THE UNIVERSITY OF HULL

A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture

being a Thesis submitted for the Degree of

Doctor of Philosophy in Computer Science and Engineering

in the University of Hull

by

Vijayakumar Maragal Venkatamuni

B.E (Bachelor of Engineering) in Computer Science and Engineering,

SJCE, Mysore, India (1999).

M.Tech (Master of Technology), National Institute of Technology

Karnataka, (KREC) Surathkal, India (2002).

April 2008

ABSTRACT

This thesis investigates the concept of mind as a control system using the "Society of Agents" metaphor. "Society of Agents" describes collective behaviours of simple and intelligent agents. "Society of Mind" is more than a collection of task-oriented and deliberative agents; it is a powerful concept for mind research and can benefit from the use of metacognition. The aim is to develop a self configurable computational model using the concept of metacognition. A six tiered SMCA (Society of Mind Cognitive Architecture) control model is designed that relies on a society of agents operating using metrics associated with the principles of artificial economics in animal cognition. This research investigates the concept of metacognition as a powerful catalyst for control, unify and self-reflection. Metacognition is used on BDI models with respect to planning, reasoning, decision making, self reflection, problem solving, learning and the general process of cognition to improve performance.

One perspective on how to develop metacognition in a SMCA model is based on the differentiation between metacognitive strategies and metacomponents or metacognitive aids. Metacognitive strategies denote activities such as metacomphrension (remedial action) and metamanagement (self management) and schema training (meaning full learning over cognitive structures). Metacomponents are aids for the representation of thoughts. To develop an efficient, intelligent and optimal agent through the use of metacognition requires the design of a multiple layered control model which includes simple to complex levels of agent action and behaviours. This SMCA model has designed and implemented for six layers which includes reflexive, reactive, deliberative (BDI), learning (Q-learner), metacontrol and metacognition layers.

ACKNOWLEDGEMENTS

This thesis is a product of my obsessive interest in mathematics during my school days and special interest in Artificial Intelligence and Software Engineering subjects after my graduation and post graduation in Computer science and Engineering. I use to spend long hours with plenty of my sleepless nights thinking about solving AI problems and writing a reports. I would like to express my gratitude's that really helped me to make my research successful. This gives me a delighted pressure to write an acknowledgement to this thesis.

First and foremost, I would like to express my gratitude to my research and thesis supervisor Dr. Darryl N Davis. Every person has their role models in their life. I found a man with so powerful, illustrious and excellence in research paradigms. I would like to thank my research panel members Dr. Chandrasekhar K (chair) and Dr.Len Bottaci (technical expert) and special thanks to Dr. Mike Brayshaw for their help and guidance during solving Prolog problems in lab demonstration classes. I would also like to thank many of friends, colleagues, staff and faculty of NEAT lab and department of Computer science, University of HULL, who have supported me for my research. I would like to thank sponsors of my research ORSAS (Overseas Research Students Award Scheme) committee, UK, Director of the research for giving write up bursary, department of Computer Science and University of HULL.

I am grateful to my mother in law, A.Saralavathy and father in law M. Chandrasekar, Chairman of Field marshal K.M Cariyappa educational institutions, SEJI School of nursing, Amar and Diganth TTI, Hanchala gate, Kolar, India, for their constant support during my research. I would also like to thank C. Amarnath, Secretary, C.

Arunprasad, and LLB, S.Lakshmi narayanareddy (SLN), M.N.Shwetha (BDS), M.N Manas, M.N Chaithra, Prof. D.V.Ashok BNMIT, Shylaja K.R, Prof. Mageshwaran Dr.AIT, Bangalore, and Avik Banarjee, India.

I am thankful to my parents Venkatamuni B.A, retired head master and G Sarojamma, Maragal Village, Bangarpet Taluk, Kolar (DT), India, for their encouragement from my childhood towards my successes in education. I am thankful to my brothers Dr M.V.Narayanaswamy, MBBS, Medical Officer of Health, and M.V Thyagaraj, LLM, Advocate, for their encouragement during my research. Finally most important contributions from my wife C. Anitha Vijayakumar, my children's M.V.Diganth Roy, student of New horizon Public school and M.V.Yashmith Roy, for their constant encouragement, love, patience, support, shown during my research period.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF GRAPHS

Chapter 1 Introduction

1.1 Introduction to the Research

Artificial Intelligence originated with the desire to develop artificial minds capable of performing or behaving like an animal or person. It has developed in a number of directions including intelligent systems, reasoning, knowledge representation, and robotics. Cognitive Science originated in the desire to integrate expertise in the traditionally separate disciplines of computer science, psychology and philosophy, in order to advance our insight into cognitive tasks like problem solving, decision making, reasoning, perception, language, memory, learning etc. One perspective on how to do this is to develop cognitive architectures. These cognitive architectures are also called artificial mind models.

Cognitive architectures are designed to be capable of performing certain behaviours and functions based on our understanding of human and non human minds. Important issues in developing cognitive architectures include task effectiveness, goal achievement, and the ability to perform well in novel situations. There are many examples of developed cognitive architectures. Those relevant to this research include SOAR (Newell, 1990), ACT-R (Anderson, 1993), CRIBB (Bartsch and Wellman, 1989), EM-ONE (Singh, 2005), CogAff (Sloman, 2001) and CAMAL (Davis, 2002). Any cognitive or intelligent or robotic architecture can be viewed as a single agent or a large collection of agents.

Intelligent behaviour can be viewed as a combination of more simple behaviours. Imagine a simple reactive agent that can only move towards and collect a resource in the environment. Building an optimal or metacognition agent cannot be done with a single simple agent, as it needs to interact or take a help from other agents. Hence developing SMCA (Society of Mind Cognitive Architecture) can be viewed from the perspective of Minsky (1985), which leads to the development of many different types of simple agents, with different behaviours. Metacognition is useful for framing the constraints for this swarm intelligence. Swarm intelligence requires the inclusion of a mathematical theory of how the group of agents work together to achieve a common goal. Swarm intelligence uses different mathematical algorithms so as to cover all processing and functioning associated with the adopted architecture or mind model (Bedau, 2003, Martinoli, 2001)

"The Society of Mind is more than just collection of theorems. It is a powerful catalyst for Thinking about Thinking" (Singh, 2003).

Hence "Society of Mind" needs a catalyst like metacognition on top of the society of agents. Metacognition is a relatively new buzz word in cognitive theory. Metacognition is defined as thinking about thinking and can be viewed in two ways:

Monitoring a group of agents in an intelligent or cognitive or robotic architecture (i.e. self reflection)

Making changes by adapting effective strategies in that group of agents.

Metacontrol task is a part of metacognition. Metacognition agent is designed to testbed based on metacognitive strategies, such as metacomprehension (remedial action), self regulation (metamanagement) and schema training (meaningful learning).Metacognitive aids or metacomponents are used for the representation of thoughts (Zalta, 2005, Adkins, 2004), that can be made with the help of some aids such as: (1)using an abstraction, metasyntactic variable (matching variables) or metacomponents and (2) goal setting variables such as perceptual range, affect, norms and higher level rules are metacomponents. The term "norm" is an interdisciplinary term, and can be used to refer to a standard principle or a model used for a right action. The executive processes that controls the other cognitive components are responsible for "figuring out how to do a particular task or set of tasks, and then making sure that the task or set of tasks are done correctly". Metacomponents affects on the agent behaviour from a sense of what is important instead of what to do. Metacognition agents follow well aligned norms, perceptual range, metarules, and learning and affect values. A well driven agent will maximize its performance as a consequence of learning to maximize its own reward.

The approach taken addresses these issues through the design and implementation of a model of mind building on the "Society of Agents" metaphor with different behaviours and capabilities encapsulated as micro-agents within an encompassing framework. For example the implemented Society of Mind Cognitive Architecture (SMCA) has reflexive (six behaviours), reactive (eight behaviours), deliberative (fifteen behaviours), perceptual (nineteen behaviours), learning (fifteen behaviours, fifteen agents), metacontrol (thirty behaviours, one agent) and metacognitive (seventy seven behaviours) agents. Indeed, from an extreme perspective in this distributed model of mind is designed from combinations of reflexive, reactive, BDI (Belief, Desire, and Intention) agents (deliberative), perceptual, learner (Q learning), metacontrol, and metacognitive agents. Agent behaviours can be analysed using many different metrics. The major metrics are metabolic activity, competition and social interaction with respect to environment and microeconomics. Application of economics on artificial life to watch adaptive behaviours. This follows the microeconomic regularities such as cost and utility. Testbeds and benchmarks are mainly using for simulating, comparing architectures and outcomes in the field of robotics or cognitive architectures. Pfeiffer (1988) describes the fungus eater concept as a testbed for simulating models in emotion psychology. The fungus world environment allows the principles and behaviours of a robot or simulated animal or any artificial mind simulation to be monitored, measured and compared (Pfeifer, 1996). This research explores metacontrol and metacognition mechanisms in developing optimal agents for the fungus world testbed.

1.2 Research Questions

This research project addresses issues associated with the development of a SMCA distributed model of mind, using the "Society of Mind" approach and in doing so impacts on the following questions.

What is SMCA? What are the principles used for designing a SMCA?

What is the difference between reflexive, reactive, deliberative, learning, metacontrol and metacognition level processes in a Society of Mind cognitive architecture?

What are the different parts of metacognition? How this concept fits with SMCA?

What are BDI models? How can BDI models plan in different circumstances in SMCA?

What are the metrics used for measuring a performance of agents in SMCA?

The above key questions are raised with the intention of providing solutions or at least some steps or progress towards answers. These questions are answered in the chapters of this thesis.

1.3 Roadmap of Thesis

The chapters of this thesis present along with answering the above given questions, material related to these research questions including research background, theoretical and design principles and results from experimenting with a society of agents in a fungus world environment. Each chapter concludes with a summary highlighting the main points of that chapter.

Chapter One gives an introduction to the project, including an overview of the problem area, research questions, the background of the study, a statement of the problem, the purpose of the study and the chapter organization.

Chapter Two discusses the nature of cognition, the qualities of natural and artificial minds, cognitive science and approaches on the study of Mind. It also discusses the use of principles of natural minds in artificial minds such as cost function, goal directed behaviour and learning.

Chapter Three discusses models of artificial minds and different types of cognitive architectures, with a specific focus on one developing cognitive architecture. Cognitive architectures discussed include SOAR (Newell, 1990), ACT-R (Anderson, 1993), CRIBB (Bartech and Wellman, 1989), EM-ONE (Singh, 2005), CogAff (Sloman, 2001) and CAMAL (Davis, 2002). It raises issues related to the design of a testbed with which to conduct experiments.

Chapter Four discusses metacognition, and its relation to metacontrol and learning. This chapter also provides ideas of designing a metacognition on specific cognitive architecture. This Chapter also discusses specific interest on the topics of execution engine and expertise model, Minsky A, B & C-Brain, generic architecture for metacognition and metacomponents.

Chapter Five discusses the definition for society of agents, artificial agents and different types of agents, with a specific focus on developing new society of agents. Previous work relevant to this research includes Brustoloni (1991), Sloman (2002), Franklin (1997) and Minsky (1985). This Chapter also discusses the Minsky's Society of Mind model with focus to the newly developing SMCA (Society of Mind Cognitive Architecture).

Chapter Six discusses the newly developing SMCA (Society of Mind Cognitive Architecture). The design part includes reflexive, reactive, deliberative level agents or BDI models, general structure of the BDI model and metacognition agent general structure. The design also includes metacomponents such as affect, higher level rules for resource set, norms and meaningful learning.

Chapter Seven describes the design of the fungus world testbed and development undertaken enabling experimental setup. It gives experimental setup with parameters for fungus world environment such as: replenish rates, agent performance parameters, output parameters, society of agent's setup in the experiment and general structure of the fungus world simulation.

Chapter Eight discusses the results for three different experimental cases, with analysis, discussions and summary. Three sets of experimental combinations were created: (1) reactive versus BDI agents; (2) experiments on BDI models; and (2) cognition, BDI, and metacontrol versus metacognition agents. The analysis of experiment1 explains how the society of agents behaves in different combinations. The experiment2 on BDI agents follow the principles such as cost function, optimal decision making and decision boundary variables. Experiment3 provides a comparison between cognition and metacognition agents. This explains through result graphs, how the concept of metacognition improves the performance through unification. The experiment also demonstrates how the cognition and metacognition agents demonstrate the "Society of Mind" concept.

Chapter Nine concludes the thesis with discussions of the research work. This chapter presents research contributions to the field of artificial intelligence and cognitive sciences in terms of the research objectives established in the first chapter. This Chapter also discusses certain limitations and future research directions.

Chapter 2 Artificial Minds

2.1 Origin of the Study of Mind

How does a baby learn to recognize its parents? What is the mind? Are artificial minds possible? These are questions that cognitive science and artificial intelligence can address. The AI era started with John McCarthy, who named "Artificial Intelligence" as the new topic for the 1956 Dartmouth conference (Newell, 1957). At the same conference, Alan Newell, J.C Shaw, and Herbert Simon demonstrated the first AI programme (Logic Theorist) that could construct logical proofs from a given set of premises. This event has been interpreted as the first example of a machine performing a cognitive task. A cognitive task is considered to be an element of the mind. The mind is a core concept for the field of cognitive science.

Artificial Intelligence includes many aspirations. Some researchers simply want machines to do the various sorts of things that people call intelligence. Others hope to understand what enables people to do such things. Some researchers want to simplify programming, wondering how to build machines that grow and improve themselves by learning from experience. Why can we not simply explain what we want, and then let our machines do experiments, read some books, or go to school, which is the way that people learn things. According to Minsky (1990), machines today do not do such things.

2.2 Cognitive Science

Cognition is defined as a mental process or activity that involves the acquisition, storage, retrieval, and use of knowledge. The mental processes include perception, memory, imagery, language, problem solving, reasoning, and decision making (Zalta, 2005; Wilson & Kiel, 1999). Cognitive science is the interdisciplinary study of the mind and the nature of intelligence. Cognitive science is a self-identified academic discipline. It includes different backgrounds, such as philosophy, psychology, artificial intelligence, neuroscience, mathematics, computer science, linguistics, and anthropology. Scholars may come from a wide range of disciplines, but they share a common interest, that of the mind (Wilson & Kiel, 1999). Stillings (1995) defines cognitive science as the "The science of mind".

2.3 Definition of Mind

Minsky (1985) defines mind as the functioning of the brain. Franklin (1995) defines mind as a mechanism of the brain. Minsky says "minds are just what brains do". Franklin (1995) argues that the foundation of exploring a mechanism of mind can be done through the possibility of artificial minds. The implemented artificial minds are man made systems that exhibit behavioural and characteristics of natural living or natural minds. Examples of such artificial minds are briefly discussed in the next chapter.

2.4 Reasons for Studying Artificial minds

Why do we need to study artificial minds? What is the need for studying nonhuman minds such as animals or robots? In "Artificial Minds", Franklin (1995) gave three important reasons for studying artificial minds.

- Questions related to the nature of intelligence in human and nonhuman minds are inherently fascinating. The research on artificial minds may well throw a light on these questions.
- To better understand upcoming man machine mechanisms.
- To build better robots or intelligent machines and to work with them more effectively.

Stillings (1995) also gives some important reasons for simulating human and nonhuman minds in the form of artificial minds.

- Cognitive science theories are complicated and sometimes impossible to understand without simulating and observing in software.
- Comparing people with different capabilities and their cognitive processes via simulation. These different cognitive capabilities are applied on arts and science to give rise to diverse practical applications.

2.5 Artificial Life

Artificial life aims to understand the essential and general principles or properties of natural living systems by synthesizing life like behaviours in software, hardware and biochemical systems. The artificial life area overlaps with cognitive science, artificial intelligence and machine learning. Artificial life originated with Neumann (1946). He designed computational universal living systems for understanding living properties like self reproduction and complex adaptive structures. Artificial life's can be classified into three different synthetic methods (Bedau, 2003).

- Soft artificial lives:-This creates software in the form of purely digital constructions that exhibits a life like behaviours.
- 'Hard artificial life's':- This creates a hardware implementations to exhibit a life like behaviours.
- 'Wet artificial life's':-These are synthesized living systems and use fluid biochemical substances.

The behaviours of a life can be analysed using many different metrics. The major metrics are metabolic activity, competition and social interaction (Bedau, 2003).The conversion from a life to artificial system can be done in three stages

- Understanding fundamental properties of the living systems.
- Simulating a basic unicellular organism and their entire life cycle, and
- Finally, designing the rules and symbols for governing behaviour by interacting with an environment.

The mind can be considered to demonstrate the principles and emergent intelligence associated with artificial life. The society of agents approach exhibits a 'Swarm Intelligence'. The swarm intelligence can be described using mathematical theorems, and based on a group of agents working towards to achieve a common goal (Bedau, 2003). Economic theory can be applied to artificial life in order to analyse and model adaptive or intelligent behaviours. The money or energy spent in such a way is the utility to be maximized. This follows the economic concepts such as price (cost) and utility (Bedau, 2003).

2.6 Approaches on Mind

The mind can be considered as a dynamic structure of asynchronous data, knowledge execution systems, and rich information control states.

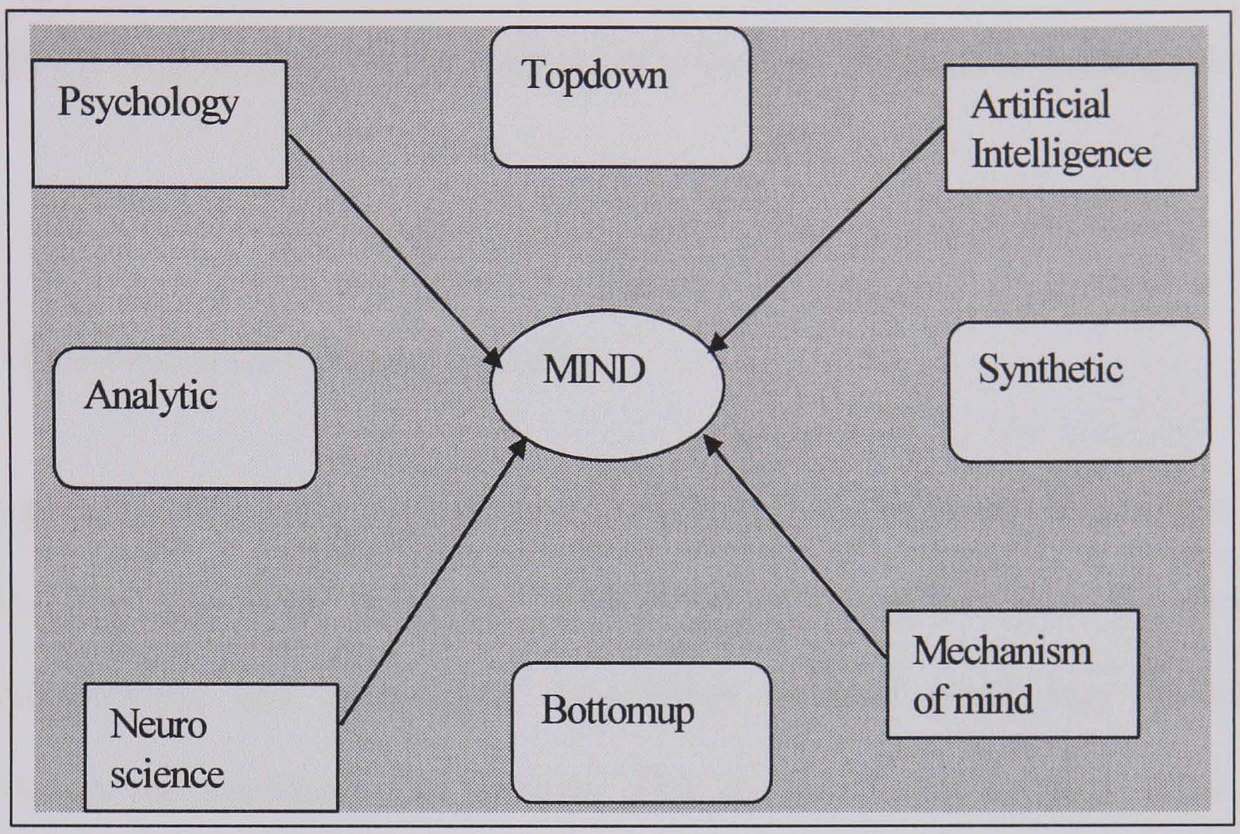


Figure 2.1 Perspectives of Mind.

According to Franklin (1995), the mind can be viewed in four different ways: top-down; bottom-up; analytic; and synthetic. Franklin suggested theses approaches can be based on four disciplines, as given in Figure 2.1. The four approaches towards the study of the mind are

- First, psychology considers a top-down analytic approach and tries to understand existing minds.
- Secondly, Artificial Intelligence considers a top-down synthetic approach and tries to build a mind based on synthetic mechanism.
- Thirdly, neuroscience considers a bottom-up analytic approach and tries to build a mind based on the activity of a neuron or group of neurons.

- Fourthly, the mechanism of the mind considers a bottom-up synthetic approach and tries to build a mind based on the mechanisms that represent properties of artificial agents.

Franklin approaches clearly divides the study of the mind into four perspectives. Franklin (1995) suggests that, the final approach is the best approach to build artificial minds.

2.7 Principles of Natural Minds

Animal cognition is defined as the mental process, or activity, or mental capabilities of an animal. This has been developed from different disciplines like ethnology, behavioural ecology, and evolutionary psychology. Animal psychology includes experiments on the intelligence of animals. This is one of the simplest ways of exploring the complex behaviour of human beings. Most cognitive scientists are interested in comparing human cognition with machine cognitions, only few are interested in animal cognition (McFarland, 1993; Bosser, 1993; Berger, 1980).

The common biological origin of animal and human cognition suggests that there is a great resemblance in animal and human cognition, rather than the resemblance between machine and human cognition. Animal cognition is similar to human cognition, and follows, more or less, human cognitive psychology. According to Berger (1980), animals are both like and unlike humans. Children sometimes behave like animals, through their reflexive behaviours way. Examples include feeding and training children, and so on.

The behaviour of an animal has consequences which depend on situation, energy use and other physiological commodities such as water, weather etc. The important consequence of behaviour is energy expenditure. Such expenditure must be taken into account, because it influences the animal state. According to Thorndike (1911), the behaviour of animals is predictable and follows the uniformity of nature. He says that "any mind will produce the same effect, when it is in the same situation." Similarly, an animal produces the same response, and if the same response is produced on two occasions, then the animal behaviour for that response must changes. The law of instinct or original behaviour is that an animal in any situation, apart from learning, responds by its inherited nature.

Animal behaviour is not simply a matter of cognition; rather it is product of the behavioural capacity and the environmental circumstances (McFarland, 1993; Bosser, 1993).Charles Darwin in his book "Descent of Man" (1871), argued that animals possess some power of reasoning. This research is concerned with the principles whereby an animal is competent for its resources, and so demonstrates intelligent behaviour (McFarland, 1993; Bosser, 1993).

2.8 Rational Behavior in Animals

Theories of rational behaviour are commonly use metaphors from the disciplines of economics, statistics and cognitive science. These theories mainly focus on state and action of an animal under certain circumstances. There are four basic requirements for rational behaviour (McFarland, 1993; Bosser, 1993).

