cambridge University Press.

Lewis, R.L. (1999) Cognitive Modelling – Symbolic. In Wilson, R. & Keil, F. (Eds.), *MIT Encyclopaedia of Cognitive Sciences,* MIT Press, Massachusetts.

Lin, Y.P. & **Li, X.Y.** (2003) Reinforcement Learning based on Local State Feature Learning and Policy Adjustment. *Information Sciences,* 154(1): 59-70.

Lloyd, D.E. (1989) *Simple Minds.* MIT Press, Massachusetts.

Maes, P. (1989) How to do the right thing. *Connection Science,* 1(3): 291-323.

Maes, P. (1990) Situated Agents can have Goals. In Maes, P. (Ed.), *Designing Autonomous Agents.* MIT Press, Massachusetts.

Maes, P. (1991) The Agent Network Architecture (ANA). SIGART Bulletin, 2(4): 115-120.

Maes, P. (1991) *Designing Autonomous Agents.* MIT Press, Massachusetts. (Individual articles of this book are also listed in this bibliography, in the form: Author, (Year) In Maes, P. (Ed.), *Designing Autonomous Agents.* MIT Press, Massachusetts.)

Maes, P. (1991) A Bottom-Up Mechanism for Behaviour Selection in an Artificial Creature. In Meyer, J. A. & Wilson, S. W. (Eds.), *From Animals to Animats:* In *Proceedings of the 1st International Conference on Simulation of Adaptive Behaviour,* 238-246. MIT Press, Massachusetts.

Maes, P. (1992) Learning Behaviour Networks from Experience. In *Proceedings of the 1st European Conference on Artificial Intelligence,* 48-57.

Maes, P. (1994) Modelling Adaptive Autonomous Agents. *Artificial Life: An Overview,* MIT Press, 135-162.

Mamdani, E.H. & **Sembi, B.S.** (1980) Process Control Using Fuzzy Logic. *Fuzzy Sets: Theory and Applications To Policy Analysis and Information Systems,* 249-265.

Marr, D.C. (1977) Artificial Intelligence: A Personal View. *Artificial Intelligence,* 9(1): 37-48.

Marshall, J.A.; **Brown, G.**, & **Kovacs, T**. (2007) Bayesian Estimation of Rule Accuracy in UCS. In *Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation.* 2831-2834.

Matsuka, T.; **Yamauchi, T.**; **Hanson, C.**, & **Hanson, S.** (2005) *Representing Categorical Knowledge: An fMRI Study.* RUMBA, Psychology Department, Rutgers University – Newark, Department of Psychology, Texas A&M University, USA.

McCarthy, J. (1986). Applications of Circumscription to Formalizing Common-Sense Knowledge. *Artificial Intelligence,* 28: 89-116.

McDermott, D. (1987) We have been Framed! Or Why AI is Innocent of the Frame Problem. In Pylyshyn, Z. W. (Ed.), *Robot's Dilemma: The Frame Problem in Artificial*

Metta, G.; **Sandini, G.**; **Natale, L.**; **Manzotti, R.** & **Panerai, F.** (2001) Development in Artificial Systems. In *Proceedings of EDEC Symposium at the International Conference on Cognitive Science,* Beijing, China.

Meyers, S. (2005) *Effective C++.* 3rd edition. Addison-Wesley.

Microsoft Robotics Developer Centre (http://www.microsoft.com/robotics) On-Line Tutorials [Accessed 6th & 7th December 2008].

Microsoft .Net Centre (http://www.microsoft.com/net) On-Line Tutorials [Accessed 8th & 9th December 2008].

Minsky, M.L. (1974) A Framework for Representing Knowledge. *MIT AI Memo 306,* Massachusetts.

Minsky, M.L. (2006) *The Emotion Machine.* Simon & Schuster, New York.

Muller, J.P. (1965) *The Design of Intelligent Agents: A Layered Approach.* Springer-Verlag.

Murphy, K.P. (2002) *Dynamic Bayesian Network: Representation, Inference, and Learning.* Ph.D. Thesis, Department of Computer Science, University of California in Berkeley, USA.

Nebel, B. & **Babovich-Lierler, Y.** (2004) When are Behaviour Networks Well-Behaved. In *Proceedings of the 16th European Conference on Artificial Intelligence,* ECAI '04, 672-676.

Nichols, E. (2006) *Levels of Granularity in Cognitive Modelling.* Centre for Research on Concept and Cognition, Bloomington, USA.