2.8.1 Incompatibility

There are some tasks an animal or person or robot cannot perform simultaneously. For example a robot can not move up and down or left and right at the same time.

2.8.2 Common currency

If the robot or animal cannot perform two activities simultaneously, they must choose among available resources on the basis of some index or potential to optimise their performance. The potential can be measured across different activities using the same unit or currency.

2.8.3 Consistency

A person, robot or animal makes a particular choice when it is in the particular state. It will make the same choice when it is next in the same state. This follows the assumption that a choice from a set of incompatible activities of a robot a person or animals are uniquely determined by the state.

2.8.4 Transitivity

A robot chooses among potential activities on the basis of some common currency. The robot can choose potential activity A over potential activity B (A > B) and potential activity B over C (B > C), how will it choose between A and C? If the robot chooses A>B>C it is said to be transitive. If the robot chooses A > B, B > C and C > A then its choices would be intransitive (McFarland, 1993; Bosser, 1993).The rational agent searches for feasible action. If there is no feasible action is found, then it chooses which is preferable. A rational agent consistently makes the same choice when in the same state and when given the same set of options. The rational decision maker maximizes a quantity, usually called utility (McFarland, 1993; Bosser, 1993).

2.9 Principles of Minds.

2.9.1 Optimal Behaviours in Artificial Minds

Animal behaviour is a trade off between the native courses of action, i.e. physiological, and goal oriented behaviour. Animal is engaged with activities to optimize its pattern of behaviour with respect to the use of energy and time. If the conditions are relevant to two or more activities simultaneously, it chooses the most optimal action among them in terms of its innate and learnt decision boundaries. The mechanisms of designing a machine are different from the animal's kingdom, but the principles remain the same (McFarland, 1993; Bosser, 1993).

2.9.2 Goal directed Behaviour in artificial minds

As shown in Figure 2.2, goal directed behaviour in artificial minds (a human, animal or machine) involves representation of the goal to be achieved. This means that behaviour can be actively controlled by internally represented states. Goal directed behaviour aims to minimize the difference between the "desired" state of affairs and the actual state of affairs. This difference is called as error in the behaviour. This can be corrected by using different factors. The design of an animal is genetically based and product of natural selection. But the robot is based on human engineering principles. However, the principles of their function and goal achievement can be similar (McFarland, 1993; Bosser, 1993).

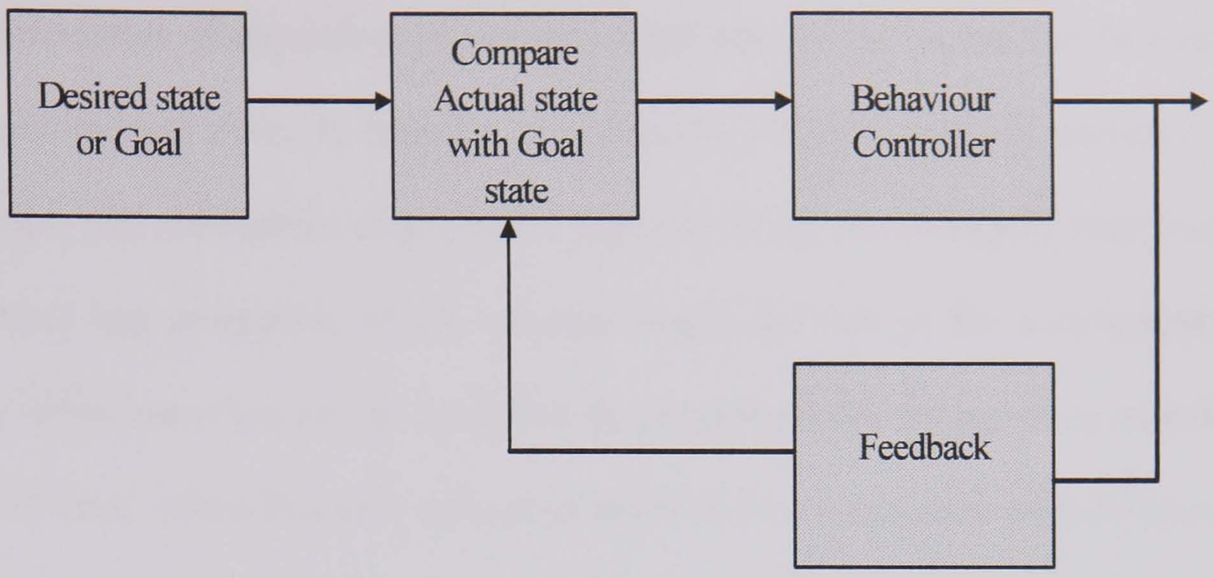


Figure 2.2 Goal directed Behaviour

2.9.3 Cost of Behaviour

The decision making level in animals can be defined in terms of cost functions and utility behaviours - the microeconomic level. Cost functions and utility behaviour in animals operate in such a way that a utility (for example, energy) is maximized or minimized. Let us consider an example as brick laying robot. Initially the robot has stored some sort of energy. The building a bricks is an energy consuming process. The robot monitors its energy level and recharges its energy level when low. This principle relies on some boundary condition and the same is true for animals. The boundary or hunger condition can be varied and sometimes the variable must be nearer the risk of death. It is dangerous to allow hunger condition level if the food supply is not guaranteed (McFarland, 1993; Bosser, 1993). There are three aspects for calculating a cost.

Cost of being in a particular state,

Cost to performing an activity;

Cost of changing the activity.

The combination of physiological and perceptual state of the animal can be represented as a motivational state. It includes the animal's activities and the animal's present behaviour. The motivation of an animal depends on the physiological state (ecological properties) and perception of the external world, as well as the consequence of its current behaviour. Cost can be measured by considering the fitness of an animal over a period of time, where fitness is defined in terms of future expected reproductive success after this period. The cost function deals with real risks, real costs and the benefits. The utility function is the inverse function of the goal function in ethology. Animal behaviour is rational and behaves optimally with respect to this utility.

2.9.4 Decision Variables

A decision-making of a person, animal or robot can be described as an activity whereby decision variables are compared to decision boundaries. From the economic point of view, the decision-making unit is the cost or performance. Decision-making with respect to use of a cost and utility function depends on given thresholds, decision variables and decision boundaries.

Cognitive modelling designs implementation mainly based on the analogies between animals and products. The product may be food, benefit (goal) and physiological aspect. We can also analyse life cycle of the product and life cycle of the animal. A decision of a robot, a person or animal is simply the process by which the decision variables are changed.

2.10 Animal and Finite-state automata

A finite state automation behaves like a simple mathematical animal, that can be regarded as a discrete- time system with finite input and output sets. This responds to only a finite number of different stimuli (the input set or alphabet) and output alphabets. Automata theory is applied in mathematics and computer science. Automata theory applied to animal behaviour uses both neural networks and learning machines (McFarland, 1993; Bosser, 1993).

2.11 Learning in Animals

Learning is a part of development. It is a result of adaptation to accidental or uncertain circumstance. When an animal learns environmental situations, it undergoes permanent change. We expect that learning should, in general, bring beneficial results. Animal learning is similar to reinforcement learning in machine learning or robotics.

2.12 Summary

This chapter introduces the concept of mind and its relationship with the field of cognitive science. This chapter gives reasons for the simulation and the different approaches and principles of artificial minds. Several researchers give different definitions for mind. For example, Minsky argues that "minds are just what brains do", or functioning of the brain. Franklin (1995) argues that mind is a mechanism of the brain. The principles of artificial minds such as cost function, goal directed behaviour, decision making, and learning are introduced.

McFarland (1993) and Bosser (1993) argue that behaviour in animals can be described in terms of cost function and utility. Money or energy is spent in such a way that utility can be maximized or minimized. The animals cost function and utility behaviour can also be described at the microeconomic level. They argue that animal behaviour is a trade-off between the native courses of action. The animals will engage with the activities to optimize its pattern of behaviour with respect to the use of energy and time. The decision-making action always follows the microeconomic level of cost and utility function.

Chapter 3 Cognitive Architectures

3.1 Introduction

Cognitive architecture refers to the design and organization of mind, and provides the means for the integration of cognitive abilities (Langley, 1994). Young (2001) defines a cognitive architecture as an embodiment of the scientific hypothesis of human and nonhuman cognition. Different types of cognitive architectures can be designed, implemented and applied to various tasks. Cognitive architectures are designed to be capable of performing certain behaviours and functions based on our understanding of human and nonhuman minds. Important issues in developing cognitive architectures include task effectiveness, goal achievement, and the ability to perform well in novel situations.

The evaluation of cognitive architectures has always been challenging. Several common concepts and different methodologies have been applied on developing new architectures. There are many examples of developed cognitive architectures developed for different purpose. Different cognitive architectures and paradigms can be said to be modelling different aspects of cognition, different aims, with different metaphors, and from different contexts. To develop a better and sophisticated cognitive architecture, researchers need to understand: (1) the sufficient description of theoretical, design and implementation level of different architectures and; (2) the missing, common and generalised factors of relevant cognitive architectures.

This chapter discusses case study on different types of cognitive architectures, with a specific focus on developing and extending a new cognitive model, for simulating artificial minds using principles from the study of animal cognition. Those relevant to this research include general overview of cognitive architecture (Neumann, 1946), ACT-R (Anderson, 1993), SOAR (Newell, 1990), CRIBB (Bartsch and Wellman, 1989), EM-ONE (Singh, 2005), CogAff (Sloman, 2001) and CAMAL (Davis, 2002). The newly developing Society of Mind Cognitive Architecture (SMCA) extends the CAMAL cognitive architecture with extra processing layers and a distributed model of mind.

3.2 General Overview of Cognitive architecture

Cognitive architectures are young branch of science, but some of the architectures and theories are old. For example the ancient Greeks and philosophers, like Aristotle in the 5th century BC, invented syllogistic logic, the first formal deductive reasoning system (Buchanan, 2002). Basically, theories of cognitive architectures are divided into two classes. (1) First class, the computer is used to model a class of cognitive architecture similar to and inspired by the structure of human knowledge; (2) Second class of cognitive architecture is based on information processing theory. The information can be processed in a sequence of stages from an input to encoding the memory, storage, retrieval, and output.

A general overview of the basic Neumann cognitive architecture is shown in the Figure 3.1 (Neumann, 1946). The first layer receives incoming signals as physical signals through perception. The storage of data, controlling, retrieving, and processing is performed in the next layer. Knowledge can be represented, for example, through the use of production systems using well defined principles for knowledge description. Production systems consist of a set of rules and special class of rule-based system, whose architecture is restricted to fit assumptions about mental structure. In GPS (General Problem Solver) (Newell, 1957) production systems are used to represent information in memory and reasoning strategies are used. Production systems are a fundamental concept for representation in much cognitive architecture. The regulatory system works as controller for the application of these set of rules. The processed data can be stored in working memory, and passed into permanent memory.

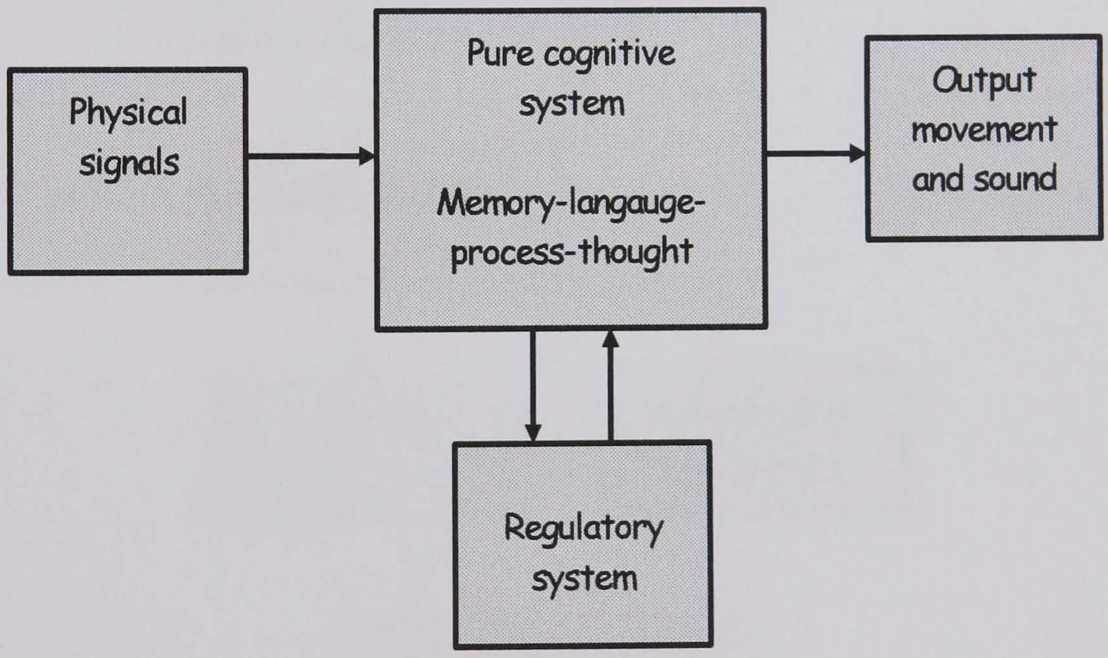


Figure 3.1 Overview of a Cognitive Architecture.

The computational theory of mind claims that mind is like computer, and some functional equivalence to Turing machines. Pylyshyn (1999) suggested that, the task of cognitive science is to give details about mental computation mechanisms, and to determine the kind of computer human mind belongs. The cognitive science determines the cognitive architecture of mind. Cognitive architectures are analogous to computer architectures, and have similar parts.

3.3 Adaptive Control of Thought Rational (ACT-R)

ACT-R stands for Adaptive Control of Thought Rational, or alternatively Atomic Components of Thought – Rational (Anderson, 1976; Ritter and Shiskowskia, 2003). ACT-R is a combined product of Anderson(1976), Bower's existing model of declarative memory(Anderson and Bower, 1973), and production system based on Newell's model (1973b) .The ACT-R theory was presented in 1983 Anderson's "The Architecture of Cognition".

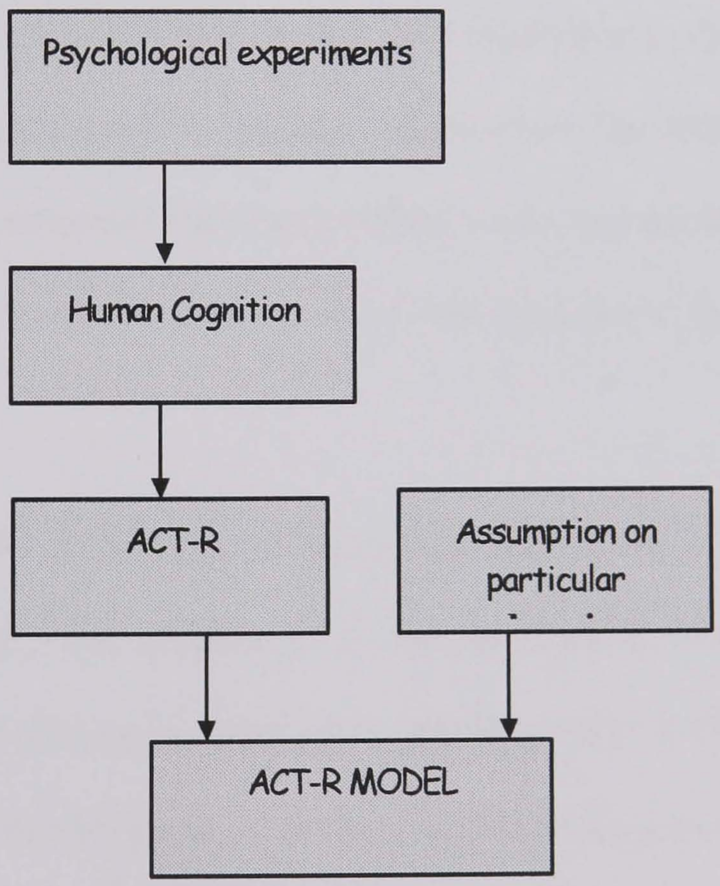


Figure 3.2 ACT- R Structure

Researchers working on ACT-R strive to understand how people organize knowledge, represent and produce intelligent behaviour. This is called as Human-Computer

Interface (HCI). The embodiment of human cognition factors are modelled for Human-Computer Interface. HCI[1] includes the human behaviour, and the hidden states behind the result. HCI combines the advanced work with perceptual recognition, machine learning, affective computing, computational modelling, etc. Many researchers in the area of HCI are working on cognitive architectures. For example, ACT-R and SOAR are the cognitive architectures which are used to implement more dynamic and complex HCI problems (Duric and Gray, 2002). HCI is not just concern with 'ed' designing regular interfaces. Some applications require interfaces which give a virtual human feel on interacting with those machines. This implies that interfaces must exhibit the intelligence, which is built into the applications. For example, an airline company wants to employ a pilot, to test his ability. He can not use a real aircraft. The poor performance can result into fatal accidents and it is also very expensive to use real aircraft. In this scenario, the pilot is trained with a simulated application. The designed interface gives a feel of real world environment. Here the interface works like an experienced pilot, and it generates situations through simulations where the pilot has to make decisions (Byrne, 2005).

ACT-R is designed for modelling individual experiments. ACT-R is the most popular cognitive architecture. The popularity is due to ACT-R being theoretically well grounded, and this is allowed researchers to produce various cognitive phenomena's. ACT –R (Figure 3.2) model can be developed based on the results of experiments done by psychologists on human cognition.

---

[1]Here HCI means Human Computer Interface and not the more common meaning of Human Computer Interaction

ACT -R has a complex cognition structure. A fundamental characteristic of ACT-R is that uses a production system theory. Production system theory uses a production rules for representing human knowledge. The basic premise of ACT-R is a cognitive task. The cognitive tasks are achieved by combining production rules and applying on memory (Anderson, 1993; Budiu, 1998).There are two different categories of long-term memories: (1) procedural and (2) declarative memory. The procedural memory stored with human knowledge. For example, the knowledge of swimming, "2 * 3 = 6", drive a car, speak English, etc. The declarative knowledge is represented in the ACT-R units called chunks. The chunk encodes the information. Each chunk consists of several slots or variables.

The ACT-R designers suggested that, average slot should consist of three to four slots, and one must have an ISA slot. The ISA slot determines the ontological type of a chunk token corresponds. For example, john841 is a person's chunk. Miller's "magic number" argues that chunk should not have more than seven slots. It can be plus or minus two slots (Miller, 1956). Chunks have primarily two sources: (1) perceived objects in the environment; and (2) recorded solutions for the previously solved problems. The production rules specify when and how to retrieve the chunks to solve a problems. The ISA value of chunk plays an important role, such as matching the conditions in the production. The new production rules can be generating from chunks in memory called production compilation processes. The chunks are represented as follows [Chunk-Name: ISA Chunk type; slot1-label slot1-value; etc]. For example, Fact 2 * 3 = 6, knowledge can be encoded as follows [Fact 2 * 3; multiplication-fact; multiplicand1 two; multiplicand2 three; product six].

ACT-R is applied on wide range of human cognitive tasks including knowledge compilation, problem solving, education, controlling of perceptions, etc. For example, solving Hanoi puzzle (tower of Hanoi), learning by analogy, language comprehension, aircraft controlling, etc. ACT-R system controls a mobile robot that interacts with human in building an environments, and obstacles. ACT-R allows much simpler programs for rapid construction of intelligent systems (Trafton, 2005). There are more than five hundred different scientific publications are published on above applications. ACT-R has undergone different versions. ACT-R 4.0 is the first version, ACT-R 5.0 and ACT-R/PM (Byrne and Anderson, 2001) are the next versions. As this research continues, ACT-R evolves closer to the system that can perform a full range of human tasks like memory, learning, problem solving, and decision making. Finally capturing in great detail how we perceive, think about, and act on the world(Anderson, 1996; Anderson, John, Just, Carpenter, Kieras and Meyer, 1995; Rutledge-Taylor, 2004).

Despite this, ACT-R has some errors and limitations: (1) ACT-R need to update millions of chunks for each execution cycle or fire (chunk problem); and (2) ACT-R productions fire serial, and it requires minimum fifty milliseconds for each fire. So, it takes hour's together to complete the firing, and requires thousands of productions to its knowledge base (execution delay). ACT-R has certain limitations (Hochstein, 2002): (1) the basic ACT-R model is not very much applicable to HCI, but enhancements like ACT-R/PM addresses this issues. (2) ACT-R is not sophisticated for larger problems; it is only useful for small set of applications (Hochstein (2002).

3.4 State Operator And Result (SOAR)

SOAR cognitive architecture is based on the computational theory of human cognition. This architecture follows the multiple constraints and computational theories of mind (Anderson and Lebiere, 1998; Newell, 1990). SOAR was introduced by Allen Newell and two of his graduate students, John Laird and Paul Rosenbloom, in 1980 (Newell, 1957). SOAR addresses sufficient description of theoretical, design, functional and implementation issues of human minds. SOAR is well organized to produce a general intelligence. SOAR is considered as a symbolic artificial intelligence mechanism for understanding and simulating human mind. SOAR uses physical symbol system hypothesis for the best way to implement flexible and intelligent behaviours, by manipulating and composing symbols. In parallel to ACT-R, Newell developed SOAR to handle full range of human capabilities. The views of SOAR cognition are tied with psychological theory, and it is expressed in Newell's Unified Theory of Cognition (Newell, 1990). SOAR as a cognitive architecture specify fixed set of a process, memory and control structure. SOAR cognitive model exhibits flexible and goal driven behaviour. The knowledge of the model is continuously enhanced by learning.

SOAR cognitive architecture is an example of software as theorem for models of mind unification. In this theorem, cognitive capabilities like (1) stimulus and response (perception); (2) memory and learning; (3) problem solving; and (4) language capabilities are unified. Newell (1990) defines "Unification" as a programmed extension of a single piece of "software architecture as theorem". A unified theory will unify the existing understandings of cognitions. There are different reasons for unifying the theories of cognition. Combining a few cognitions through unification may leads to required behaviour. Unification is an aim of the science to demonstrate cognitive capabilities. The major areas covered by unified theory of cognition are given below: perception, memory, problem solving, decision making, routine action, learning, language, motivations, emotion, motor behaviour, imaging, and dreaming (Newell, 1990).

As Figure 3.3 depicts, SOAR is classified into two types of memory: (1) long term memory or procedural memory or production memory; and (2) short term or declarative memory. SOAR's long term memory is based on production system theory, where the productions are stored. SOAR's knowledge of the environment is symbolically encoded in production rules, and stored in a production memory.

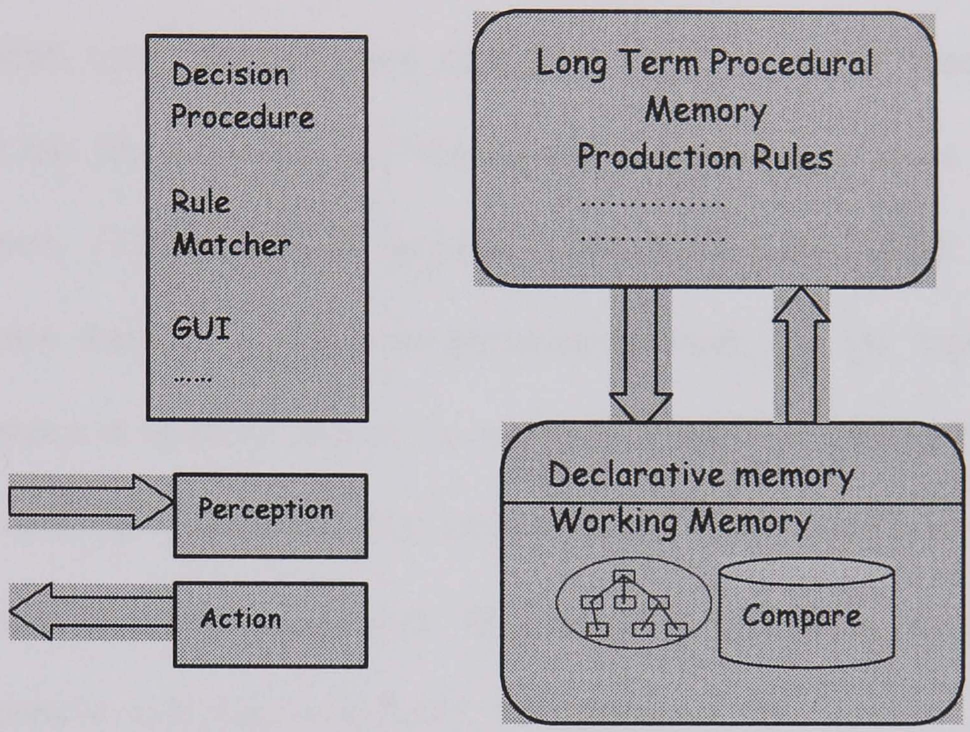


Figure 3.3 Production Rules in SOAR

The productions are in the form of IF <condition> THEN <action> format. These are fine grained and independent in nature. The working memory is loaded with initial state and the operators that are desired for the current state. The control process updates the content of the working memory by firing and matching a production from the search space by using the appropriate operator. This takes the working memory to the next state where the next matching productions are fired and this process continues until the goal state is reached or there are no matching productions available in the search space. The mapping of a production in SOAR model takes approximately 10ms. The SOAR's productions have a set of conditions, which are patterns for matching working memory, and actions to perform when production fires. The actions are elements of working memory. The elements are a set of attribute value pairs of an object. The object has goal, state, operator and problem space. SOAR has a two phases decision cycle.

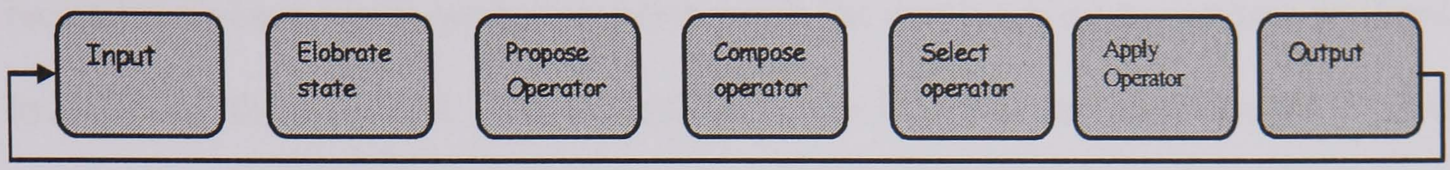


Figure 3.4 SOAR's Decision Cycle

SOAR's decision cycle impasses are repeated until the goal is reached. SOAR's decision phase has four impasses: (1) "tie impasse", where two or more elements have equal preference; (2) "no-change impasse", where no rules match; (3) "conflict impasses", where there is two or more preferences claim; and (4) "reject impasses", where a preference in working memory is rejected. These four impasses are done with SOAR's execution cycle. The execution cycle (Figure 3.4) consists of seven operations for pattern matching: input, elaborate state, propose operators, compare operator, selecting an operator, applying an operator, and output.

For example, to stop a car the operator's are represented as follows :(1) propose operator: If (colour, red) then propose to stop car; and (2) apply operator: If operator proposed to stop car, then stop car. SOAR has a built-in learning mechanism called chunking (Waldrop, 1988). This mechanism is using for to create a new production by storing an output of impasses obtained from the decision cycle. For example, during a decision cycle, SOAR does not understand which operator to select. Then it creates a sub-goal to choose an operator. Once, sub-goal selects the operator, the sub-goal goes away. This goal is stored in a chunk. The next time, if the SOAR faces a same problem, the same stored chunk is executed instead of re-solving a problem.

SOAR adopts the problem space hypothesis to search in right direction that converges to the goal state by adopting the strategies of (i) knowledge-intensive processing and (ii) knowledge-lean processing (Newell, 1990). The process of problem solving is a step by step procedure. For each step an appropriate operator must be selected. In the recognition phase all the productions that match the working memory content are fired by selecting the operators. In each step there may be many operators desired for the current state. In a decision phase the operators are sorted into preferences. Among these preferences the best operator must be selected. This leads to the goal state. Their may be possible that, two operators can be lead to the goal state.

There may not be any operators that can be selected, in which case the system is not defined with what to do next. This state is called as an impasse. Each time an impasse is arising, the SOAR sets a sub goal to overcome the current impasse. The decision process sets another problem space by saving the current content of the working memory. This leads to the cascade of sub goals. SOAR continuously stores new knowledge in a long term memory based on the experienced learning mechanism.

The learning technique adopted in SOAR is called as chunking. Learning adds a new production to the long term memory in an elaboration phase. A chunk is a by-product of an impasse. All the productions are fired. The operators applied to overcome an impasse are stored as a single entity called as chunk. This can be added to the procedural memory by creating an index. Chunking mechanism has two important functional properties:

(1) Firstly, it provides a solution to the knowledge-indexing problem; and (2) secondly, chunking can be applied for all kinds of impasses.

The problem solving activity under impasses are placed in a chunked production. Consider an example of where 'object1' is a large and red colour. This information can be stored in a data chunk. This can be able to recall 'object1' as a large and red colour. The trick for method of learning the recall rule is called as data chunking (Anderson and

Lebiere, 1998; Newell, 1990). The elaboration phase adds special information for the next action. This is called as preference. The preferences allow the architecture to specify an action that could be taken, and action actually taking. The preference always refers to a adopting some particular object in some position in a context stack, and takes one of the following values: (1) acceptable; (2) reject; (3) relative preference; (4) absolute preference and (5) indifferent. SOAR decision cycle impasses are repeated until the goal is reached.

SOAR decision phase has four impasses: (1) "tie impasse", A scenario in which there is a collection of operators, which are desired and can not be discriminated. For example, the scenario of two or more elements having equal preferences; (2) "no-change impasse", a scenario in which there are no acceptable operators among the preferences, which can be applied for the current state. Here no rules match ;(3) "conflict impasses", a situation where the decisions that can be made in the current state are contradicting with each other. There are two or more preferences claim; and (4) "reject impasses", a scenario in which the only preference is to backtrack by rejecting the previously made decision. Preferences in working memory are rejected. These four impasses are done with SOAR execution cycle. These are chosen between multiple rules.

Research into SOAR is still continuing, and researchers are adding new mechanisms (Figure 3.5) for SOAR to solve difficult problems. For example, reinforcement learning. SOAR has successfully mimicked other expert systems, such as Neomycin and Designer (Franklin, 1995), Robo-SOAR (Laird et al., 1989), etc. Some versions of SOAR have been developed within the SOAR architecture including a problem solving sentence processing, etc (Newell, 1990).

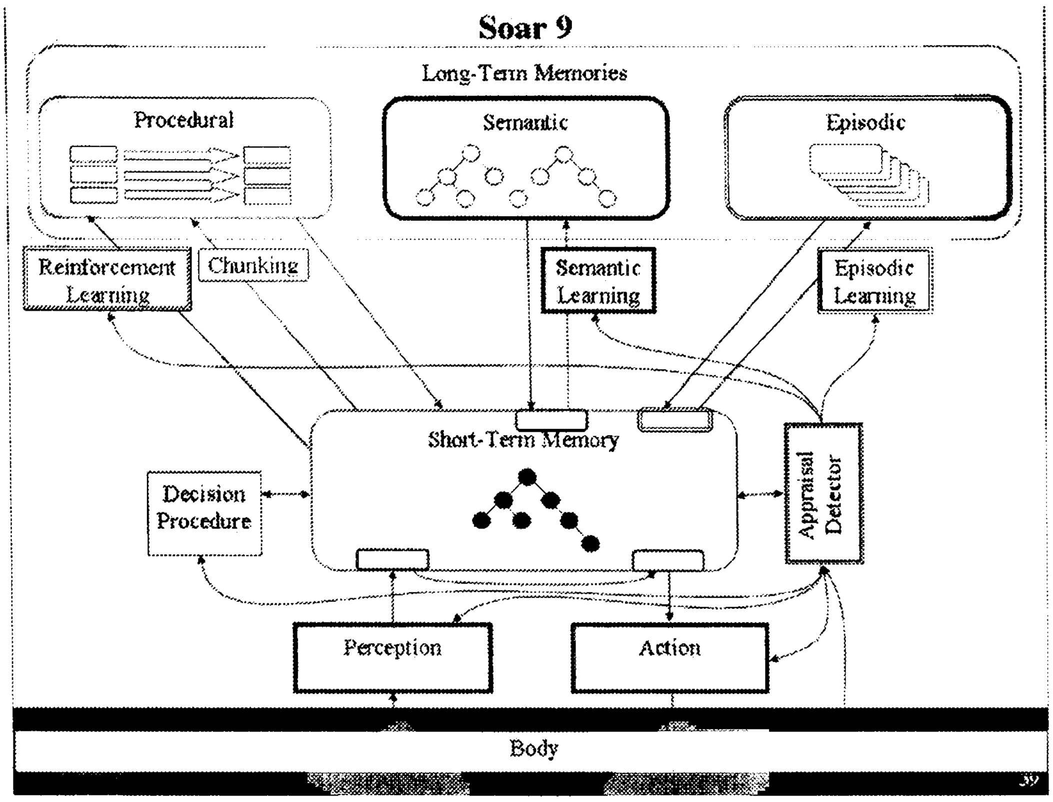


Figure 3.5 SOAR's Recent Version

3.5 CRIBB

The development of competency in reasoning about mental states has been studied intensively in the field of 'theory of mind'. CRIBB architecture (Children Reasoning about Intentions, Beliefs and Behaviour) (Bartsch and Wellman, 1989) is a cognitive model that simulates the knowledge and inference processes of a competent child solving theory of mind problems. The CRIBB model was designed to investigate the belief-desire reasoning model in young children (Wahl and Spada, 2000).

Dennett (1978) explained a story about maxi and his chocolate in front of the group of children's. The boy named maxi puts a chocolate into a blue cup board. Later, his mother puts the chocolate into green cupboard while maxi is not present. When maxi returns, the children's are questioned where maxi can look for a chocolate. The children's group, which are lower than or equal to four years, has more false beliefs.

The age group in between four to six years older children's almost all answered correctly.

The CRIBB architecture Figure 3.6 consists of representations (rectangular boxes), inference schemata (ellipse) and a consistency mechanism. The CRIBB simulation starts by giving propositions as an input. This proposition contains the children information received during the experimenting.

Representations (rectangular boxes in Figure 3.6) fall into two main categories: primary and secondary representations. Primary representations are the system's own beliefs about the situation and, behaviour of person and the others facts about a physical world (Wahl and Spada, 2000). The secondary representations are the systems beliefs about mental states. This includes another person's perceptions, beliefs and intentions. This model states that a person's actions can be explained by his beliefs and desires, where beliefs can be derived from perceptions and previously held beliefs.

CRIBB is given a proposition; a belief is inferred from it. The consistency of the belief is checked with the existing set of beliefs, and if no contradiction is found, then added to the belief set. The consistency mechanism detects and resolves contradictions in the system's belief set. Any beliefs deemed to be false become secondary representations (Davis and Lewis, 2003; Lewis 2004). Beliefs can be derived from perceptions and previously held beliefs. The CRIBB is given a proposition; a belief is inferred from it. The consistency of the belief is checked with the existing set of beliefs (Davis and Lewis, 2003). If no contradiction is found, and then it has added to the belief set. For example, The CRIBB B-D-I mechanism can be seen at the deliberative level. P, B, D and I are named as sets. The minimal "logic" symbol model of CRIBB (Bartsch and wellman 1989; Wahl and Spada, 2000) with reasoning over perceptions (P), beliefs (B), desires (D) and intentions (I) as follows:

$P: = \{r, s, q, p\}$

$B: = \{\neg p\}$

$P \otimes B \rightarrow B'$

$B' := \{r, s, q, p\}$ % the set B' contains the updated perception of p results in the retraction of $\neg$ p.

$B' \otimes D \rightarrow D'$

$D' \otimes I \rightarrow I$

Where B is the existing belief set and P is the perception set, and where the new set B¬contains the system's new belief set with all possible contradictions resolved (Davis and Lewis, 2003). D is the desire set, and the new set D' contains the system's new desire set. I is the intention and I' is the new intention set. The inference schemata (ellipses in Figure 3.6) are based on the belief-desire reasoning scheme, while perception-belief inferences represent knowledge about the relationship between perception and belief. If a person perceives X, then the person believes X. Fact-time and belief-time inferences deal with facts and beliefs along a time scale. If a fact/belief is true at t1, then that fact/belief is also true at t2, unless there is new information.

Wahl and Spada (2000) claimed that CRIBB's theory of mind includes some commonsense-schemata that is essential for solving experimental tasks. For example, a person searching a specific location of object or an item; and the schema represents that object or item can not be in two places. This example needs commonsense to reason.

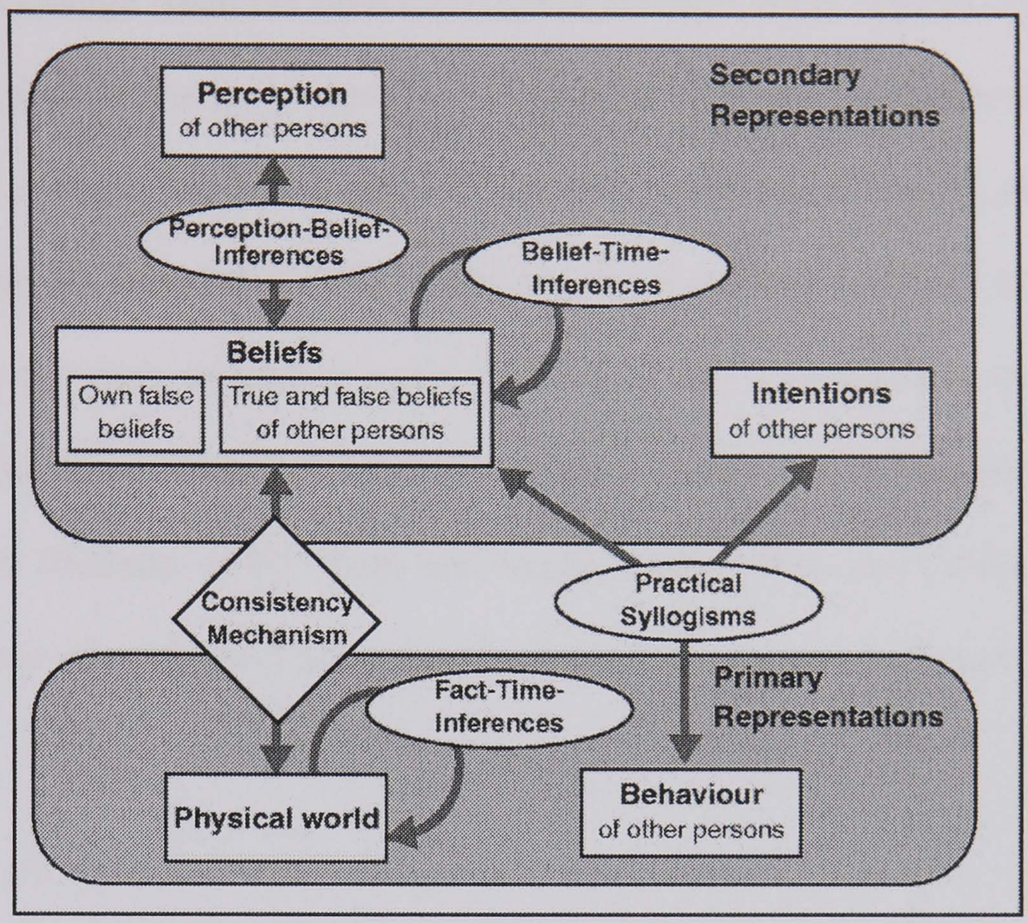


Figure 3.6 CRIBB Architecture

Lewis claims that CRIBB can be extended by performing different models of emotions (Davis and Lewis, 2004). Affective-CRIBB is Lewis (2004) extended CRIBB model. The main purpose of the Affective-CRIBB model is to extend affective capabilities by including emotion as an essential part of CRIBB. According to Lewis (2003); emotion plays a fundamental role in variety of cognitive tasks, such as perception, learning and decision making. Lewis (2003) introduced affective affordances into the perception mechanisms by extending Gibson's (1986) theory of affordances. Gibson (1986), defines affordance is "something that refers to both the environment and the animal in a way to existing term does. It implies that complementary of the animal and the environment".

As defined above Gibson's (1986), affordance is a fact of the environment, and also a fact of the observer in the environment. Gibson, claimed that affordance is something, that does not change when the need of observer changes. If the object is believed to be in the same place at the time interval, A-CRIBB helps to resolve this contradiction by maintaining a consistent belief set. CRIBB will create an affective correspondence (ACorr) value for each belief. This belief may be either true or false. ACorr value is attached for the each originated belief. If there is any contradiction, then it can be compared with the existing belief set. The minimal "logic" symbol model of A-CRIBB (Bartsch and Wellman 1989; Wahl and Spada, 2000; Davis and Lewis, 2004) with reasoning over perceptions (P), beliefs (B), desires (D), intentions (I) and Affect (A) as follows:

$P := \{r, s, q, p\}$

$A: = \{\text{importance (high, p), importance (low, r), importance (low, } \neg p)\}$ % From the Affect model

$B: = \{\neg p\}$

$A \otimes P \rightarrow AP$

$AP = \{p, s, q, r\}$

$AP \otimes B \rightarrow B'$

$B' := \{p, s, q, r\}$

ACorr value used in A-CRIBB model are consist of dynamic values, and that can be increased or decreased according to given belief set either true or false. The affective response is linked to the drives of the system. A-CRIBB model is based on the goal oriented theories of emotion. This incorporates goal base, goal oriented and goal achievement mechanisms. The goal achievement mechanism is a feedback for the sub system of A-CRIBB. A-CRIBB model consists of central monitoring system. This is responsible for communication between the sub-systems, and controlling the semantic messages. According to Lewis (2003), A-CRIBB model has certain limitations: (1) central monitoring system is not complete; and (2) the goal achievement with ACorr value is limited.

3.6 Cognition and Affect architecture (CogAff)

CogAff is a generic cognitive architecture, and introduced by Sloman (2001). The main aim of the cognition and affect architecture is to understand the different types of architectures based on human and nonhuman (minds) mental states, such as intelligent capabilities, moods, emotions, beliefs, thoughts, and desires (Sloman, 2002). The Cognition and Affect project is concerned with understanding mechanisms of emotions, and to fit for cognitive models.

According to Sloman, the 'Intelligence' like 'emotion' is a cluster concept. This varies with cluster of capabilities and no sharp boundaries. For example, animals (perhaps insects) consist of complex capabilities. Sloman argues that insect follows completely complex reactive mechanisms. Similarly an organism like a robot follows reactive or reactive with global alarm systems. Sloman claims that animals, humans and others have deliberative mechanisms. Because their capabilities are enormously rich. They can answer the questions like "what would happen if", "how shall I react know", etc.

The CogAff architecture was designed to provide a framework for describing different kinds of architectures. Sloman mainly classified CogAff architecture into two divisions: (1) CogAff and (2) H-CogAff. The CogAff specifies the broader outer line of variety of organisms or robots or machines functional roles and mechanisms. H-CogAff architecture is specific to human minds. CogAff is an abstract architecture that accepts perceptual information, processes, and action outputs in the environment. It is based on three-column architecture of perception, central processing and action. CogAff architecture (Figure 3.7), consists of three main layers: (1) reactive mechanisms; (2) deliberative reasoning and; (3) meta-management. These layers supports for different classes of emotions found in humans, animals and others.

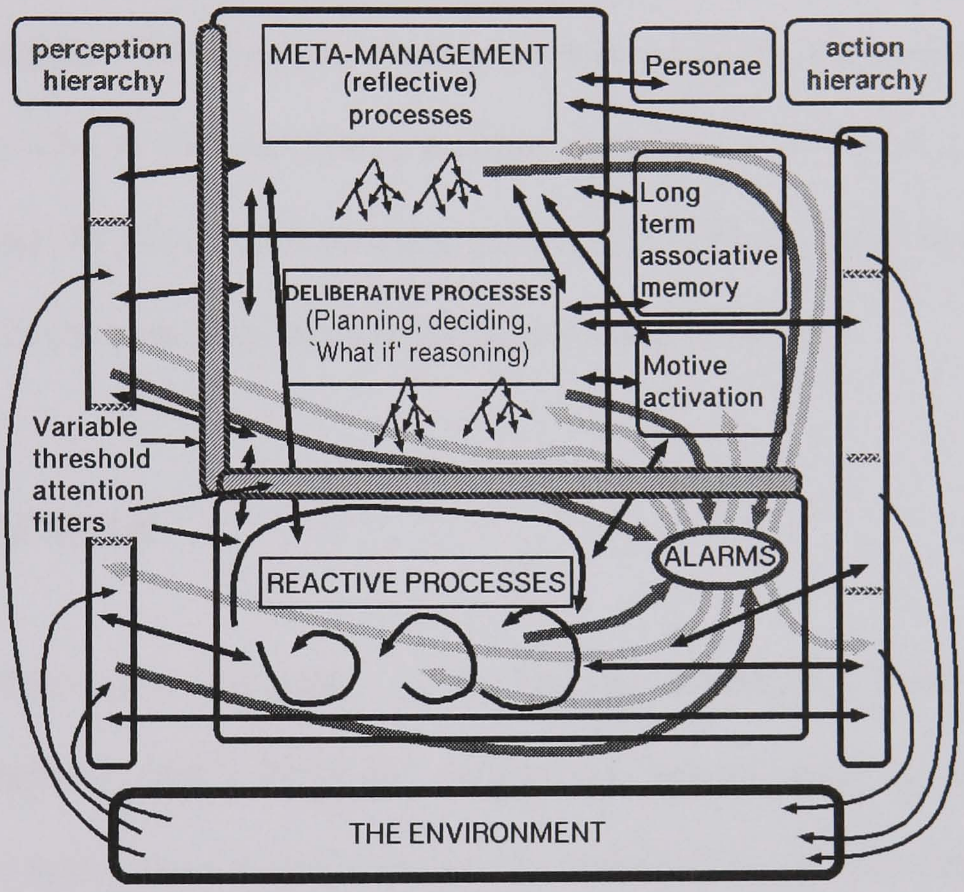


Figure 3.7 CogAff architecture from (Sloman, 2001, 2002).

These layers consist of primary and secondary emotions. The reactive layer detects the objects in environment, executes, and then determines how to react. This layer interacts with the internal, external conditions. Then it produces internal or external state changes. The reactive system is very complex and powerful. This layer needs to store all the mechanisms of particular mind. For example, reactive children like behaviours.