Nilsson, N.J. (2007) The Physical Symbol System Hypothesis: Status and Prospects. In Lungarella, M. (Ed.), *Fifty Years of AI,* Festschrift, LNAI 4850, 9-17.

Oatley, K. (1992) *Best Laid Schemes.* Cambridge University Press.

Oatley, K. & **Jenkins, J.** (1996) *Understanding Emotions.* Blackwell: Oxford.

Parker, L.E. (1994) *Heterogeneous Multi-Robot Co-operation.* Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, USA.

Parker, L.E. (1998) ALLIANCE: An Architecture for Fault-Tolerant Multi-Robot Co-operation. *IEEE Transactions on Robotics and Automation,* 14(2): 220-240.

Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books.

Penrose, R. (1989) *The Emperor's New Mind.* Oxford University Press.

Penrose, R. (1994) *Shadows of the Mind.* Oxford University Press.

Pfeifer, R. (1993) Studying Emotions: Fungus Eaters. In *Proceedings of the 1st European Conference on Artificial Life,* ECAL '93, 916-927.

Pfeifer, R. (1994) The "Fungus Eater Approach" to Emotion: A View from Artificial Intelligence. *Cognitive Studies,* 1: 42-57.

Pfeifer, R. (1995) Cognition – Perspectives from Autonomous Agents. *Robotics and Autonomous Systems,* 15: 47-70.

Pfeifer, R. (1996) Building Fungus Eaters: Design Principles of Autonomous Agents. In *Proceedings of From Animals to Animats,* SAB '96, Cape Cod, Massachusetts.

Phung, T. (2003) *A Historical-Based Adaptation Mechanism for BDI Agents*. M.Sc. Thesis, School of Computer Science and Information Technology, RMIT University, Melbourne, Victoria, Australia.

Plato. (1955) *Phaedo.* Cambridge University Press.

Poole, D.; **Macworth, A.** & **Randy, G.** (1998) *Computational Intelligence: A Logical Approach.* Oxford University Press.

Price, B. & **Boutilier, C.** (2003) A Bayesian approach to Imitation in Reinforcement Learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence.* 712-720.

Putnam, H. (1981) *Reason, Truth, and History.* Cambridge University Press.

Ribeiro, M.I. (2005) *Obstacle Avoidance.* Instituto de Sistemas e Robótica, Instituto Superio Técnico.

Ritter, F.E. (2004) *Choosing and Getting Started with a Cognitive Architecture to Test and Use Human-Machine Interfaces,* Applied Cognitive Science Lab, School of Information Sciences and Technology, The Pennsylvania State University, University Park, USA.

Robert, C. (1990) An Entropy Concentration Theorem: Applications in Artificial Intelligence and Descriptive Statistics. *Journal of Applied Probabilities,* 27: 303-313.

Rolls, E.T. (1997) *Cognition, Computation, and Consciousness.* Oxford University Press.

Rolls, E.T. (2000) *The Brain and Emotion.* Oxford University Press.

Rolls, E.T. (2005) *Emotion Explained.* Oxford University Press.

Rolls, E.T. (2007) *Memory, Attention, and Decision Making.* Oxford University Press.

Ruiz, A.; **Lopez-de-Teruel, P.E.** & **Garrido, M.C.** (1998) Probabilistic Inference from Arbitrary Uncertainty using Mixtures of Factorized Generalized Gaussians. *Journal of Artificial Intelligence Research,* 9: 167-217.

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

Sanguk, N. & **Piotr, J.G.** (1997) Bayesian Belief Update in Anti-Air Defence. In *Workshop of Machine Learning for User Modelling – The 6th International Conference on User Modelling.*

Saul, L.K.; **Jaakkola, T.** & **Jordan, M.I.** (1996) Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research,* 4: 61-76.

Scheutz, M. (2005) APOC: An Architecture Framework for Complex Agents. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Schmidhuber, J. (2009) Ultimate Cognition *á la* Gödel. *Cognitive Computation,* 1: 177-193.

Schocken, S. & **Finin, T.** (1987) Prolog Meta-Interpreters for Rule-Based Inference Under Uncertainty. *Information Systems Working Papers Series,* Available at SSRN: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1289731

Scott, S. (2001) *Chimpanzee Theory of Mind: A Proposal from the Armchair.* Department of Cognitive Science, Carleton University, Ottawa, CA.

Searle, J.R. (1969) *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press.

Searle, J.R. (1980) Minds, Brains, and Programs. *Behavioural and Brain Sciences,* 3: 417-424.