Reactive layer includes a global alarm mechanism, which belongs to primary emotions. The deliberative layer supports for secondary emotions. Secondary emotions are semantically rich emotions.

This layer is responsible for perception, planning, evaluation, allocation of resources, and decision making. This layer can learn the generalizations, and pass to the other layers. The metamanagement layer or reflective layer supervises, and controls the other layers of architecture, more efficiently. Sloman describes that; this layer can support and control the thoughts. For example, human emotions such as infatuation, humiliation, thrill etc. According to Sloman (2001), dividing the layers is left to the researchers. For example, Minsky (2001) divided metamangement layers into separate layers, and Davis (1996) separated reflexes from the reactive layer (Sloman, 2001).

3.7 EM-ONE

EM-ONE architecture originated from Marvin Minsky's "emotion machine" architecture (Minsky, 2002). EM-ONE architecture was proposed by Minsky and his student Singh (2005), from MIT media lab. According to Singh, EM-ONE architecture is an example for its predecessors Minsky and Sloman, and hence he called Minsky-Sloman Architecture. Main goal of EM-ONE cognitive architecture is to support human-level intelligence in systems. According to Singh his architecture refers to the "structure and arrangement of commonsense knowledge and processes".

EM-ONE architecture for commonsense computing, that is capable of reflective reasoning about situations involving physical, social, and mental dimensions. EM-ONE architecture involves complex interactions among the several "actors" along with physical, social, and mental dimensions.

As Figure 3.8 depicts, table building environment is an example for AI architecture, and uses artificial environment called Roboverse. This is simulated world with rigid body physics, and populated by several actors. These actors are guided by EM-ONE cognitive architecture. These actors work together to build a tables and chairs, by using simple and modular components like sticks and boards. Components are looks like small toys, and they can attach one other with their corner and endpoints. The actors are simulated robots, and possess a perceptual system to take physical actions. They are roughly looks like a human like shape, with a single arm. The hands can be turned off and on. These hands will act like magnets by attracting the nearer objects. Singh demonstrated the commonsense mechanism, using as an illustration, the building a table (Figure 3.8) in Roboverse.

Green (left side) wants to build a table; Green watches there is any partly built table to attach more legs to complete a table building. Green moves and grabs a stick, and then moves nearest to the table. Green tries to build a table by using its single arm. It tries to match and attach the table legs, but it fails. Green immediately realizes it needs help. Afterwards, Green calls the Pink. Until that, Pink has been involved with its own project, and has not been paying attention towards Green. The Pink looks over a Green, and realizes Green is trying to disassemble the partially built table. Pink comes nearer, and removes the one of the table leg. Green realizes Pink does not understand the intention of Green, and then complains. Green realizes Pink, did not see Green attaching a stick. Afterwards, Green again tries to attach a stick to the partly built table. This time, the Pink watches the construction. Pink realizes that the Green's intention is to complete a table, rather disassemble the table. Green expects help from the Pink. Green expects Pink to hold a table, so that Green can attach a table leg. Pink holds a table and Green inserts the stick. This mechanism proposes a course of action and intentions of other actors for reflecting upon repairing mistaken errors. It shows the aspects of physical, social, and mental actions.

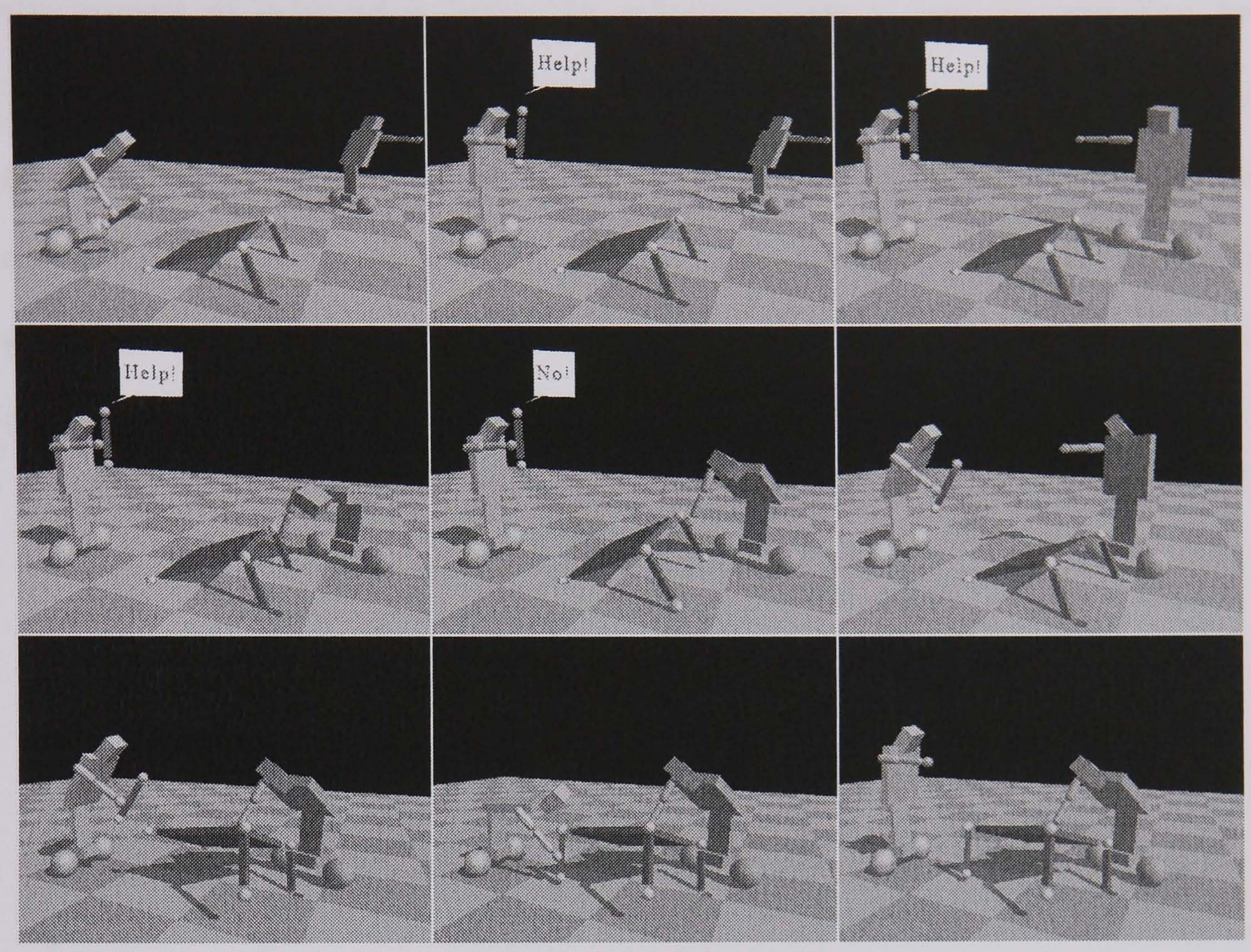


Figure 3.8 Singh's Table Building Mechanism

EM-ONE architecture was proposed and designed for six layers (Figure 3.9), but was implemented for the first three layers. Each of the layers is represented by terms called mental critics. The mental critics are encoded in the form of frame-based knowledge, and support a description of connected actors (two wooden one-armed robots) with actions, situations, events (moving, picking, attaching), objects (table sticks), and their properties.

Singh considered each layer as mental critics: (1) reactive critics; (2) deliberative critics; and (3) reflective critics. The reactive critics interact with the environment. The deliberative critic's reasons about the circumstances, actions and consequences. Deliberative critics interact and coordinate with actors, and objects in the environment from, deliberative actions. For example, knowing the exact positions, picking the right object, connecting the exact position or edge, and so on. The reflective critics asses the effectiveness of deliberative layer. The reflective layer is used for correcting the incorrect predicted actions. The self-reflection, self-conscious, and self-idea critics are self reflection layers. According to Singh, EM-ONE architecture has meta-managerial critics. This has been supported with top-level critics. In, EM-ONE architecture has a great flexibility for critics to activate. The central idea of having critical-sector model is that, when the system encounters a problem, it brings knowledge of reasoning.

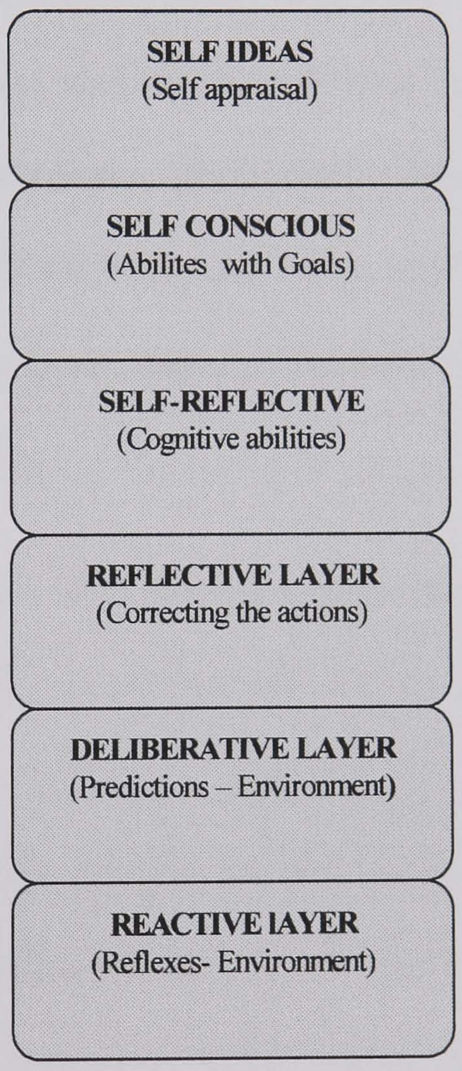


Figure 3.9 Singh's Proposed Six-layers Architecture.

Singh (2005) describes, metacritics are concerned with coordinating the activities of a layers of mental critics. Metacritics are operated at each 'cognitive cycle'. Cognitive cycle is the time between a after sensing the world, and before taking an action. This will decide which subset of mental critics should be active in the present time. Singh states that EM-ONE architecture is a product of his own style of thinking. He had a plan to extend this architecture with meta-metamangement. He argues with his EM-ONE example, metamangement is a higher order of thinking that could be used for guiding deliberation and reflection (Singh, 2005).

3.8 CAMAL

CAMAL (Computational Architectures for Motivation, Affect & Learning) architecture was proposed by Davis (2002, 2004, 2005, and 2006) from the University of Hull. CAMAL is a theoretical framework developed from Guardian (Hayes-Roth, 1993), Cogaff's (Sloman, 2001) three column, three level architecture; and Sense-Think-ACT-Cycles mechanism, CRIBB (Wahl & Spada, 2000), and Singh's EM-ONE (Singh and Minsky, 2005) commonsense-frame based architectures. The purpose of CAMAL is used to simulate artificial minds.

CAMAL cognitive architecture attempting to demonstrate some theoretical; and design issues associated with, linking perception and action through motivation and affect mechanism. CAMAL uses different testbeds and physical environments for demonstration. Presently using testbeds are five-aside football; tile-worlds; fungus world; and physical environments like Robot-CAMAL for control of multiple reactive architectures. Some of the experiments are still under investigations (Davis, 2007).As Figure 3.10 depicts CAMAL has four tier and five column architecture. This provides a basic template for all explanations (Davis, 2007). Cognition tasks involve the control of external and internal behaviour of the environment. The control of behaviour, for further of its goals. Affect mechanism in CAMAL uses BDI models for adaptive decision making across the architecture. BDI (Beliefs, Desires Intentions) are the mental components present in rational agent architectures (Bratman, 1987; Cohen 1990; Rao and Georgeff, 1993).CAMAL (Davis, 2002) uses a logical model of reasoning based on Beliefs, Desires, Intentions that mirrors the motivation and learning. The BDI model intentions are adopted plans or strategies for achieving desires. The adoption of specific plans converts desires into achieved intended desires.

Lewis (Davis and Lewis, 2003) research on Affect-CRIBB distinguished affect as emotion in terms of their magnitude and type. Emotion is a kind of Affect. The emotions are anger, joy, intelligence, etc. A-CRIBB theory affect mechanism uses control states and motivators and affordances.

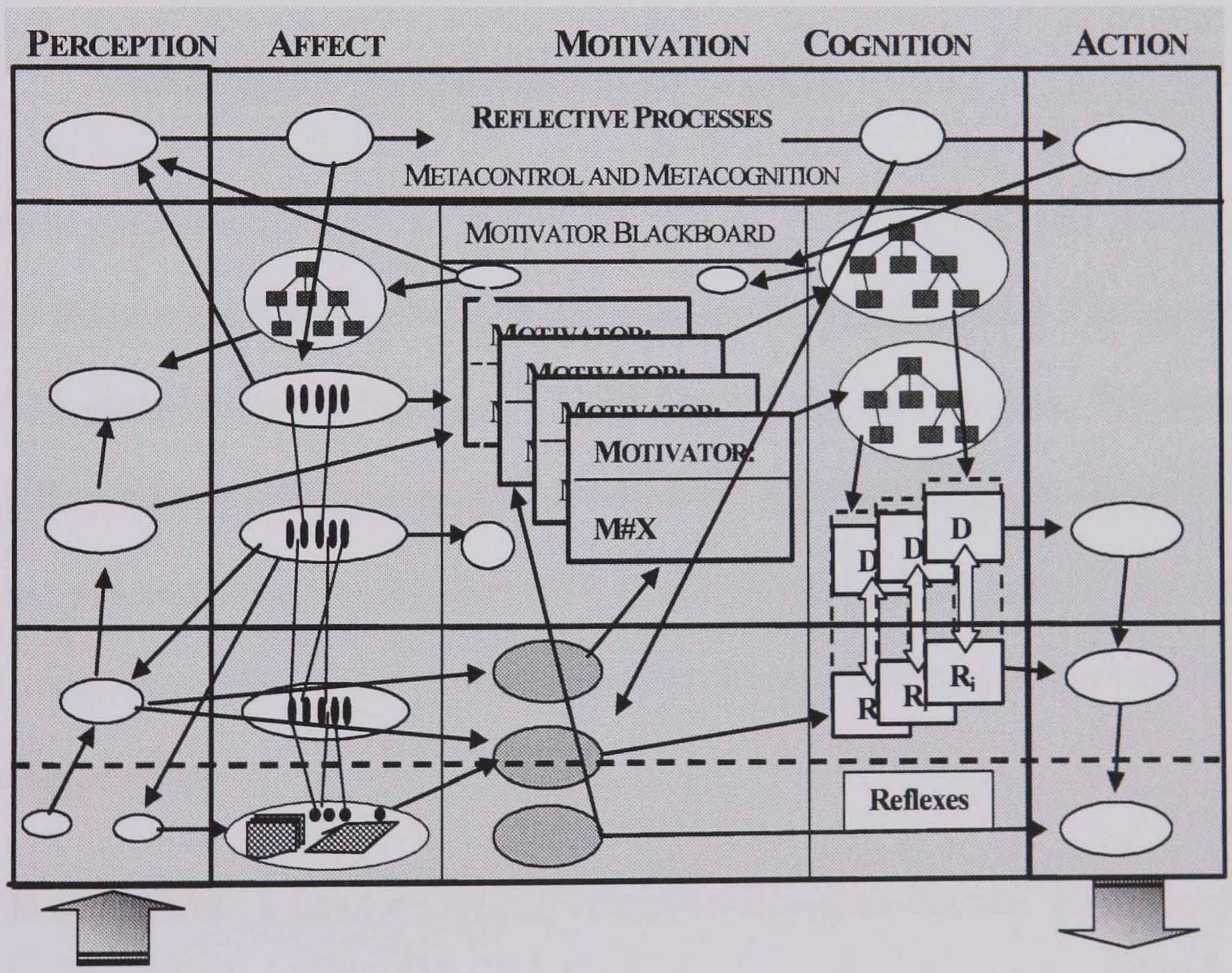


Figure 3.10 CAMAL Architecture.

As discussed in section 3.5, CRIBB (Wahl and Spada, 2000) model uses BDI models for representing reasoning capabilities of five year old child. This model has been extended with an affordance affect to map onto motivational structures (discussed in section 3.5).The CAMAL agent navigates around the environment, recognizes the actors or objects. This follows some of the cognitive capabilities like, perception, problem solving and reasoning. Action means recognizing the object and navigating around the environment (Davies, 2005). CAMAL architecture has explored to adopt affect and learning models over the affect model. This affect magnitude is useful for "fitness function". The investigation of deeper learning capabilities should, in general, bring beneficial results. The CAMAL principles are under investigation, through a satellite project; and a metacontrol and metacognition mechanism on extended CAMAL with extra processing layers, for distributed model of mind. This extended architecture is discussed separately in chapter 6.

3.9 Comparison of cognitive architectures

Cognitive architectures can be assessed in terms of their ability and efficiency to support the construction of models and simulations of cognition tasks. The comparison Table 3.1 given below explains the different types of cognitive models, their purposes, and skills used to develop a cognitive architecture. ACT-R and SOAR are well known and very old cognitive architectures. ACT-R and SOAR are very popular and contains many users. Popularity is due to their flexibility for researchers to expand for different useful applications.

ACT-R and SOAR incorporate aspects of human-like reasoning and specific problem-solving capabilities. ACT-R is an example of a moderately specified architecture, in which one can build such simulation models. There are some features that are important in the study of complex tasks that ACT-R is not well-adapted to model. ACT-R has certain errors in chunks management, and time delay in execution. According Wahl and Spada (2000), the CRIBB can be re-implemented by using general architecture of ACT-

R, and is useful. Some of the theoretical constructs adopted for CRIBB's inference schema is directly correspondence with ACT-R. The operational resources of a child can be expressed with source activation (production rules) in ACT-R.

Cognitive model	Refrence	Mechanism	Purpose
ACT-R	Anderson(1976)	Production systems	To demonstrate and understand human Skills
SOAR	Newell(1980)	Production systems and Chunking mechanisms	learning, reasoning, decision making. (Human level)
EM-ONE	Singh(2005)	Encoded in the form of Frames (6 layers)	Commonsense thinking and Reflective reasoning.
CRIBB	Wahl and Spada (2000)	Belief, Desire, Intentions(BDI)(By using Primary and Secondary representations)	Reasons like a 5 year old child
CogAff	Sloman (2001- Ongoing)	Reactive, Deliberative Reasoning and Meta-management	Generic purpose (Human, animals &machine minds)
A-CRIBB	Davis and Lewis (2004)	Belief, Desire, Intentions(BDI), and Affect(Acorr) value	Updating CRIBB values, through consistency mechanism
CAMAL	Davis (2002 - Ongoing)	Belief, Desire, Intentions(BDI) Models for reasoning	Artificial minds
SMCA	Vijayakumar and Davis (2007)	Metacontrol and Metacognition on BDI models	Artificial minds (Extending CAMAL)

Table 3.1 Cognitive model's Comparison Table

Singh (2005) argues that SOAR addresses orthogonal systems, because SOAR is a rule based system. EM-ONE is built by using rules. Singh (2005), claims that it is not a difficult to implement version of EM-ONE using SOAR as a substrate. In, SOAR "architecture" refers to the minimum set of mechanisms. In EM-ONE, architecture refers to the "structure and arrangement of commonsense knowledge and processes". This discussion, reviewed different researchers views on cognitive architectures aims, representations, principles, working mechanisms, common factors, generalised factors, missing factors, limitations, problems, advantages and disadvantages.

This Chapter gives clear idea for developing new cognitive architecture. The extended CAMAL cognitive architecture with processing layers and distributed model of mind or "Society of Mind" approach to Cognitive Architecture (SMCA) is described and explained in the Chapter six.

3.10 Summary

This chapter summarized models of artificial minds, and different types of cognitive architectures, with a specific focus on one developing cognitive architecture. Cognitive architectures discussed include SOAR (Newell, 1990), ACT-R (Anderson, 1993), CRIBB (Bartsch and Wellman, 1989), EM-ONE (Singh, 2005), CogAff (Sloman, 2001) and CAMAL (Davis, 2002). Cognitive architecture refers to the design and organization of mind, and provides the means for the integration of cognitive abilities. Cognitive architectures are designed to be capable of performing certain behaviours, and functions based on our understanding of human and non human minds. Important issues in developing cognitive architectures include task effectiveness, goal achievement, and the ability to perform well in novel situations. ACT-R and SOAR incorporate aspects of humans, like reasoning and specific problem-solving capabilities. CogAff is a generic architecture that includes the different levels of abstraction in human and nonhuman minds. The CRIBB model was designed to investigate the belief-desire reasoning model in young children. The purpose of CAMAL is used to simulate artificial minds like animals and other robots. Singh and Minsky's EM-ONE architecture used for commonsense computing.

This discussion section assessed SOAR (Newell, 1990), ACT-R (Anderson, 1993), CRIBB (Bartsch and Wellman, 1989), EM-ONE (Singh, 2005), CogAff (Sloman, 2001) and CAMAL (Davis, 2002) cognitive models, in terms of their ability, and efficiency to support the construction of models and simulations of cognition tasks.

Chapter 4 Metacognition

4.1 Introduction

"The Society of Mind is more than just collection of theorems. It is a powerful catalyst for Thinking about Thinking" (Singh, 2003).

Metacognition is a relatively new buzz word in cognitive theory. The study of metacognition has grown since the 1970s, in educational psychology. The metacognition concept provides a powerful tool towards developing efficient and quality computational models. Metacognition is often simply defined as "thinking about thinking" (Wilson & Keil, 1999). Metacognition is any knowledge or cognitive process that refers to monitoring and controlling any aspect of cognition. Adkins (2004) defines "metacognition is thinking about knowing, learning about thinking, control of learning, knowing about knowing and thinking about thinking". Minsky (1985) defines "we cannot think about thinking, without thinking about thinking about something". The metacognitive act can be referred to as metacontrol. Metacognition can be viewed in two ways: (1) monitoring a group of agents in an intelligent or cognitive or robotic architecture (i.e. self reflection) and; (2) making changes by adapting effective strategies in the group of agents.

4.2 Elements of Metacognition

From one perspective, there are four elements to metacognition: (Wilson & Keil, 1999; Adkins, 2004; Zalta, 2005; Flavel, 1979) (1) metamemory; (2) metacomprension; (3) metamanagement and (4) schema training. Metamemory is used for storing the information about a cognitive task. This helps for recalling information in the execution of cognitive tasks. Flavell and Wellman in 1977 proposed a theory of metamemory to explain young children's development and application of recall strategies. The young children's failure to apply strategies for recalling information because of their lack of awareness of "parameters that govern effective recall". Metacomphrension is used for detecting and rectifying the errors. This helps to improve the performance. Research on children shows that young learners have a lack of metacomprehension strategies, and limited opportunities to develop skills. Hence they need remedial action. Self regulation or metamanagement layer works for rectifying the errors in cognitive tasks, and thoughts are adjusted by giving feedback. The students make corrections or self reflection based on trail and error methods. Schema training is a meaningful learning for generating own cognitive structures or frameworks (Cox, 2004).

4.3 Object-level and Meta-levels

Cognitive processes can be split into an object-level and a meta-level. The meta-level contains information for controlling the object level. The information flow from object level to meta-level is referred to as the monitoring processes. Similarly, flow of information from meta-level to object-level is referred to as control processes.

As depicted Figure 4.1, meta-level or metacognition includes the selection of the processes, allocation of study time, termination of processes, selection of processes, execution and memory details etc. Metacognition can be constructed based on problem solving (e.g. planning) and metacomphrension (e.g. story understanding) processes of the object level (Cox 1994, 1995, 2005). "Meta-level is widely using in the reflective programming.

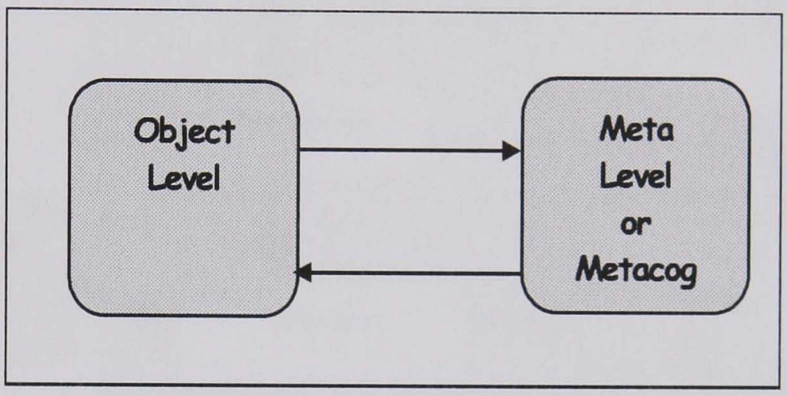


Figure 4.1 Cognition to Metacognition

Self-monitoring meta-level is a component, and it controls and monitors the object level. This also changes its behaviour if necessary. Meta-level is also called as self monitoring layer. This layer inspects and modifies the self-monitoring meta-level. If M is a meta-level and P is its object-level then relationship as follows: (1) M is a program that interprets P, and it takes P as an argument; (2) P calls M to monitor, inspect, modify, correct or improve the P. This is called as reflective" (Kennedy, 2003).On the other hand, Cox (1994, 1995, and 2005) and Kennedy (2003) argue their similar views on meta-level, metacognition and reflective concepts. Kennedy attempted to differentiate metacognition and reflective terms. Similarly, Davis and Buchanan (1977) argues that metareasoing and reasoning raises similar confusion between cognition and metacognition. Metareasoning is reasoning about the reasoning. The programs need to reason about the functioning part of cognitive processes, and each level of program execution (Cox 1994, 1995, 2005).

4.4 Execution engine & Expertise model

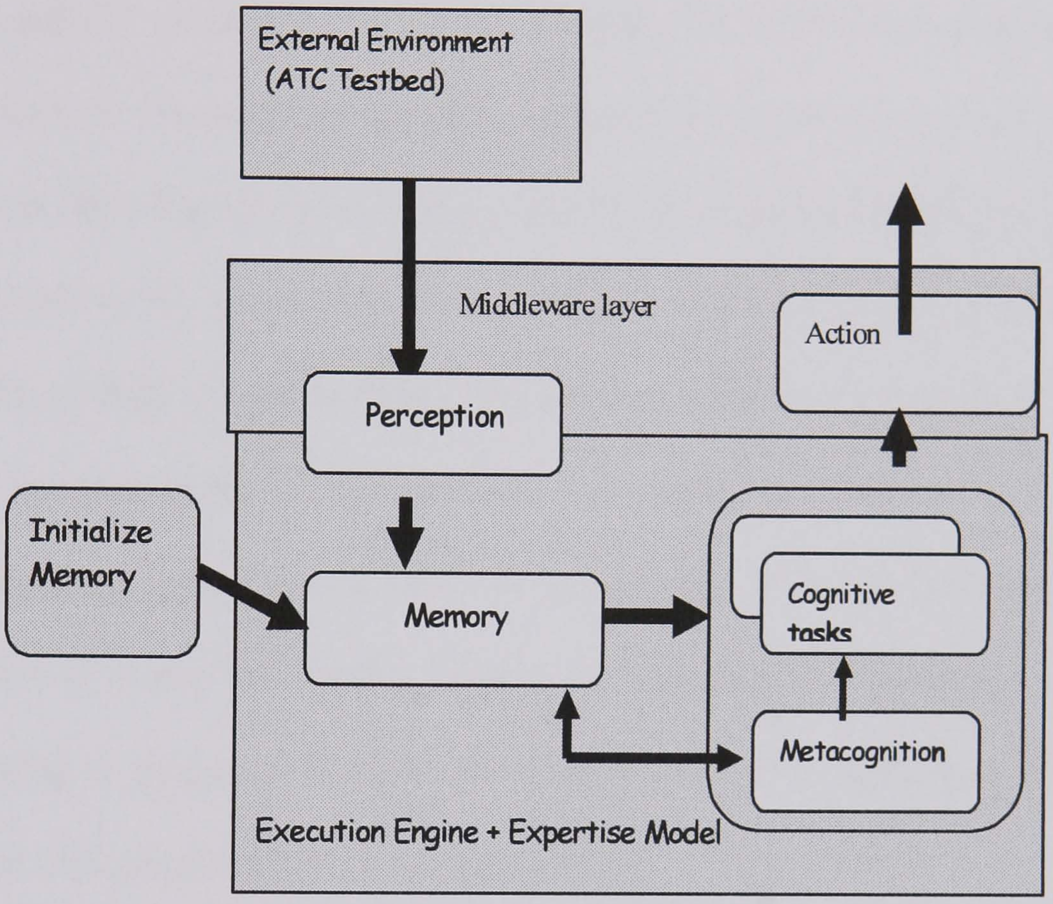


Figure 4.2 Execution engine and Expertise model.

The execution engine and expertise model is a metacognition model. Metacognition layer can be added, from a simple to expertise models. As Figure 4.2 depicts, there are three levels in this model: (1) external environment; (2) shell or middleware layer; and (3) metacognition. Information can be received from the external environment through perception, and can be transferred to memory. The external environment is connected to the different cognitions like perception, memory, decision making etc. These cognitive tasks are controlled by metacognition. There are three stages in metacognition layer: (1)

metacognitive knowledge; (2) metacognitive experience; and (3) metacognitive regulation (Flavel, 1977).

Metacognitive knowledge contains a database of knowing about an environment, the nature of the task, and strategies used for knowing the facts. Metacognitive knowledge contains three types of knowledge: (1) declarative knowledge; (2) procedural knowledge; and (3) conditional knowledge (Peirce, 2003).The declarative knowledge is the actual facts of the information. This contains the agent's or person's knowledge about formula, knowing the facts, places, etc. The procedural knowledge is knowledge about execution of the given facts. For example, in solving a mathematical problem procedural knowledge is used to select which of the available formula in the declarative knowledge are appropriate to the problem. The conditional knowledge is knowledge about particular skill and strategy used for conditions. The experience after applying metacognition to a cognitive task is termed as metacognitive experience. Controlling and monitoring a progress of cognitive task is termed as metacognitive regulation (Peirce, 2003; Wilson & Keil, 1999; Adkins, 2004).

4.5 Generic architecture for Metacognition

Metacognition can be used as a generic concept for computational theories with respect to problem solving, reasoning and the decision making. Metacognition can be applied on simple to complex cognitive architectures.

As depicted in Figure 4.3, a generic architecture for metacognition consists of three layers: (1) application layer; (2) metacognition; and (3) metacognition architectures.

The second layer is called as metacognition consist of three levels: metamemory, metacomphrension and self regulation. Metamemory stores strategies used for executing a task. An execution strategy includes execution processes and knowledge about the environment. Metacomphrension is used for detecting and rectifying the errors. The self regulation or metamanagement is using for adjusting an error thoughts and to give a feedback. Adding metacognition concept on top of cognitive architectures, improves the performance. This is similar to updating a systems memory or processor speed of a computer. The architecture remains same but it shows advanced behaviour and functioning.

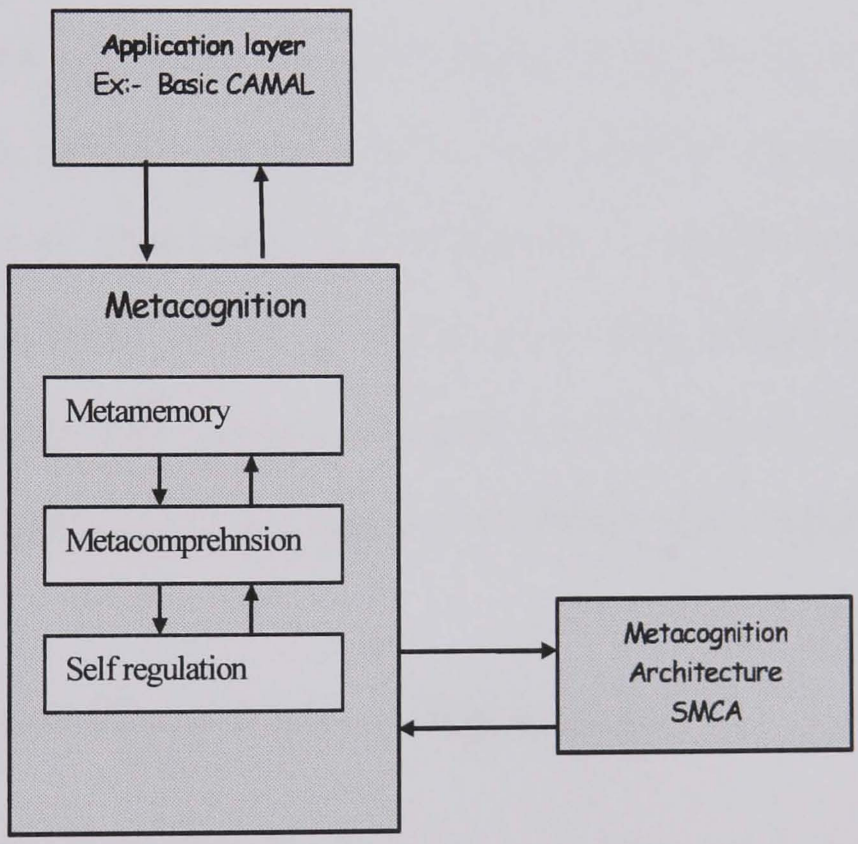


Figure 4.3 Generic Architecture (My New Perspective).

4.6 Metacognitive aids or Metacomponents

Metacognitive aids or metacomponents are used for the representation of thoughts. Metacomponents can be represented with the help of some aids such as: (1) using an abstraction, metasyntactic variable (matching variables) or metacomponents and; (2)

goal setting variables for increasing the performance. Metacomponents affects on the agent behaviour from a sense of what is important instead of what to do. Metacognition agents will follow well aligned norms, perceptual range, metarules, and learning and affect values. A well driven agent will maximize its performance as a consequence of learning to maximize its own reward. These executive processes involve planning, evaluating and monitoring the problem solving activities (Zalta, 2005, Adkins, 2004).

The term "norm" is an interdisciplinary term, and can be used to refer to a standard principle or a model used for a right action. The executive processes that controls the other cognitive components are responsible for "figuring out how to do a particular task or set of tasks, and then making sure that the task or set of tasks are done correctly". Norms in society of minds can be guided, controlled and regulates the proper and acceptable behaviours. Norms are recent development in cognitive science and artificial intelligence. The different models can be formed by using multiple norms. Norms can be used in the social laws, learning of norms, etc (Livingston, 1997). For example perceptual range, affect, norms, and higher level rules are metacomponents

4.7 Different Types of Monitoring systems

According to Kennedy (2003) there are three types of monitoring systems: (1) reflective rule-based systems; (2) metacognition for plan adoption; and (3) introspective reasoning. Kennedy defines that "A network of mutually monitoring agents are closed, but not necessarily reflective" (Kennedy, 2003). According to Kennedy, there are two types of reflective-rule based systems. This uses mathematical concepts, and other rules. The rules are operating on object level and meta level. The meta-rules are kind of meta-management. The metacognition system monitors and reasons during its plan adoption.

The plan adoption to new situation using its previous experience. Kennedy argues that Metacognition provides an efficient plan adoption. Metacognition can not monitor the integrity of agents at hostile environments. Finally, the introspective reasoning used on refines memory search, and retrieval. This is similar to finding an expected and ideal behaviour with reasoning.

4.8 Minsky A B & C-Brain

Minsky (2002) addressed the possible inner mechanisms, and higher level thinking of the mind. Minsky initially postulated an A-Brain and B-Brain mechanisms (Figure 4.4). A-Brain is connected to outer world through sensors and effectors. A-Brian collects information form the outer world or environment. A-Brain will control the cognitive tasks or mental processes in the architecture. The mental processes include perception, memory, imagery, language, problem solving, reasoning and decision making activities. B-Brain act like a supervisor for A-Brian. When A-Brain stops or struck or in confusion state to react, then B-Brain makes self reflection of A-Brain. B-Brain can supervise an A-Brain without understanding A-Brain working mechanisms.

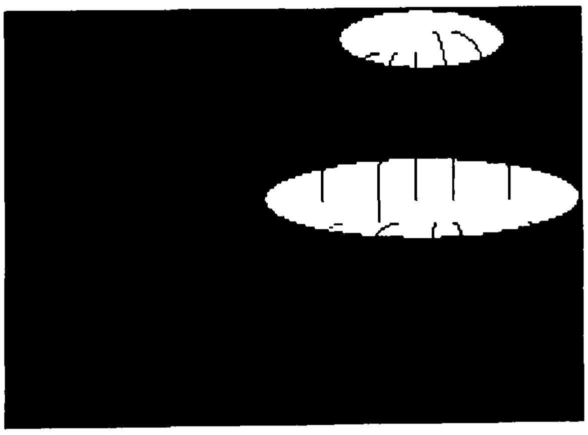


Figure 4.4 Minsky A B and C-Brain (Minsky, 2002).

Minsky suggested that, A & B-Brains can have C-Brain. This can control, watch, and influence the B-Brain. B-Brain and C-Brain works as similar to the A and B-Brains. In addition to this, Minsky suggested "closed loop" concept. The closed loop concept follows transitive mechanism. For example, B is a supervisor of A, C is a supervisor of B then C is also a supervisor of A. According to Kennedy (2006), A and B-Brain's can not mutually monitor and can modify each other. This is called as closed system, but not reflective.

4.9 "Blind spot" or "Reflective blindness"

According to Kennedy (2003), Minsky's A and B-Brain architecture interacts with the real world and, follows "closed system". The simpler system is sufficient and appropriate. The simpler systems can self-observe, and monitor to detect the errors or anomalies. Kennedy argues that, higher brain, can not detect the M-brain, when code has been deleted or replaced, from the lower level. This needs to add other layer to resolve this problem. This may end up with infinite number of brains. This weakness or problem is called as "blind spot" or "reflective blindness".

Kennedy (2003) framed the possible configurations of reflective networks. As Figure 4.5 depicts, each circle represents an agent. He explained that, the term "agent" is the highest level of component in a diagram, and this is sequentially controlled and hierarchically organized. The arrow indicates the monitoring relationship between agents, and all components. The monitoring system is similar to metalevel. This arrow also indicates equality in distributed monitoring.

Meta-level can be used in the reflective programming literature. This executes or interprets and monitors an object level (discussed in section 4.3). As depicted Figure 4.5(a), shows a "centralised" monitoring type. The single agent detects its errors, from self execution. Figure 4.5(b) shows an open reflection, where B monitors A, not viceversa. Figure 4.5(c) shows a closed system, where all meta-levels are object levels. Figure 4.5(d) shows an open reflection because there is one agent exist C, source of an arrow, but not a destination.

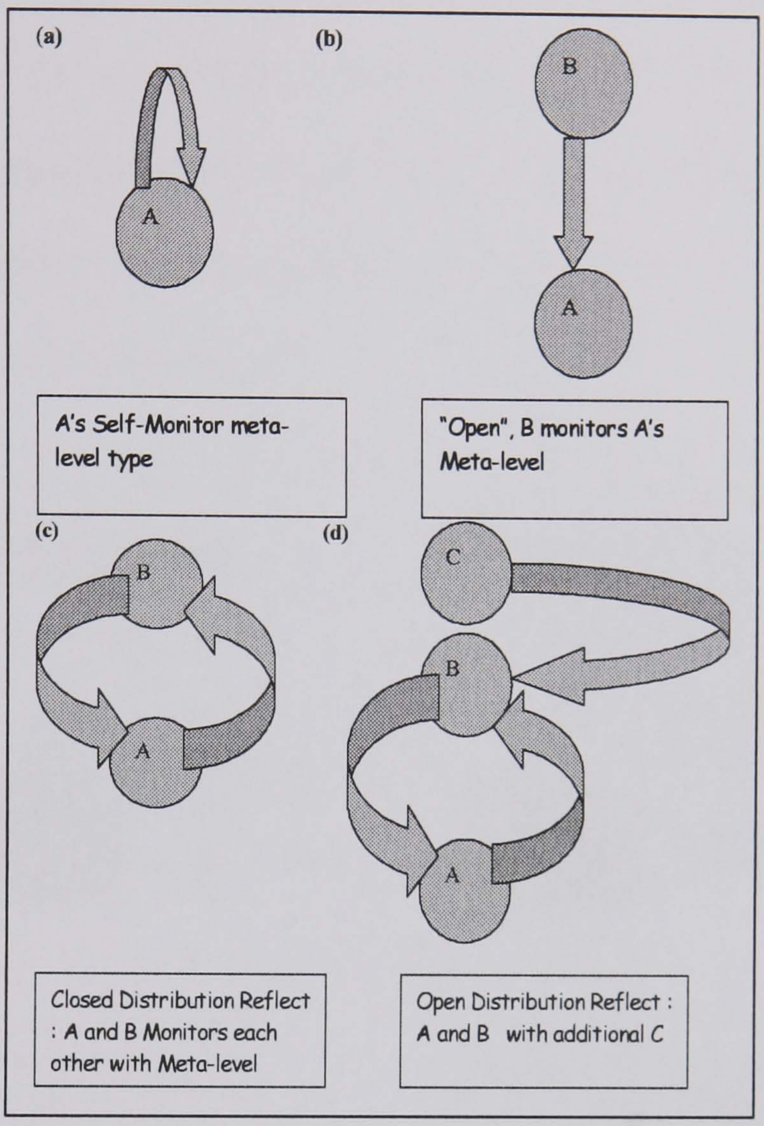


Figure 4.5 Types of Meta-levels

Kennedy argues that, mutually monitoring agents may be closed, and may not be reflective. She also proposed more complex configurations by using above examples. For example she describes organisational structure as follows: "L1 monitors L2, which monitors L1, which monitors L2, which...." (Kennedy, 2003).

4.10 Metacognition applications

Metacognition can be applied on different applications: (1) problem solving through computational models; (2) education field; and (3) human problem solving, etc.The problem solving is a one area where, a natural mind (robots, animals, humans) fits for the artificial computational theories in artificial intelligence. The executive control and monitoring are important divisions of problem solving to manage problem complexity and to evaluate progress towards goals.

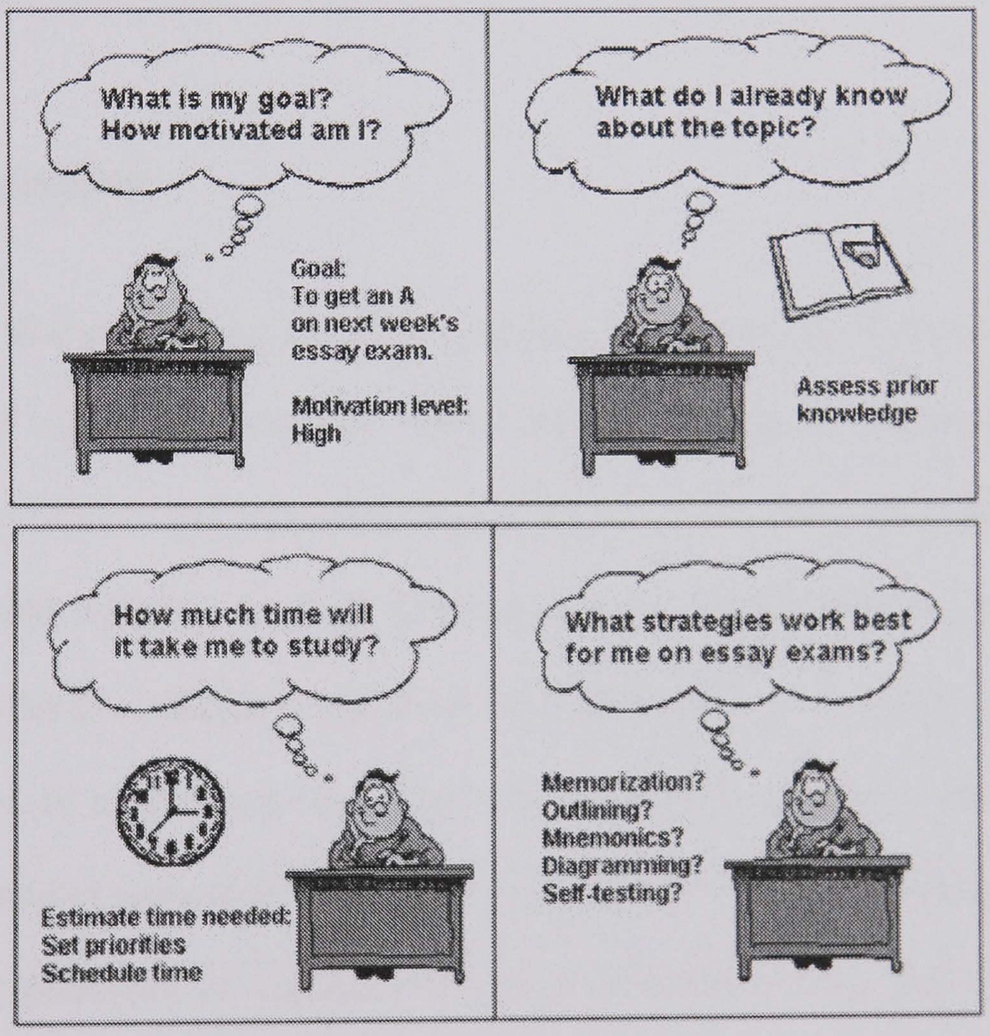


Figure 4.6 Metacognitive Strategies for Successful Learning

Metacognition concept can be richly applied on educational field. This includes theories of human cognitions can be improved by using self reflection or metacognition. For example (Figure 4.6) taken from (Halter, 2004), shows children's improved progress through successful learning.Metacognition systems are useful for improving the results by learning. As Figure 4.6, depicts, their are four metacognitive strategies for a student to learn and increase his performance:(1) aware of student goal and motivation; (2) knowing his own known information; (3) estimating the time required and time scheduling; and (4) learning; and self testing (Halter, 2004).

The number of cognitive scientists has built computational models for the human performances related to metacognition. Two ways metacogniton technique can be applied on humans: controlling and monitoring cognitions; and self reflection of individuals their own mental process.

4.11 Summary

Chapter four discussed about the nature, scope and applications of metacognition with the field of cognitive science. This chapter explained elements, strategies, metacognition models and metacomponents. Major issues about execution engine and expertise model, Minsky A, B & C-Brian, generic architecture for metacognition, Kennedy's reflective blindness, and Kennedy (2003) reflective networks are discussed. Metacognition is a relatively new buzz word in cognitive theory. Metacognition is defined as thinking about thinking. It can be viewed as two ways monitoring a group of agents in an intelligent or cognitive or robotic architecture (i.e. self reflection) and making changes by adapting effective strategies in that society of agents, to constitute a metacognition architectures.