Searle, J.R. (1983) *Intentionality: An Essay in the Philosophy of Mind.* Cambridge University Press.

Searle, J.R. (1992) *The Re-Discovery of the Mind.* MIT Press, Massachusetts.

Searle, J.R. (1996) *Readings in Language and Mind.* Geirsson, H. & Losonsky, M. (Eds.), Blackwell.

Senglaub, M.E.; **Harris, D.L.** & **Raybourn, E.M.** (2001) Foundations for Reasoning in Cognition-Based Computational Representations of Human Decision Making. Sandia National Laboratories, Albuquerque, NM, USA.

Setiono, R. & **Liu, H.** (1996) Symbolic Representation of Neural Networks. *IEEE Computer Magazine,* 71-77.

Singh, P. & **Minsky, M.** (2005) An Architecture for Cognitive Diversity. In Davis, D. N. (Ed.), *Visions of Mind.* Information Science Publication.

Sloman, A. (1992a) Towards an Information Processing Theory of Emotions. Notes for a Cognitive Science seminar.

Sloman, A. (1992b) What are the phenomena to be explained? Notes for the Attention and Affect Project.

Sloman, A. & **Chrisley, R.** (2003) Virtual Machines and Consciousness. *Journal of Consciousness Studies,* 10(4-5): 133-172.

Sloman, A. & **Humphreys, G.** (1992) The Attention and Affect Project. Appendix to JCI proposal.

Sonka, M.; **Hlacav, V.** & **Boyle, R.** (1993) *Image Processing, Analysis, and Machine Vision.* International Thomson Computer Press.

Sontag, E. (1998) *Mathematical Control Theory: Deterministic Finite Dimensional Systems.* Springer-Verlag.

Spier, E. (1997) *From Reactive Behaviour to Adaptive Behaviour.* Ph.D. Thesis, Balliol College, University of Oxford, UK.

Spratling, M.W. & **Johnson, M.H.** (2006) A Feedback Model of Perceptual Learning and Categorization. *Visual Cognition,* 13(2): 129-165.

Stoytchev, A. & **Arkin, R.C.** (2004) Incorporating Motivation in a Hybrid Robot Architecture. *Journal of Advanced Computational Intelligence & Intelligent Informatics,* 8(3): 269-274.

Sutton R.S. & **Barto, A.G.** (1999) *Reinforcement Learning: An Introduction.* MIT Press, Massachusetts.

Tang, A .; **Nicholson, A.** ; **Jin, Y.** & **Han, J.** (2007) Using Bayesian Belief Networks for Change Impact Analysis in Architecture Design. *Journal of Systems and Software,* 80(1): 127-148.

Taylor, J.G. (2009) Cognitive Computation. *Cognitive Computation,* 1: 4-16.

Thrun, S. (1998) Bayesian Landmark Learning for Mobile-Robot Navigation, *Machine Learning,* 33(1): 41-76.

Thrun, S.; **Burgard, W.** & **Fox, D.** (2005) *Probabilistic Robotics.* MIT Press, Massachusetts.

Tirassa, M. (1997) Mental States in Communication. In *Proceedings of the 2nd European Conference on Cognitive Science,* Manchester, UK.

Tirassa, M. (1999) Taking the Trivial Doctrine Seriously: Functionalism, Eliminativism, and Materialism. *Behavioural and Brain Sciences,* 22: 851-852.

Tirassa, M.; **Carassa, A.** & **Geminiani, G.** (2000) A Theoretical Framework for the Study of Spatial Cognition. *Spatial Cognition: Foundations and Applications,* Amsterdam and Philadelphia, John Benjamin's Publishing Company, 19-31.

Tulving, E. (1983) *Elements of Episodic Memory.* Oxford University Press.

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

VanGelder, T.J. (1998) The Dynamical Hypothesis in Cognitive Science. *Behavioural and Brain Sciences,* 21: 1-14.

Velasquez, J.D. (1996) *Cathexis: A Computational Model for the Generation of Emotions and their Influence in the Behaviour of Autonomous Agents.* Master's Thesis, MIT Media Lab, USA.

Velasquez, J.D. (1997) Modelling Emotions and other Motivations in Synthetic Agents. In *AAAI 97 ACM,* 10-15.

Velasquez, J.D. (1998) Modelling Emotion-Based Decision Making. *Emotional and Intelligent: The Tangled Knot of Cognition,* AAAI Press, Menlo Park, CA, 164-169.

Verbeek, M. (2003) *3APL as Programming Language for Cognitive Robots*. M.Sc. Thesis, Department of Computer Science, Utrecht University, Netherlands.

Wang, Y. (2010) Abstract Intelligence and Cognitive Robots. *PALADYN Journal of Behavioural Robotics,* 1: 66-72.

Watkins, C. (1989) *Learning from Delayed Rewards.* Ph.D. Thesis, Cambridge University, UK.

Wichert, A. (2009) Sub-Symbols and Icons. *Cognitive Computation,* 1: 342-347.

Wilson, S.W. (1985) Knowledge Growth in an Artificial Animal. In *Proceedings of the 1st International Conference on Genetic Algorithms and their Applications,* 16-23.

Wilson, S.W. (1991) The Animat Path to AI. *From Animals to Animats,* MIT Press, Massachusetts.

Wimmer, H. & **Perner, J.** (1985) "John thinks that Mary thinks that..." Attribution of Second-Order Beliefs by 5 to 10 Year-Old Children. *Journal of Experimental Child Psychology,* 39: 437-471.

Yaqub, T. (2008) *Mobile Robot Motion, Perception, and Environment Modelling.* Ph.D. Thesis, School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, Australia.

Zadeh, L.A. (1965) Fuzzy Sets. *Information and Control,* 8(3): 338-353.

Zadeh, L.A. (1981) Test Score Semantics for Natural Language and Meaning Representation via PRUF. *SRI Technical Report 247,* Stanford Research Institute.

Zadeh, L.A. (1983) The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. *Fuzzy Sets and Systems,* 11: 199-227.

Zadeh, L.A. (1983) Common-Sense Knowledge Representation based on Fuzzy Logic. *IEEE Computer,* 16(10): 61-65.

Psychology:

De Houwer, J. and Hermans. D. Cognition & Emotion: Reviews of Current Research and Theories, Psychology Press, 2010

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

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

Frank, R. H. (1988). Passions within reason: The strategic role of the emotions. New York: W. W. Norton & Company.

Frankel, C.B. & Ray, R.D. (2001). Competence, Emotion and Self-Regulatory Architecture. AISB'0 Symposium on Emotion, Cognition and Affective Computing, University of York.

Frijda, N. (1986). The Emotions. Cambridge University Press.

Izard, C. E. (1993) Four Systems for Emotion Activation: Cognitive and Non-cognitive Processes. Psychological Review. 100(1), 68-90.

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

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

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

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

Philosophy:

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

Damasio, A R. (1994) Descartes Error: Emotion, Reason and the Human Brain. Avon Books. New York.

Epstein, A. "Instinct and motivation as explanations for complex behaviour", in The Physiological Mechanisms of Motivation, D.W. Pfaff, Ed., New York: Springer, 1982.

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

Robinson, P. and El Kaliouby, R. Computation of emotions in man and machines, Phil. Trans. R. Soc. B December 12, 2009 364:3441-3447.

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

Wollheim, R. On The Emotions, Yale University Press, 1999

Neuroscience:

Banich,Marie T. (2004). Cognitive Neuroscience and Neuropsychology. Housthon Mifflin Company.

Chapman, C.R. (1996). Limbic processes and the affective dimension of pain. In: G.Carli & M. Zimmerman (Eds.). Towards the Neurobiology of Chronic Pain. 110, 63-81.

Phelps E. Emotion and cognition: Insights from studies of the human amygdala. Annual Review of Psychology, 2006, 24(57):27–53.

Panksepp, J. (1991). Affective neuroscience: A conceptual framework for the neurobiological study of emotions. In K. T. Strongman (Ed.), International Review of Studies on Emotion: Vol. 1 (pp. 59-99). New York: John Wiley & Sons

Panksepp, J. 1998. Affective Neuroscience. Oxford: Oxford University Press.

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

Rolls, E.T. The Brain and Emotion, Oxford University Press, 1999

Other:

Gros, C. Cognition and Emotion: Perspectives of a Closing Gap Cognitive Computation, 2010, 2(2): 78 – 85.

Ziemke, T. and Lowe, R. On the Role of Emotion in Embodied Cognitive Architectures: From Organisms to Robots, Cognitive Computation, 2009, 1:104–117.

10 Appendix I – Blackboard and Domain Model

For the majority of work in simulation and robotic worlds, variations of an initial domain model and motivational blackboard have been used. This appendix describes the components that make up the domain model and motivational blackboard:

`time( T ).`

> details the number of processing cycles that the deliberative component has executed.