Representation of thoughts (Zalta, 2005, Adkins, 2004) can be made by the help of some aids such as: (1) using an abstraction, metasyntactic variable (matching variables) or metacomponents; and (2) goal setting. The term norm is an interdisciplinary term. This term can be used as a standard principle or a model used for a right action."

figuring out how to do a particular task or set of tasks, and then making sure that the task or set of tasks are done correctly". Combining agents or society of minds can be guided, controlled and regulates the proper and acceptable behaviours. The simpler systems can self-observe, and monitor to detect the errors or anomalies. Kennedy points that there is a weakness in Minsky's A and B brain. The higher brain, can not detect the M-brain, when code is recently deleted or replaced, from the lower level. This needs to add other layer to resolve this problem. This may end up with infinite number of brains. This weakness or problem is called as "blind spot" or "reflective blindness".

Chapter 5 Agents, Testbed and the "Society of Mind".

5.1 Introduction

Artificial mind can be viewed as a control structure for an autonomous software agent. Any cognitive or computational architecture can be viewed as either a single agent or a large collection of agents. There is a long history of representing mind as collection of agents, dating back to Selfridges's Pandemonium model (Selfridge; 1959). This model attempts to explain mind as a collection of agent type tiny demons. The pioneers such as Selfridge(1959), McCarthy(1962), Allen Newell and Herbert Simon(1972), Minsky(1977), Fodor(1982), Baars(1988), Brustoloni (1991), Anderson(1993), Franklin(1995), Sloman(2001), Davis(2002) and Singh(2004) were viewed computational theories of mind, from artificial agents.

Different skills and cognitive tasks may be represented as individual micro agents. These individual micro agents will demonstrate simple, complex or intelligent behaviour, and serve to fulfil the capabilities expected of an intelligent agent, such as planning, decision making, problem solving, and learning. The purpose of this research is to understand the theory of natural minds and adopt these principles into simulations of artificial minds. The theory of mind includes abstract and broad sketches of architectures to support the functioning associated with mind. The design and implementation of a specific architecture follows hypotheses about human and nonhuman minds. This broad approach necessarily requires designing different computational simple and complex level agents. Agents are verified by seeing how they coordinate their goals by planned solutions and the general process of cognition to improve performance (Franklin, 1995, 1996, 1997).

5.2 Agent Classifications

An agent senses and acts in its environment. The researchers involved in agent research have offered a variety of formal and informal definitions for an agent. Russell (1995) defines an agent as "anything that can be viewed as perceiving its environment through sensors and acting through the environment through effectors" (Russell and Norvig, 1995). Brustoloni (1991) says that "autonomous agents are systems capable of autonomous, purposeful action in the real world" (Brustoloni, 1991). Intelligent agents continuously perform three functions: (1) perceptions, (2) action to effect a change in conditions and (3) reasoning to interpret perceptions, solve problems, draw inferences and determine actions. Some of the relevant agent classifications are explained below.

5.2.1 Brustoloni Agent Types

Brustoloni (1991) classified three types of autonomous agents: (1) regulation agents, (2) planning agents and (3) adaptive agents. Regulation agents follow a set of predefined rules and regulate things, similar to the way a thermostat controls temperature. There are four types of regulation agents: (1) problem-solving agents (2) case-based agents, (3) operational research agents and (4) randomizing agents. Problem solving agents may search for planned solutions, and some agents can provide satisfactory solutions. The agents can store or remember their moves, plans, and actions.

A case-based agent uses the search and analogy method. Case-based agents can store plans, and test their application in specific circumstances. To solve a problem, a case-based agent finds the most suitable plan. Operational research agents use a mathematical model, such as a queuing theory, to provide an optimal control. Randomizing agents simply work by trial-and-error methods. Planning agents follow regulation agents with a planned sequence of actions. The adaptive agents learn by chunking and other methods involving learning and modification (Brustoloni, 1991).

5.2.2 Sloman Agent Types

Sloman (2001) defines an agent as a "behaving system with something like motives". Agents can sense and act on the environment. Sloman classifies agent groups based on motivations such as thirst, hunger, sex, communication, preference, society norms, etc. According to Sloman agents can compare and visualize plans, sense and memorize, to various extents based on their degree of mind. According to Sloman, to make a complete functioning human agent needs the design of a human-like flexible architecture. This architecture may be biological or synthetic or take the form of a robot agent. This architecture needs integral diverse capabilities. The agent's architecture requires a wide range of cognitive science components such as vision, speech understanding, concept formation, rule learning, planning, motor control, etc (Sloman, 2001, 2002, and 2003).

Sloman argues that a part of human-like agent should have a diverse collection of tasks, both externally and internally. Internal actions include generating the motives, verifying the motives, selecting the motivations, creating plans, judging inferences, creating, monitoring, and identifying new possibilities. External tasks involve actions such as finding and eating food, avoiding enemies, building houses, making tools and finding friends (Sloman, 2001, 2002, and 2003). Sloman argues that there is no particular or unique design for human intelligence. There is no fixed architecture for an intelligent agent. This also includes many kinds of human learning, such as learning to drive a car, learning to read and write text, learning to play a piece of music, learning to write software, and learning many sports skills. Sloman classified the agents based on the three layers, as shown in Table 5.1.

Reactive sub system	Follows external sensors and internal monitoring, acts like more primitive parts of the simple brains. Example like insects.
Deliberative system	Follows the reactive system and works with triggering responses like non human minds.
Metamanagement system	Monitoring and controlling the deliberative models.Metamanagement activities like self monitoring and self modification capabilities.

Table 5.1 Sloman Agent Types

5.2.3 Franklin Agent Types

Franklin compares natural kinds of taxonomy with artificial agent classifications. Franklin contends that there are two possible models for building a mind from agents:

biological and mathematical. Franklin argues that artificial agents can be classified in a way similar to biological taxonomy (Table 5.2). For example, humans belong to the animal kingdom, sub family “hominid” and genus-species “homo sapiens.” He believes that artificial or computational agents, such as task-specific agents, entertainment agents and computer viruses, can be classified in a similar way (Franklin, 1997). According to Franklin, giving a definition to autonomous agents is too restrictive; the properties of an agent provide a better method of classification. Franklin named several agents based on their properties, as shown below.

Agent name	Meaning
Reactive agents	Senses and acts in the environment through timely fashion.
Autonomous agents	Control exercises over the actions.
Goal oriented agents	Purposeful actions not based on the environmental conditions.
Temporarily continuous agents	These agents are continuously running agents.
Communicative agents	Agents have social interaction and communication with other agents.
Learning agents	Changes the behaviour based on the previous experience.
Mobile agents	Agents can transport from one machine to other

Table 5.2 Franklin agent types

5.2.4 Minsky's Agent Types

Minsky states that a complete cognitive agent needs four separate and highly interrelated layers (Minsky, 1985). This necessitates the consideration of ongoing arguments in agent research. Each mental agent can, by it self, do some simple things, and when these agents are joined in a special way, the result may lead to true intelligence. Most of the "agents" grow in the mind, from learned experience.

According to Minsky, each agent looks simple and smaller (micro) like a toy, and does only small cognitive tasks. Combining all these micro agents in a meaningful way, almost anything can be built (Minsky, 1985). Minsky considers the following properties: (1) how the agents work, (2) origin and heredity, (3) learning and authority, (4) communication, (5) self awareness or consciousness, (6) feelings and emotions (7) ambition, jealousy, and humour. Minsky argues that creating machines that do the entire range of things people do is very far in the future, if it occurs at all. According to Minsky (1985), intelligence is a combination of relatively simple things. Imagine a child playing with building blocks, and how the child likes to watch a tower grow as each new block is added. Minsky says that the mind is like a tower, except that it is composed of processes instead of blocks.

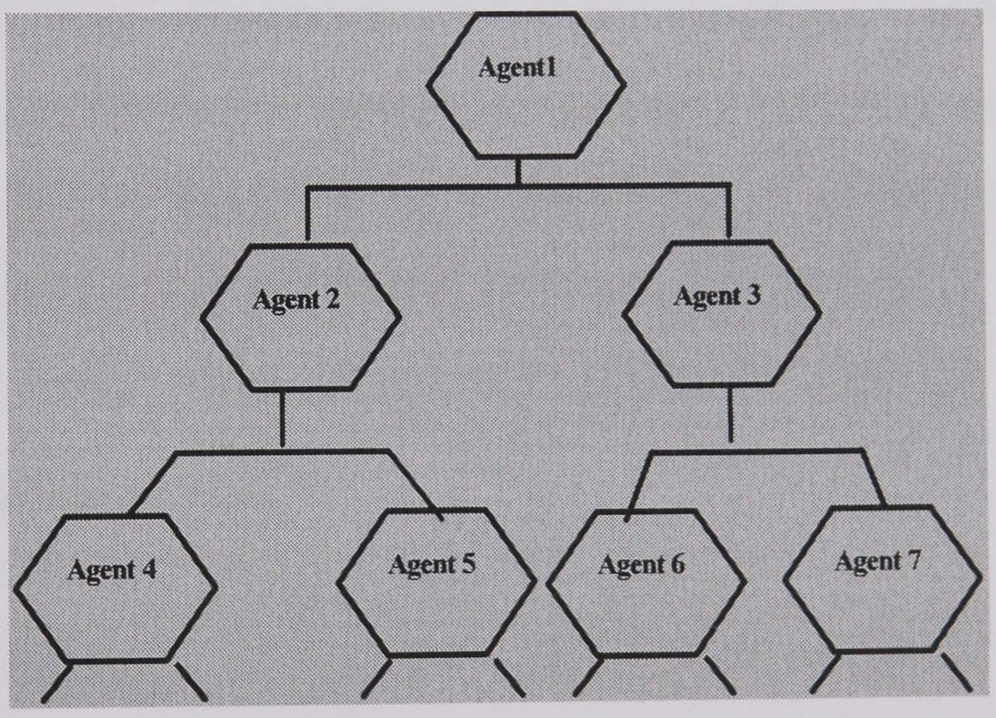


Figure 5.1 Society of Agents (Minsky, 1985).

As shown in the Figure 5.1, agents can be designed in a way similar to a child playing with building blocks. Any cognitive architecture contains a large collection of micro agents. Each agent may used in a different way to represent knowledge and reason with it. Each agent is specialized for some type of knowledge or cognitive process (Singh, 2004). Building an optimal agent cannot be done with a single and simple agent, as it needs to interact with or take help from other agents. Hence, developing a cognitive architecture can be viewed from the perspective of Minsky (1985), which leads to the development of many different types of simple agents with different behaviours. Figure 5.1 depicts a tree (graph) like structure, similar to tree concept in graph theory. This contains nodes and branches. Each node represents an individual micro agent. Each branch represents a link between nodes. This illustrates Minsky's K-line theorem. The K-lines are data and control lines (buses) in the design of computer architecture. Assume if there are two different cognitive tasks T1 and T2 to perform in a society of agents. Agents 2, 4 and 5 can perform T1 and, agents 3, 6 and 7 can perform T2 cognitive task. Afterwards T1 and T2 performing agents can be combined as T1 agency. Similarly, any number of agents and agencies can be combined to form as "Society of Mind". Society of Mind can be framed from any smaller degree to any large extent. For example, human mind as a "Society of Mind" is larger, and rat mind as a "Society of Mind" is smaller in degree, with a smaller set of agents and agencies.

5.3 Society of Mind

The Society of Mind theorem was initially proposed by Marvin Minsky in the 1970s, at MIT's AI lab. Minsky's inspiration dates back to the 1960s and the famous 'copy-demo' project. At the end of the 1960s, Minsky, Papert and his students developed one of the first autonomous hand-eye robots. The hand-eye robot demonstration involved the robot constructing children's building block structures. It uses a camera to see, and a robotic hand to move. From this idea Minsky framed the term "Society of Mind." Minsky argues that, a mechanical hand, television eye, and a computer robot can build a block structures. This technology took many years for the researchers to analyse cognitive operations like seeing, grasping, and move through developed micro-agents.

Minsky argues that, this development gives many ideas for "Society of Mind" (Minsky, 1986). Minsky views intelligence as not just a simple recipe or as an algorithm for thinking, but a combined social activity of more specialized cognitive processes. According to Minsky, every mind is a "Society of Mind." The mind consists of a great diversity of mechanisms. Minsky proposes that the mind is made up of simple and smaller entities called micro-agents. Minsky argues that each agent is like a simple piece of code, and can do simple work. The agents can be connected within a larger system called a society of agents. Each individual agent, having a different background, plays a different role in society. The society of mind results from combining more specialized cognitive processes.

Minsky views mind as a vast diversity of cognitive processes, each specialized to perform some function. Some important functions include predicting, expecting, acting, remembering, explaining, comparing, generalizing, analysing and ways of thinking. To handle this complex diversity, Minsky introduces the simple terms "agent" and "agency." Minsky states that the term "agent" describes any component of cognitive processes. The term "agency" describes the specific combination within a society of such simple agents combined to perform some more complex function. As similar to agents in agency, "Society of Mind" can have multiple agencies. Minsky ask, rhetorically, "If two minds are better than one, how about two thousand?" (Minsky, 1991).

Baars (1977) "Global workspace theory", explains mind in terms of conscious and unconscious terms as similar to the on and off state of the "Society of Mind". In the

"Society of Mind", the active agents are in the "on" state, and non-active agents are in the "off" state. The combined activity of active agents constitutes a "total state" of mind, and the subset of the activities represents a "partial state of mind." Minsky claims that K-lines are the selecting most common agents in the Society of Mind theory. These K-lines turn agents on, and are interconnected to each other. K-lines can cause cascade effects within a Mind. Many K-lines problems and their solutions are stored in a 'chunking' mechanism. If the system faces the same type of problem again, then it uses the previous solution to resolve it. Minsky divides these K-lines into two general classes: (1) nemes and (2) nomes. These are similar to data and control lines in the design of a computer. Nemes represents aspects of the world, and nomes control these representations. The nemes are divided into polynemes, and micronemes. The polynemes agencies are concerned with representing properties of an object. A micronemes agency provides global contextual signals across the brain. The nomes are divided into three types: (1) isonomes, (2) pronomes, and (3) paranomes. The isonomes signals for different agencies (collection of agents) to perform the same type of cognitive operations. The pronomes are used for controlling the short-term memory representations. The paranomes are used to represent the changes to related operations. Minsky argues that K-lines can learn by accumulating and reformulating knowledge. Accumulating is the remembering of an example, for simple learning (Minsky, 1980, 1985; Singh, 2003).

As was quoted at the beginning of last chapter, "The Society of Mind is more than a collection of theorems. It is a powerful catalyst for Thinking about Thinking" (Singh, 2003). Singh argues that any Society of Mind needs thinking about thinking (i.e., metacognition) to make complete functioning of mind.

5.4 Proposed Generic Cognitive Agents

The currently developing SMCA (Society of Mind approach to Cognitive Architecture) makes use of a generic architecture, and developed in terms of generic cognitive and metacognitive agent types. It aims to model cognitive abilities, functions and mechanisms (e.g., planning, optimal decision making, problem solving and learning) in terms of combinations of agents or isonomes. Each agent is designed to fit one of the following categories: (1) reflexive agents, (2) reactive agents, (3) deliberative (BDI models) agents, (4) learning, (5) metacontrol and (6) metacognitive agents.

Reflex action is basically derived from human and animal biological neuromuscular action. The reflexes are built-in mechanisms where action can occur quickly, before thinking. In some cases, reflexes can be changed or overridden; a reflexive agent does not have any explicit motivational states like belief, desire, or intentions. For example, in the developed testbed, a reflexive agent can move in one of four directions (left, up, down, right) in response to the nature of the environment immediately in front of it; simply moving into free space and away from obstacles.

Reactive agent mechanisms, having more flexible control mechanisms, similar to the architecture described by Kaelbling (1986). This class of agent has extra perceptual pathways and mechanisms for integrating decision making, and behaviours across intended actions. For example, in the developed testbed, reactive agents can follow a specific goal. The goal is to identify the resource, move shortest way and collect any one or more resources (fungus, ore, crystal, medicine, standard ore, golden ore, ore and golden ore, ore and crystal, etc) available in testbed.

The reactive class of agent, in turn, provides a computational platform for the deliberative agents. Deliberative or BDI (Belief-Desire-Intention) agents build on the behaviours used in the reflexive and reactive agents. The deliberative actions are planned and coordinated in terms of the agent, its internal state, its motivations and its perception of resources in the environment. Mind is made of many small processes; these are called deliberative or mental agents. Each mental agent by itself can do some simple things. BDI (Beliefs, Desires Intentions) are the mental components present in rational agent architectures (Bratman, 1987; Cohen 1990; Rao and Georgeff, 1993). In the developed testbed, deliberative agents reasons about their own tasks and plans. Deliberative agents in a fungus world testbed are capable of performing different tasks. The deliberative agents can alter the reactive and reflective agents, from the reasons based on the Belief, Desire and Intention set. The deliberative agents can manage their internal conditions, through managing their metabolism and food (affect). Based on the management of energy, from given threshold value or predicting energy level, and goals different BDI models are framed (explained briefly in the next chapter).

Learning agent's main objective is to maximize the total rewards in the architecture (Sutton, 2004). The value function decides how to maximize the total rewards. The model of the environment defines how the state and action can occur in the different location of the environment. For example, in the developed testbed, q-learning algorithm applied on reflexive, reactive, deliberative, metacontrol and metacognition agents. The learning maximizes the collection of a needed resource or reward (fungus, ore, crystal, medicine, standard ore, golden ore, crystal, etc) available in testbed. A metacontrol agent operates in a resource-oriented environment. Metacontrol agent chooses between the available deliberative and reactive actions according to current conditions. If deliberative actions are called control actions, the learned control actions are called metacontrol actions. The metacontrol agents determine the relevant control actions (Raja, 2003). Metacontrol actions are part of metacognition. For example, in the developed testbed, metacontrol agent chooses the selective learned deliberative (BDI) model, but can not change and reason whenever necessary. The learning applied on BDI models maximizes the performance of a particular BDI agent.

Metacognitive agents can control and monitor their own progress in performing cognitive tasks or metacognitive regulations (Wilson & Keil, 1999; Adkins, 2004). Reflective processes with learning capabilities can lead to metacognition mechanisms. Metacognitive agents are necessary for the optimal decision-making capabilities in a varying environment. For example, in the developed testbed, metacognitive agents can choose any metacontrol task. They can select, change, update and reason for any metacontrol task (briefly explained in the next chapter).

5.5 Testbeds and Benchmarks

Testbeds and benchmarks are used for simulating and comparing architectures and outcomes in the field of robotics or cognitive architectures. A testbed is a development environment for experimenting and implementing standard tasks. The testbeds are the environments, where standard tasks may be implemented, observed and measured. In addition to the environment, it provides a method for data collection, the ability to control environmental parameters, and scenario generation techniques (Hanks, 1993). Testbed tools provide a method for data collection, ability to control environmental parameters, and scenario generation techniques. Testbeds provide a metrics for comparing the agent architectures. The main purpose of a testbed is to provide metrics for evaluation (objective comparison) in testing agents. Hanks (1993) define

"Benchmarks as precisely defined standardized tasks". Task means the job given for a robot to perform, and the standard means a benchmark accepted by a significant set of experts in the same field. Precise means a mission goal and limited constraints in the execution environment (Dillmann, 2004). According to Dillmann (2004), the Hanks definition for benchmarks is lacking in terms of "development performance metrics". A benchmark should have the following features: repeatability, independence and unambiguity. Benchmarks can be measured using two types of metrics: (1) the analytical method, for observing a system's performance and (2) the functional method, to observe the performance of a specific problem based on a benchmark score (Dillmann, 2004). According to Hanks (1993), benchmarks are using for comparing the architectural performance of the standard tasks. The results of different architectures can be compared and measured, from standard tasks. Artificial intelligence contains standard tasks for AI problems.

5.6 Toda's Simulator Testbed Model.

An important issue in developing testbed includes complexity, metrics, flexibility and the ability to perform well in different situations. There are many examples of developed testbeds. Those relevant to this research include Packman (Nason and Laird, 2004), Tile world (Pollack and Ringuette, 1990), and Fungus world (Toda, 1986).

5.6.1 Packman

SOAR capabilities are demonstrated using packman testbed. The packman tesbed consist of pack-man like agents (eaters) called as eaters, moving around the board or environment. The board is filled with 2 types of food: (1) bonus food and, (2) normal food. If the agent receives as a reward of +10 for moving into bonus food, +5 for moving into the normal food, 0 for moving into the empty cell in environment or grid board. Agents' capabilities are tested by giving different skills including reinforcement learning. (Nason S and Laird J E, 2004).

5.6.2 Tile world

Pollack and Ringuette initially introduced tile world testbed in 1990. Tile world is a highly parameterised environment, and this can be used to investigate reasoning an agents (Lees, 2002).The tile world is an abstract testbed designed for experimenting with multi agent architectures in dynamic and unpredictable environments.Tileworld is a two dimensional grid on which located different kinds of parameters. The parameters tiles, holes, obstacles and a gas station or energy. During the simulation part, objects can appear and disappear. The parameters can be controlled with variety of characteristics associated with the objects in an environment. The original tile world consists of a grid cells (squares) on which different objects can exist. These objects are: agents, tiles, obstacles and holes. The agent can move up, down, left, or right. The goal of the agent is to collect a tile and move a tile so as to fill the holes. A hole has an associated point value. Each hole may consist of three cells on the grid, and may have a total point value of five. Once the agent completely fills the hole, it earns the points. The overall objective is to gain as many as points possible. Tile world simulations are dynamic, and the environment can change continuously (Lees, 2002).

5.6.3 Fungus world

Toda (1961) compared a "complete system" with a microcosm environment. The fungus eater is a simulated robot used for fictitious mining. It has been sent to a planet called taros to collect uranium ore. The fungus eater can run out of energy and needs to collect fungus to replenish its energy store. Uranium and fungi are usually not found in the same place. They will keep a certain distance from each other in order to avoid collision (Wehrle, 1994; Lewis, 2004). Pfeiffer (1996) assumes: (1) movement or locomotion using legs, (2) collection or consumption using arms and (3) decision making using a brain. Fungus eaters are synthetic artificial agents and a particular species of animats. Animats can be productively viewed from a designer's perspective. Masanao and Toda (1961) invented a new fungus eater testbed, for simulating artificial synthetic agents. Multiple agents can be present in the environment at the same time.

As in Figure 5.2 depicts Toda (1986) proposed the use of micro worlds (a micro cosmos) for the purposes of modelling mechanisms and for collecting empirical data (Wehrle, 1994). The animats approach will play an important role in the design and experimentation on intelligence and cognition. The fungus world environment and working principles allows a robot or any artificial mind a simulation for exhibiting a specific behaviour in a specific environment (Pfeifer, 1996). Pfeiffer (1988) describes the "fungus eater" concept as a testbed for simulating models in emotion psychology. The fungus world environment allows the principles and behaviours of a robot or simulated models of artificial minds simulation can be monitored, measured and compared (Pfeifer, 1996).