`goal object`

> used to describe the focus of a goal, such as `sphere` or `blue_ball`

`object / colour profile`

> used to identify and distinguish objects based on their RGB values, etc.

`sparse_to_cluttered ( 2 ).`

> akin to a threshold that determines the minimum number of objects that must be present for the environment to be considered as cluttered.

`descriptor_set ( sparse, cluttered, dynamic, static ).`

> details all the possible states the environment can be in. The environment is considered dynamic if there is an agent present, e.g. robot.

`object_set ( object, agent, sphere, prey, pred ).`

> provides information on the possible objects that can be present within the environment. The term `object` is used when an unidentified object is detected.

Reactive Cycles

highlights the maximum number of reactive processing cycles that the agent should run for, before it constructs a deliberative statement, as in the *Cycles* element in a motivator construct:

motivator (Goal, Association, Deterministic, Cycles, Intensity).

domain_synonym(found, near).

details the belief *found* can be deduced if the belief *near* is present.

object_predicate_set (avoid, find, hit, attack, eat, herd,
lost, found, near, ate, attacked, destroyed,
location, herded, know_of, instance_of).

details the possible states that various objects can be in. It is further divided into *belief_predicate_set* and *goal_predicate_set*. As the names suggest, *belief_predicate_set* details the possible beliefs about objects and *goal_predicate_set* details the possible goals pertaining to objects. They allow the agent to construct beliefs and goals about its surroundings:

goal_predicate_set (avoid, find, hit, attack, eat, herd).

belief_predicate_set (hit, lost, found, near, ate, attacked,
destroyed, location, herded,
know_of, instance_of).

An example set of intentions is outlined below in the ontological specification of one of the experimental testbeds:

Intention :- methodavoid | methodfind |
methodhit | methodattack |
methodherd | methodeat

negate(X, ~X).

points out which two belief statements are conflicting (opposite), such as:

negate(found(red_robot), lost(red_robot)).

Beliefs Preference Model:

```
belief_preference( perception, assumption ).

belief_preference( perception, deduction ).

belief_preference( deduction, assumption ).
```

Three sources that determine which belief is more reliable:

```
deduction  >  perception  >  assumption
```

Shallow Metacognitive (Reflective) Norms:

```
norm ( belief,    belief_decay_threshold, 15 ).

norm ( goal,      failed_goal_interval,   15 ).
```

fail(Intention).

shows the intention (action or behaviour) that has failed.

goal_minmax.

represents the minimum and maximum values that the goal-importance value can take.

belief, goal, association, motivator, etc...

have been introduced before (see chapter 3).

11 Appendix II – Software Development Engineering

In the beginning, there were only machines and assembly languages! These evolved into higher-level programming languages such as C that were able to break apart programming steps into sub-routines and procedures. The next generation allowed programmers to group collections of sub-routines and procedures into libraries and modules. A subsequent innovation added the notion of object orientation and classes, i.e. data and functions could be grouped into a single object, which further encapsulated the internals of the routines and increased modularity and re-use. Nowadays, C++ is a popular programming language. In the context of Artificial Intelligence and Cognitive Science research, Prolog is as popular.

Prolog is a powerful symbolic programming language that is particularly suited to the notion of logic programming with uncertainties and inexact reasoning. It has built-in pattern matching (unification), automatic search (backtracking), and relational database (knowledgebase). It also embodies the powers of deductive reasoning and inference. Its symbolic-deductive functional mode of operation closely resembles that of human reasoning ad cognition. This is what AI is essentially about, many believe. In addition, Prolog is well-suited for problems that involve objects and symbols, in particular structured objects and relations between them. These features of Prolog make it a powerful programming language for AI, and symbolic, non-numeric, goal-oriented programming in general.

C++ doesn't need much advertisement, particularly after Microsoft Visual Studio provided a neat, standardized, unifying framework. But Prolog, perhaps due to its rather different programming paradigm, deserves a bit of evangelism! Prolog is well-suited for interfacing with C++ enabling the application programmer to implement an object that encapsulates the Prolog services. In addition, Prolog back-end services can be easily integrated with C++ front-end user interface code – an approach to systems and embedded programming that draws on Prolog's strength for rule-based programming. In this way, the C++ interface to the Prolog services would be well-defined, and the implementation could be maintained without impacting the rest of the application. These advantages, collectively, have been major considerations in adopting Prolog and C++ for the purposes of this thesis.

12 Appendix III – Copy Rights and Permissions

Permission to reproduce figures and diagrams from the following key individuals have been obtained:

Dr. D.N. Davis

Prof. Aaron Sloman