Applying techniques, mechanisms and concepts on a testbed depends on the goals. Fungus eaters are complete autonomous creatures sent to a distant planet for collecting ore. They have to think and eat a fungus to maintain an energy level for their survey of the world and to show best performance (i.e. to collect ore). The performance can be measured in terms of two perspectives: (1) the engineering perspective, which counts number of ores collected in a particular time cycle and (2) the cognitive perspective, which looks at managing the energy level and metabolism level sufficiently to allow surveying (Pfeifer, 1996).

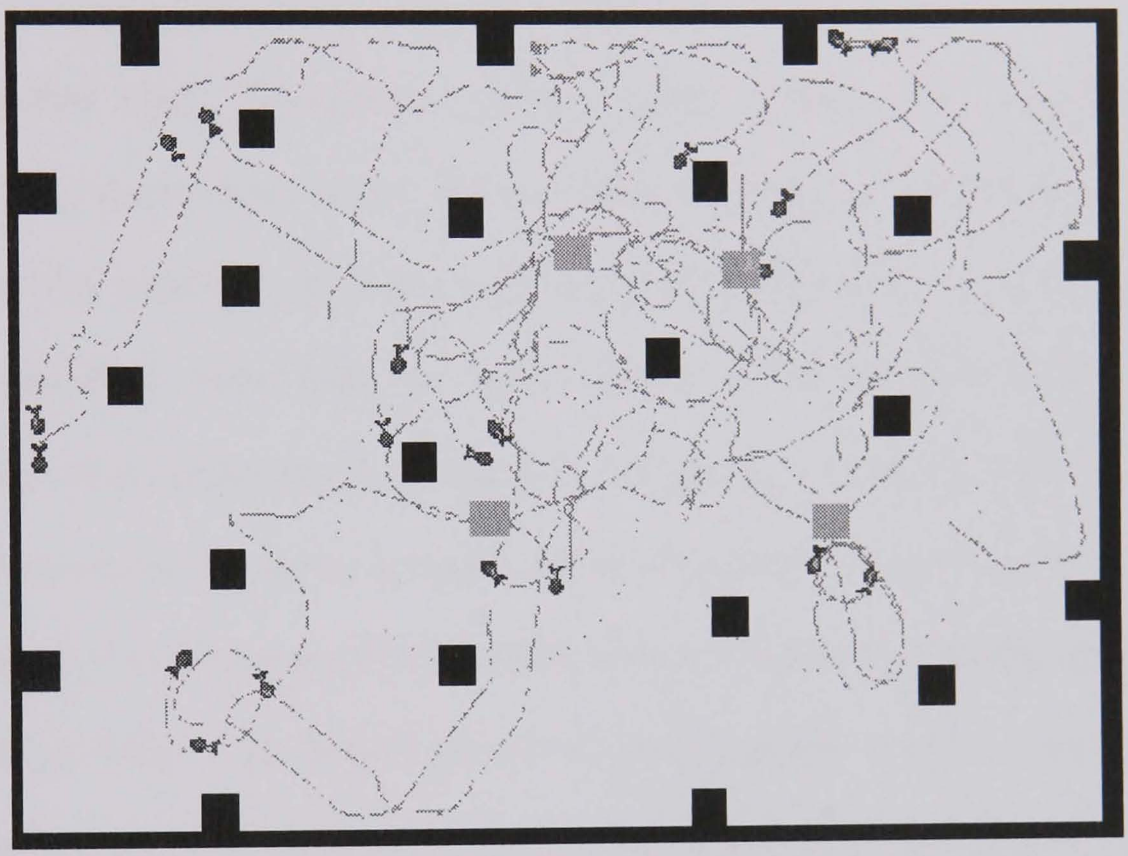


Figure 5.2 Toda's Simulator Testbed

Reasons for choosing Fungus world Testbed for implementation of SMCA:

- The set of objectives are rich to implement interesting aspects of the real world.
- The metrics used for fungus world is convenient and easy to use.
- The assigned parameters will map, interesting and measurable properties of the real world environment.
- Fungus world testbed is very flexible to extended simple to complex level.

5.7 Summary

Chapter Five discussed artificial agents and different types of agents, with a specific focus on developing new society of agents. Previous work relevant to this research includes Brustoloni (1991), Sloman (2002), Franklin (1997) and Minsky (1985). Agents can act and sense in their environment. Brustoloni (1991) explains three types of autonomous agents: (1) regulation agents; (2) planning; and (3) adaptive agents.

There are four types of regulation agents: problem-solving; case based; operational research and randomizing agents. Sloman defines agent as a "behaving system with something like motives". Sloman classified agent groups based on the motivations. Sloman classified three types of agent based on his CogAff model: reactive; deliberative; and metamanagement agents. According to Franklin (1995), similar to biological taxonomy the artificial agents can be classified. According to Minsky (1985), intelligence is a combination of more simple things. According to Minsky every Mind is a "Society of Mind". The Mind consists of great diversity of mechanisms. The term agent refers to the basic element or simplest individual which constitute a "Society of Mind". Each individual agent has a different background in order to play a different role in the society of mind. The combination of agents to perform specific tasks or demonstrate specific abilities is termed an agency.Testbeds and benchmarks are used for simulating, comparing architectures and outcomes in the field of robotics or cognitive architectures. Pfeiffer (1988) describes the fungus eater concept as a testbed for simulating models in emotion psychology. The fungus world environment allows the principles and behaviours of a robot or any artificial mind simulation to be monitored, measured and compared.

Chapter 6 Design of Society of Mind Model

6.1 Fungus World: A scenario for Mind as a Control System

This chapter describes the design of a SMCA (Society of Mind approach to Cognitive Architecture). SMCA model of mind using concepts allied to Davis basic CAMAL cognitive architecture (Davis, 2002) and Minsky's "Society of Mind" (Minsky, 1985). "Society of Mind" is a control system, and uses the "Society of Agents" metaphor. "Society of Agents" concept describes collective behaviours of simple and intelligent agents. This uses cognition to metacognition concepts for unification of agents. Metacognition has been introduced to control and unify society of intelligent agents.

Consider the scenario of the fungus world testbed shown in Figure 6.1. Simulated agents (actors) are represented as circle shapes. Different parameters exist in the environment for the agent's biochemical engine, including metabolism and performance. Biochemical engine parameters are an agent's energy level and its rate of use (metabolism). The medicine is a metabolic activity parameter, shown as gold colour square shape. The energy resources are: (1) standard fungus, green colour squares in testbed, (2) small fungus, small size green colour square shape and (3) bad fungus is represented by black colour squires in testbed. The biochemical engine parameters are energy level, metabolism,etc.The goal-based parameters are: (1) ore, red colour star shapes, (2) golden ore, gold colour star shapes and (3) crystals, white colour star shapes.

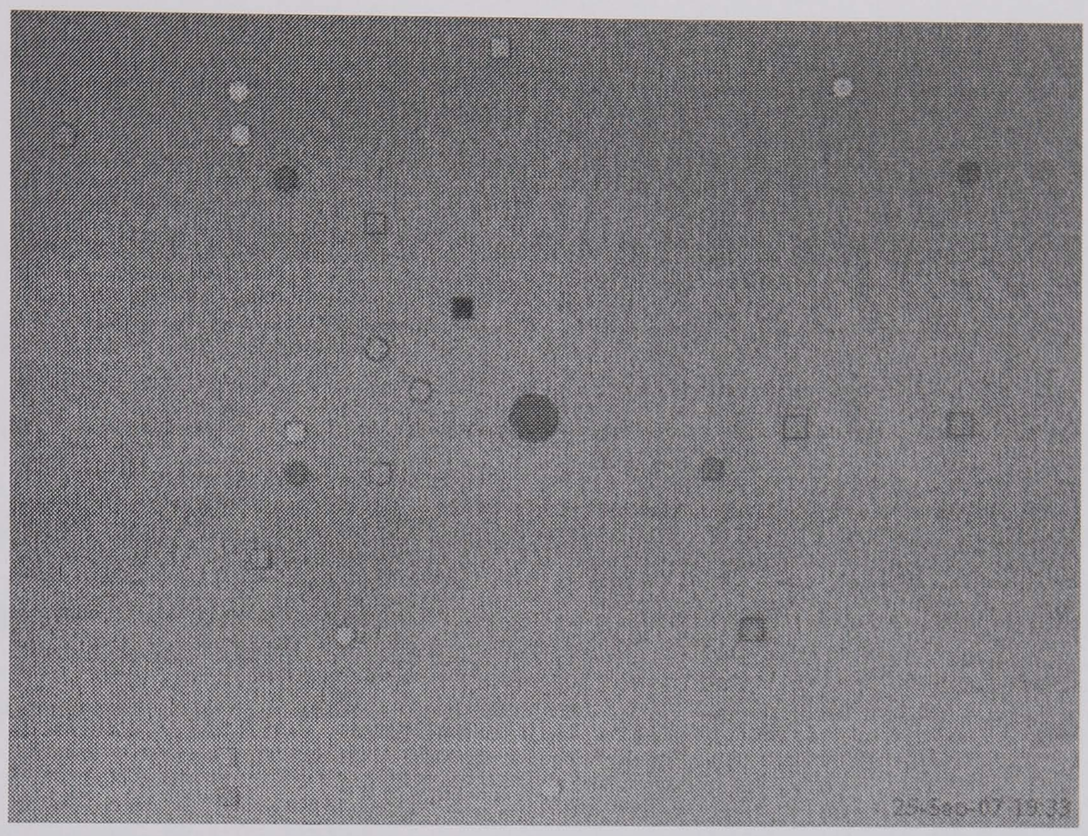


Figure 6.1 Society of Mind scenario

This simulated world is populated by several actors called artificial synthetic agents. These actors or agents are controlled, and guided by further agents and higher level mechanisms in the SMCA (described in the next section). The control architecture enables the fungus eaters to adapt to their dynamic environment. Actors are simulated agents that work together to exhibit various control mechanisms and techniques, thus demonstrating a distributed model of mind as a "Society of Mind".

The actors (agents) demonstrate the "Society of Mind" concept in terms of the arrangement of activities within a SMCA for their planning, reasoning, decision making, self reflection, problem solving and learning capabilities. SMCA can be viewed as containing vastly different types of cognitive processes, such as predicting, repairing, reviewing, comparing, generalizing, and simplifying and many other ways of thinking. Agent behaviours can be analysed using many different metrics. The major metrics are metabolic activity, competition and social interaction with respect to environment and microeconomics.

SMCA model fitted with reflexive, reactive, deliberative BDI (Belief, Desire, and Intention), perceptual, learner (Q learning), metacontrol, and metacognition agents. The reflexive layer is framed with reflexive behaviours that suggest courses of action based on the state of the environmental conditions. The reactive actions are shortest route or planned and coordinated actions in between the agent and resources. The deliberative layer helps for selecting and controlling reactive agents. The higher level layers such as learning; metacontrol and metacognition levels assess the actions based on criteria that have to do with actions that are consistent with BDI models.

Metacognition (discussed in Chapter 4) is used for adopting the Belief-Desire-Intention (BDI) models. The BDI models belong to the deliberative level of the architecture. The metacontrol and metacognition mechanisms used in controlling the BDI models belong to the top level of architecture. The relationship is as follows: (1) metacognition is a program that interprets BDI, and it takes BDI (BDI-ore, BDI-crystal, BDI-ore-crystal, BDI-adaptive, etc) as an argument and (2) BDI calls metacognition to monitor, inspect, modify, correct or improve the BDI. BDI agents use the metrics drawn from principles of artificial economics in animal cognition: (1) physiological and goal oriented behaviour; (2) cost function and utility behaviour at the microeconomic level; and (3) decision making variables (explained in the second Chapter).

The BDI (Belief- Desire-Intention) model allows different groups of coordinated capabilities to carry out a particular intention. BDI models follow and control the reactive mechanisms. The energy spent in each move of BDI type's exhibits as minimal due to maintenance of low metabolism, and utility is also maximized. A BDI agent engaged in activities optimizes its pattern of behaviour with respect to energy and time. For example, if the energy level is less than fixed energy level (threshold) or predicting energy, then it switches into fungus consumption. If the energy level is more than a given threshold or predicted energy level, then it switch towards goal oriented behaviour (i.e. collection of ore). This mechanism demonstrates physiological and goal oriented behaviour.

There are different BDI models for the different purposes used in this experiment. They are BDI-ore, BDI-crystal, BDI-ore-and-crystal, BDI-adaptive etc. The actors for each move compare, review and change their goals from their self-reflection processes, which is updated from the affect and norms. Actors can change and control their behaviours, such as switching between BDI-ore, BDI-crystal and BDI-ore-crystal, BDI-adaptive if necessary in the environment. The metacognition layer monitors the inference processes that occur in the deliberative and metacontrol processes. The activities of all of these courses of actions are managed by SMCA architecture.

6.2 Design of SMCA

Basic CAMAL has four tier and five column architecture (Figure 6.2) (Davis, 2002, 2004, 2007). This provides a basic template for all explanations (Davis, 2007). The extended CAMAL with extra processing layers, named as SMCA for distributed model of mind (Figure 6.3).

The SMCA model can be used to implement a variety of cognitive control structures. The design requires implementing different AI architectures and algorithms. The research described in this thesis implements "Society of Mind" architecture for controlling the actions of actors in the artificial life domain shown in Figure 6.3.

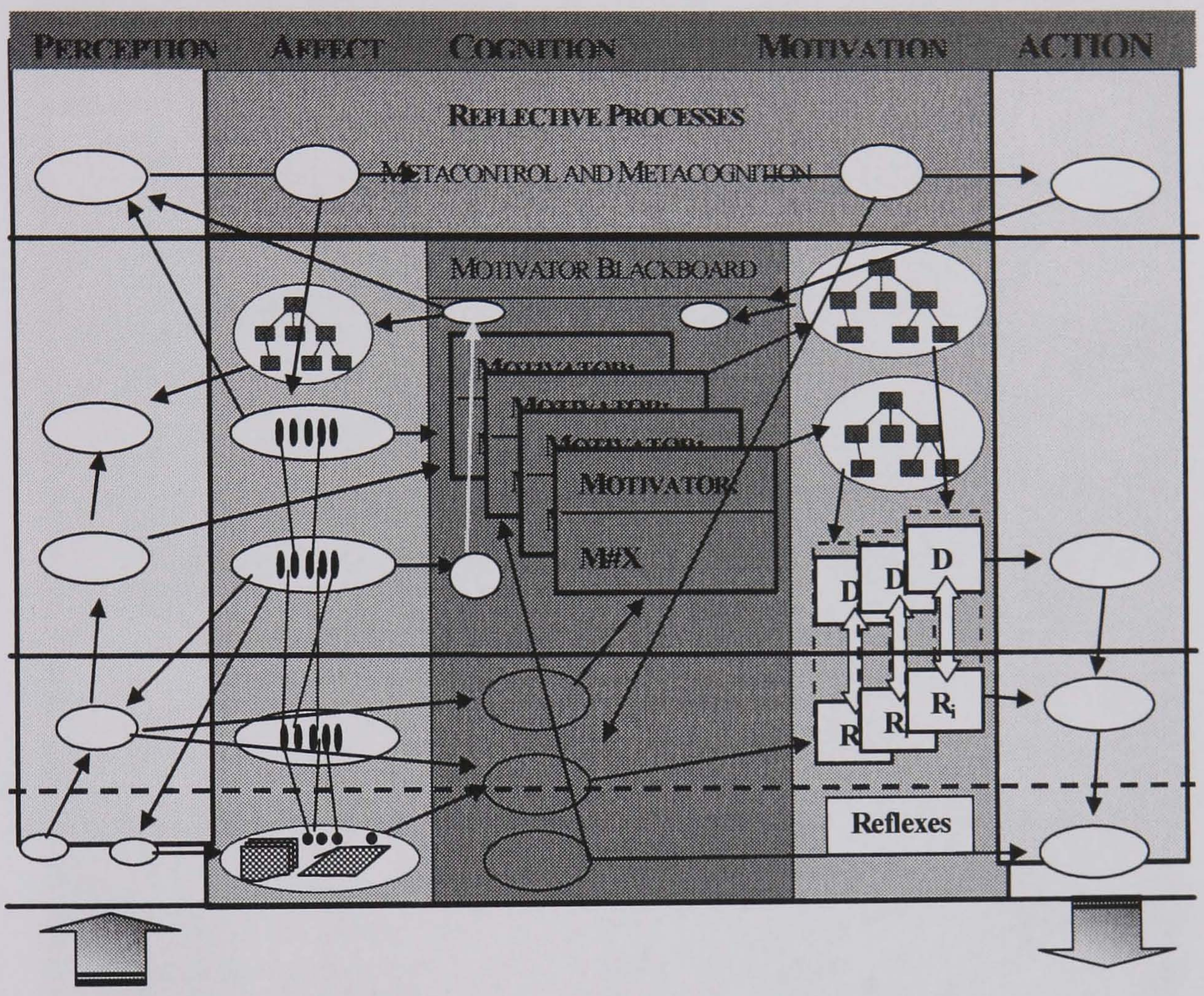


Figure 6.2 Basic CAMAL Architecture (Davis, 2002).

The reflective reasoning and multi-tier architectures are used because it is often difficult to assure optimal and perfect operation using just one layer. Hence a higher level layer that reflects upon that other layer can be added to help cope with its limitations. In this thesis six-layer architecture with reflexive, reactive, deliberative, learning, metacontrol and metacognition levels is described.

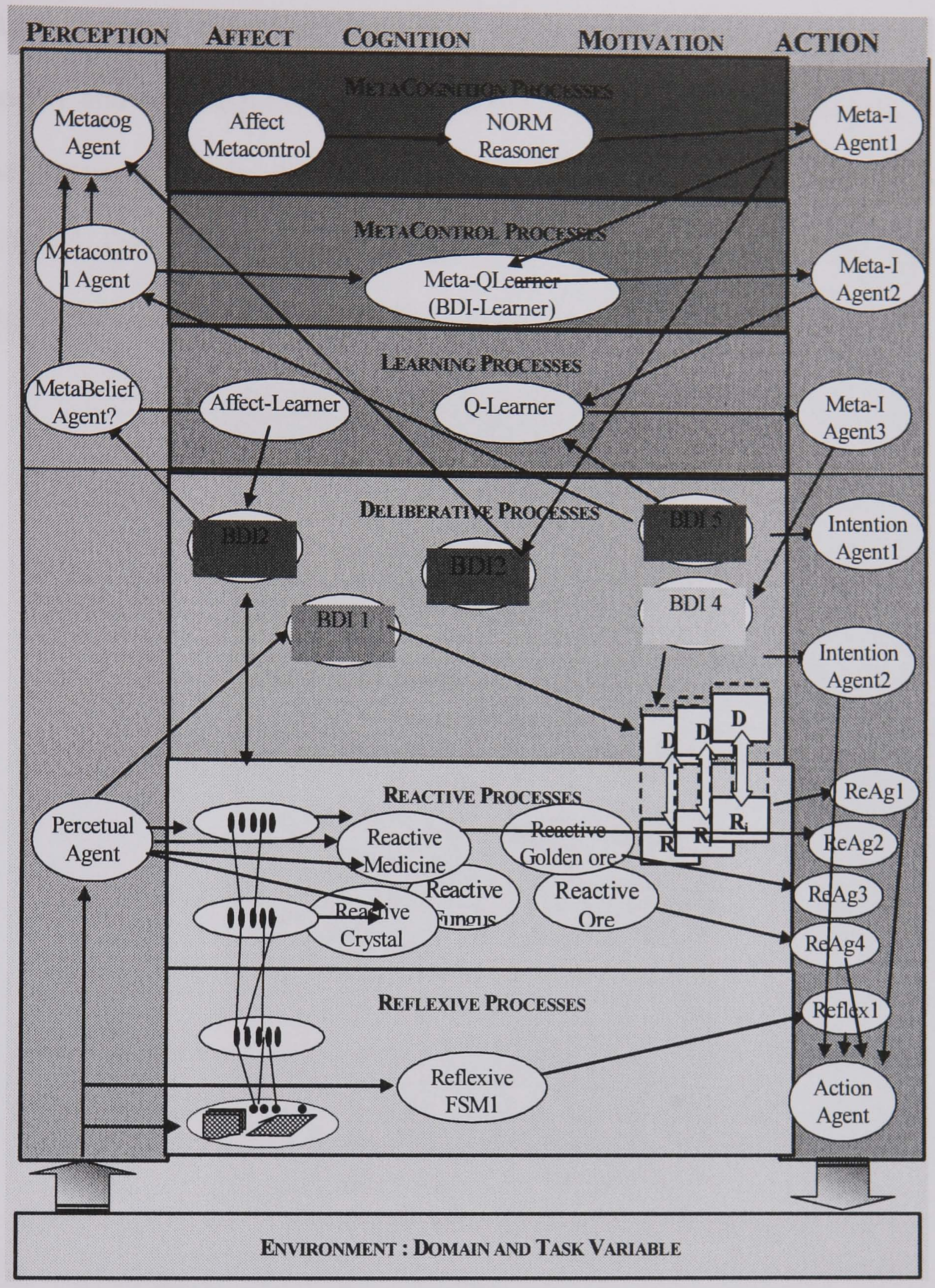


Figure 6.3 SMCA

Each of these layers is populated by several agents with behaviours that respond to problems in the layers beneath, or in the case of the lowest reflexive layer, to environment. The intelligence behaviour is a combination of simple behaviours. The presently developing mind model SMCA (Figure 6.3 and Figure 6.4) includes reflexive (six behaviours), reactive (seven behaviours), deliberative (fifteen behaviours), perceptual (nineteen behaviours), learning (fifteen behaviours), metacontrol (fifteen behaviours) and metacognitive (seventy seven behaviours) agents. Indeed, from an extreme perspective of the distributed model of mind is fitted with reflexive, reactive, BDI (Belief, Desire, and Intention) agents or deliberative, perceptual, learner (Q learning), metacontrol, and metacognitive agents.

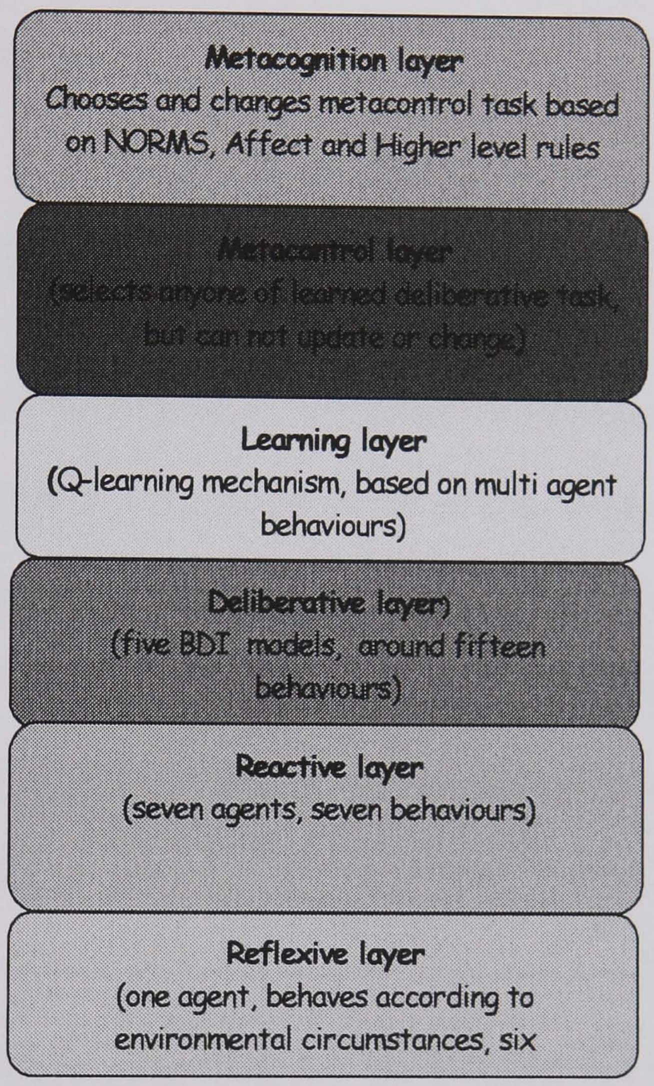


Figure 6.4 Group of Distributed Agents in "Society of Mind".

Combinations of agents organized to work or achieve the different goals (cognitive tasks). The different combination of agents is organized for different goals or different tasks: (1) collection of ore (2) collection of ore golden ore; and (3) collection of crystals can be performed by society of agents. K-lines can cause cascade effects within an SMCA. These K-lines turn agents on and are interconnected to each other. For example, the reactive-ore, BDI-ore and BDI-ore-crystal combination of agents can collect ore.

Reactive-crystal, BDI-crystal and BDI-ore-crystal combination of agents can collect crystals. This illustrates that the SMCA follows Minsky's K-line theorem.

Metacognition concept on BDI agents selects the most appropriate type BDI agents (BDI-ore, BDI-crystal, BDI-ore-crystal, etc). Metacomponents in Metacognition layer helps for switching BDI agents on and off state. Each of the individual layers is described in the next sections.

6.3 Reflexive Level

Reflexive agent's fits for the first layer of the SMCA (Society of Mind approach to a distributed Cognitive Architecture) shown in Figure 6.5. Reflexive agents are designed to perform reflexive behaviours. As Figure 6.5 depicts, agents will make decisions and take actions based on the given environmental rules. Reflexive agents are simple, reactive, and instinctual. Reflexive actions are controlled from a finite state machine. Generally, reflex action is basically derived from human and animal biological neuromuscular action. The reflexes are built-in mechanisms that can operate quickly before thinking. There are two ways that reflexes can behave: (a) simple reflex, which is automatic and requires no learning experience and (b) combined reflexes. A finite state machine behaves like a simple mathematical animal, that can be regarded as a discrete-time system with finite input and output sets. This responds to only a finite number of different stimuli (the input set or alphabet) and output alphabets. Algorithm 6.1 shows mapping of finite state machine (FSM) from perceptual inputs. The example FSM works as follows:

(1) Input * State → New state;

(2) Input * State → Output.

Finite State Machine(FSM) algorithm

```
% Finitestate( Input , State , NewState , Action).
  Finitestate(spacefree,active,active,moveahead).
  Finitestate(spacefree,turned,active,nothing).
  Finitestate(spacefree,static,active,nothing).
  Finitestate(spaceblocked,active,turned,turnleft).
  Finitestate(spaceblocked,turned,static,turnleft).
  Finitestate(spaceblocked,static,static,turnleft).

% Meaning for each of the rule as follows
  IF SpaceAhead(Free) AND State(Active)
           THEN Action(MoveAhead)
  IF SpaceAhead(Free) AND State(Turned)
           THEN State(Active) AND Action(Nothing)
  IF SpaceAhead(Free) AND State(Static)
           THEN State(Active) AND Action(Nothing)
  IF SpaceAhead(Blocked) AND State(Active)
           THEN State(Turned) AND Action(TurnLeft)
  IF SpaceAhead(Blocked) AND State(Turned)
           THEN State(Static) AND Action(TurnLeft)
  IF SpaceAhead(Blocked) AND State(Static)
           THEN Action(TurnLeft).
```

Algorithm 6.1 Finite State Machine (FSM)

Finite state machines output is mapped onto the agent action. FSM rules make use of the external and internal state conditions. Finite state machine rules can be framed for each piece of resource. For example Fsm-Ore, Fsm-Crystal, Fsm-Medicine, Fsm- Fungus, etc. As Figure 6.5 depicts, agents will make decisions and take actions based on the given environmental rules. A reflexive agent (refer Algorithm 6.2) uses any of the four control mechanisms to select the next move.

The current work make use of four reflexive behaviours: (1) first rule(Rx1) uses finite state machine, and moves arbitrary; (2) second rule (Rx2) uses finite state machine, and moves randomly;(3) third rule (Rx3) uses FSM, and move towards centre of the environment; and (4) fourth rule (Rx4) uses finite state machine, and moves towards edges of the environment.

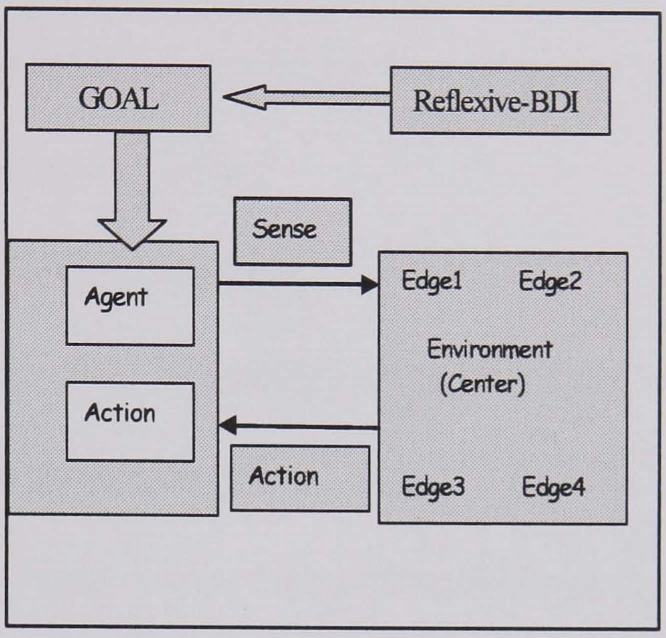


Figure 6.5 Reflexive Agent Structure

The reflexive actions or goals satisfy as those specified by the deliberative agents (described in the next section). The design for the fungus testbed includes four different reflexive agents. A BDI agent in deliberative layer determines which of the reflexive control mechanisms are active according to the goals the entire architecture attempts to satisfy. These goals determines the number of different types of reflexive behaviours required for this specific testbed (reflexive-BDI model is described in the next section). Reflexive agents understand the environment sensors, such as the locations of each edge or centre point, etc. For each move, they check the corresponding adjacent positions and determine the proper direction, either up, down, left or right. The tasks of such agents are to navigate environment and avoid collisions with other agents in the environment.

Reflexive agent design algorithm

Rx1: Uses FSM and up|left|right|Down (1st condition):-

Prefers move > nothing,

Prefers up | left| right | down (arbitrary).

Rx2: Uses FSM and up|left|right|down(2nd condition):-

Prefers move > nothing,

Prefers Random direction.

Rx3: Uses FSM X up|left|right|down (3rd condition):-

Prefers move nothing,

Prefers move towards centre of the environment.

Rx4: Uses FSM X up|left|right|down(4th condition):-

Prefers move nothing,

Prefers move towards edge of the environment.

Algorithm 6.2 Reflexive agent

Figure 6.5 shows the simplification of reflexive agent. The difference between them can be seen in Algorithm 6.2. The nature of reflexive agents is described in terms of the controlling finite state machine (FSM).

6.4 Reactive level

Reactive agents compromise the second layer of the distributed cognitive architecture shown in Figure 6.6. Reactive agents are designed to perform goal oriented behaviour, building on the mechanism of the reflexive agents described in the previous section. The goals they attempt to satisfy as those specified by the deliberative BDI agents (described in the next section). The design for the fungus testbed includes seven different reactive agents. The deliberative BDI determines which of the reactive control mechanisms are active according to the goals the entire architecture attempts to satisfy. These goals are either task related or agent-internal resource related, and determine the number of different types of reactive agent required for this specific testbed.

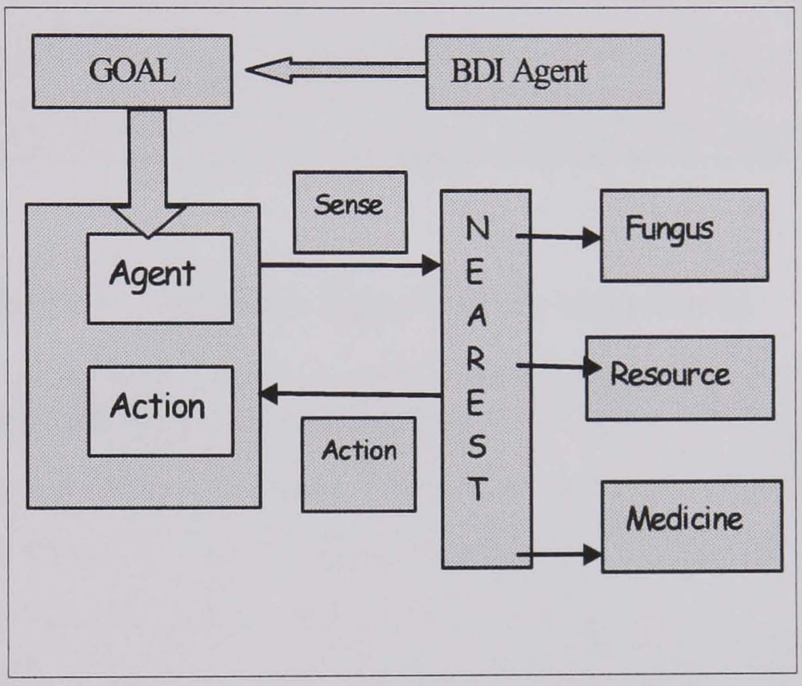


Figure 6.6 Reactive Agent

Table 6.1 shows the seven different reactive control mechanisms possible for a reactive agent in this research. Reactive actions are initiated by control mechanisms in response to the state of the agent and resources in the environment. Reactive agents respond to perceptual input according to move towards any unit: fungus, ore, golden ore, crystal or medicine. The algorithm used to control the agent varies according to the agent parameters. The reactive agent makes use of the generic perceptual mechanism used across all the agents in Figure 6.6, and depending upon its perceptual range returns a type, distance tuple for objects it can sense (i.e. agents, fungus, ore, crystal, medicine).

agent name	move towards nearest	collects (goal)
(r1)reactive-fungus	fungus	fungus
(r2)reactive-ore	ore	ore
(r3)reactive-golden_ore	golden-ore	golden-ore
(r4)reactive-crystal	crystal	crystal
(r5)reactive-medicine	medicine	medicine
(r6)reactive-resource	resource	resource
(r7)reactive-unit	any unit	any unit

Table 6.1 Reactive Agents (seven behaviours)

Resource Reactive algorithm

Goal based behaviour towards resource

Goal : one of (ore, golden ore and crystal)

Find the nearest resource by their distance,

Select the direction towards nearest resource,

Move towards resource direction (Left| Right | Up | Down).

If No Resource within Perceptual Range follows reflexive actions (i.e Move towards edges of the environment).

Algorithm 6.3 Algorithm for Reactive Fungus Agent.

This type of agent understands the parameters affecting its behaviour selection such as distance to resource, resource type, etc. The algorithm for the generic resource collector agent is shown in Algorithm 6.3. This agent collects ore, golden ore and crystal or resource. Reactive-fungus type agents can find the nearest fungus available in the environment, move towards it and collect it. If the resource required for the reactive behaviour is not sensed the agents resort to their default reflexive behaviour.

Similarly, small changes to this uncomplicated algorithm are present for all the reactive agents listed in Table 6.1; the designs of all types of reactive agents are included in Appendix A. The reactive-ore type agents can find and move towards nearest ore available in the environment, and collect it. The difference between the each unit, from r1 to r7 can be seen in Table 6.1.

6.5 Deliberative agents

Deliberative agents compromise the third layer of the distributed cognitive architecture shown in Figure 6.7. The design of deliberation mechanisms for the fungus testbed includes five different types of BDI agents. The BDI determines which of the reactive or reflexive control mechanisms are active according to the goals of the entire architecture attempts to satisfy. These goals are either task related or agent-internal resource related, and determine the number of different types of reflexive and reactive agent required for this specific testbed. For example, as in Figure 6.7 depicts BDI-Ore (BDI1) selects and controls the combination of reactive-fungus, reactive-ore, reactive-golden-ore and reactive-medicine behaviours. BDI5 or BDI-Reflexive agent selects and controls the combination set of reactive-fungus, reactive-medicine and reflexive behaviours. The different versions of deliberative models uses in this experiment are: BDI-Ore (BDI1), BDI-Crystal (BDI2), BDI-ore-and-crystal (BDI3), BDI-adaptive (BDI4); and BDI-Reflexive (BDI5).

For example consider a scenario of hungry agent in a fungus world testbed. The agent intends to collect ore. If the agent in a hunger state (energy level is less than threshold or predicted energy value) or high metabolism condition, then agent changes their desire towards fungus or medicine. Based on the agents needs and cost function, different deliberative agents can be framed. The difference between each BDI model in terms of energy level, biochemical and in terms of goal can be seen in Table 6.2. (Appendix A gives details of the entire BDI models).

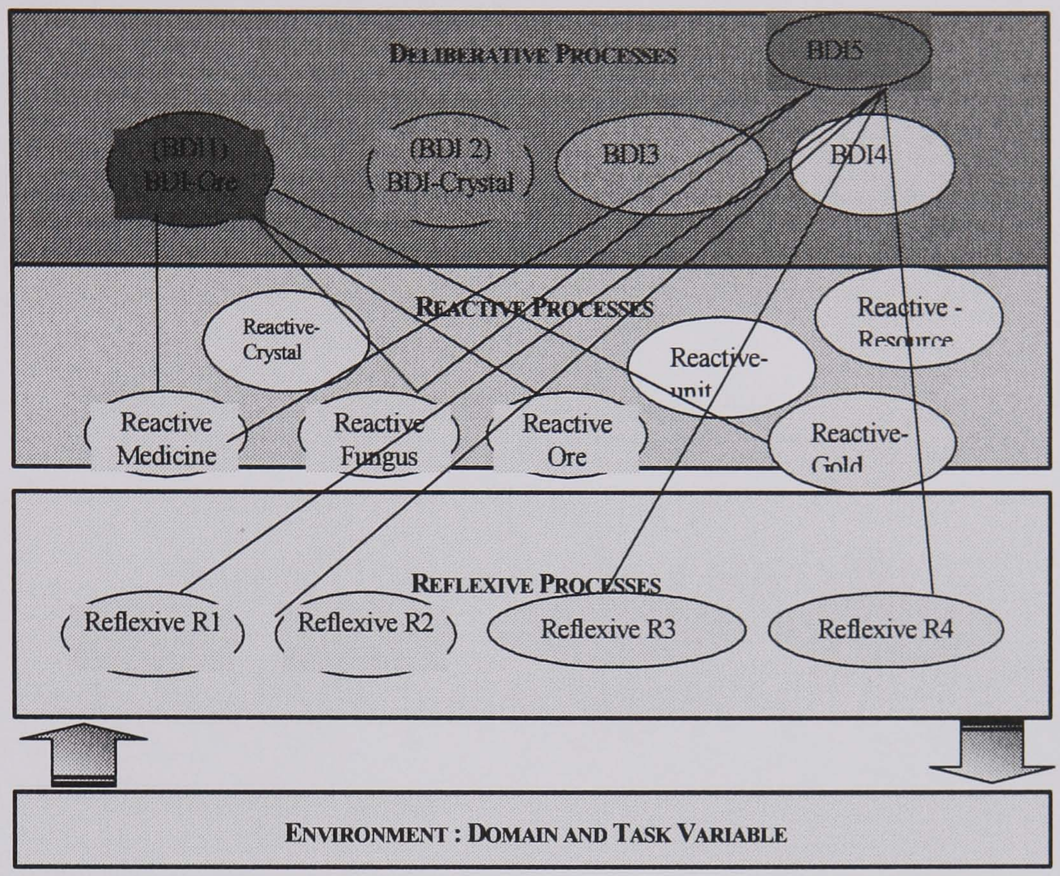


Figure 6.7 Deliberative agents and their control

First, it maps internal states onto a belief set from the perceptual range or perceptual level. This increases the agent's belief set for sensing in an environment. Second, the agent updates the belief set with perceptions and perceptual range. Thirdly, it uses the belief set to select the appropriate desire set. Fourth, the agent uses the desire set to select an intention. Finally, it selects the appropriate BDI model. The belief set includes the complete knowledge resources available in the surrounding environment. BDI models are capable of reasoning about their own internal tasks and plans.

Model	Energy resources		Biochemical	Goal based
BDI-Ore (BDI1)	40	Not applicable	Fungus (food) and Medicine (Metabolism)	Ore
BDI-Crystal (BDI2)	Not applicable	One level ahead	Fungus (food) and Medicine (Metabolism).	Crystal
BDI-Ore-Crystal	50	Not applicable	Fungus (food) and Medicine (Metabolism).	Ore and Crystal
	Not applicable	Thinks two levels further	Fungus (food) and Medicine (Metabolism).	Any unit
BDI-reflexive	Not applicable	Thinks three levels further	Fungus (food) and Medicine (Metabolism).	Reflexive

Table 6.2 BDI agents Affect and Goals

Finding First level smart energy model

Predict energy (Finding the decision boundary):-

Find the nearest fungus,

Find the nearest distance between agent and Fungus,

Find the agent needed energy by using their metabolism state and

Energy required, Predict energy is Distance/20 * Energy Use.

Algorithm 6.4 Smart Energy Level Model

Deliberative agents in a fungus world testbed are capable of performing different tasks. BDI agent follows the reactive actions in each move based on given rules and decision variables. Some BDI models favour specific goals towards: (1) ore; (2) crystal; (3) medicine, or (4) fungus. BDI models work in terms of a fixed threshold and adaptable energy use.

A smarter BDI model thinks further ahead, so the agent has sufficient energy to collect ore and collect next fungus before running out of energy available(Algorithm 6.4). The design of all types of predicting energy models are included in Appendix A.

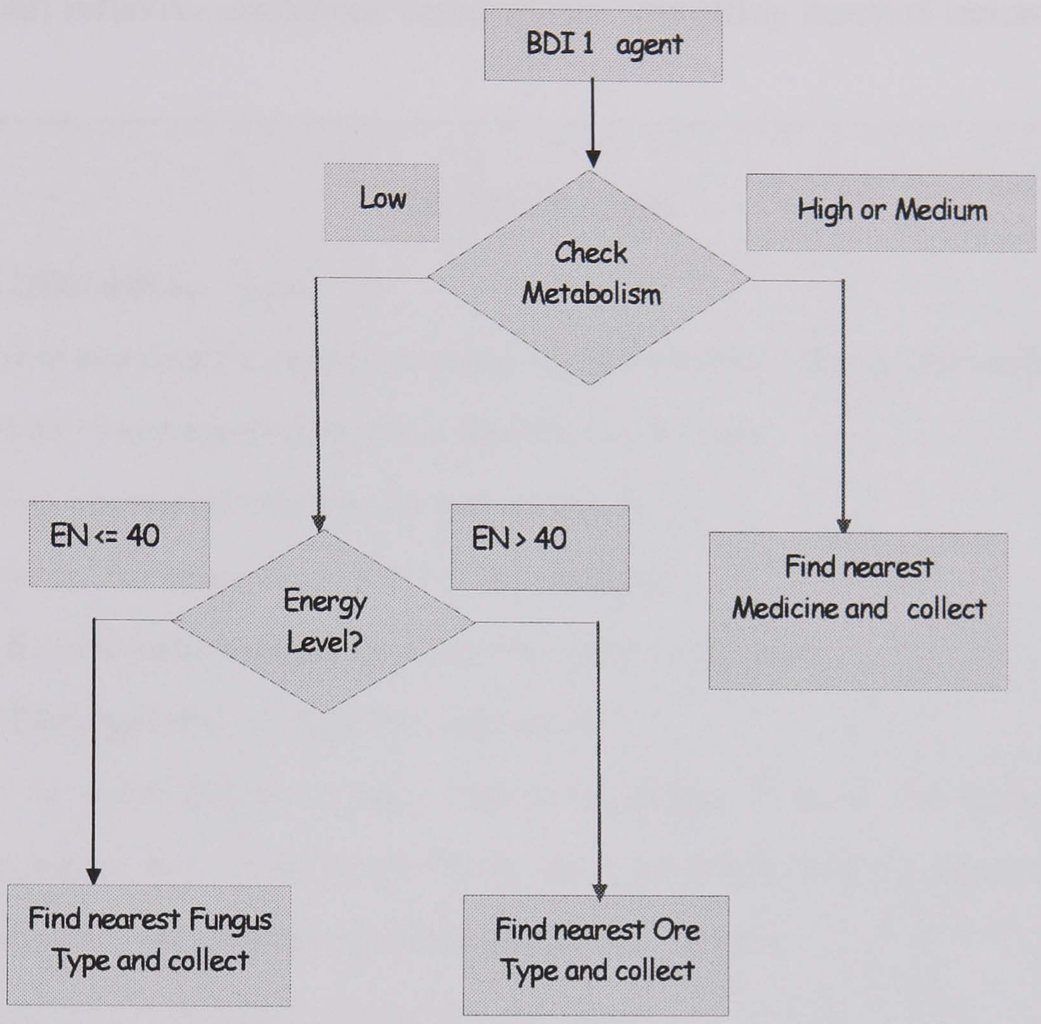


Figure 6.8 Design of BDI-Ore (BDI 1)

As in the Figure 6.8 and Algorithm 6.5 illustrates as follows: initially agent searches the nearest medicine to collect, and decreases their metabolism to low (see the metabolism effect in the testbed setup described in next Chapter). Second, the agent compares its energy level with the fixed energy value 40. If the energy level is more than predicted or threshold, then it moves towards ore (goal), based on cost and utility function (microeconomic level).

As in the Figure 6.9 illustrates as follows: initially agent searches the nearest medicine to collect, and decreases their metabolism to low (see the metabolism effect in the testbed setup described in next Chapter). Second, the agent compares its energy level with the predicted energy level (smarter energy). The energy required to survive and reach their goal. If the energy level is more than predicted energy level, then it moves (different goal) reflexive conditions based on cost and utility function (microeconomics)

BDI 1(First) model

(1)Metabolism > Low,

Then searches the nearest medicine to collect to lower the metabolism by their reactive mechanism. Uses the Reactive Medicine,

Find the nearest Medicine by their distance,

Select the direction towards nearest Medicine,

Move towards Medicine direction | left| right |Up| down.

(2)Energy Level <= 40 (Threshold value)

The agent desire to move towards to fungus to avoid the hunger condition or their death (Physiological oriented) uses the Reactive Fungus, Findes the nearest Fungus by distance formula,

Select the direction towards nearest fungus,

Move towards Fungus type direction | left| right |Up| down.

(3)Energy Level > 40 (Threshold value)

Reactive Ore (Goal based behaviour move towards nearest Ore)

Find the nearest Ore

Select the direction towards Ore.

Move towards Resource direction | left| right |Up| down.

Algorithm 6.5 BDI-Ore

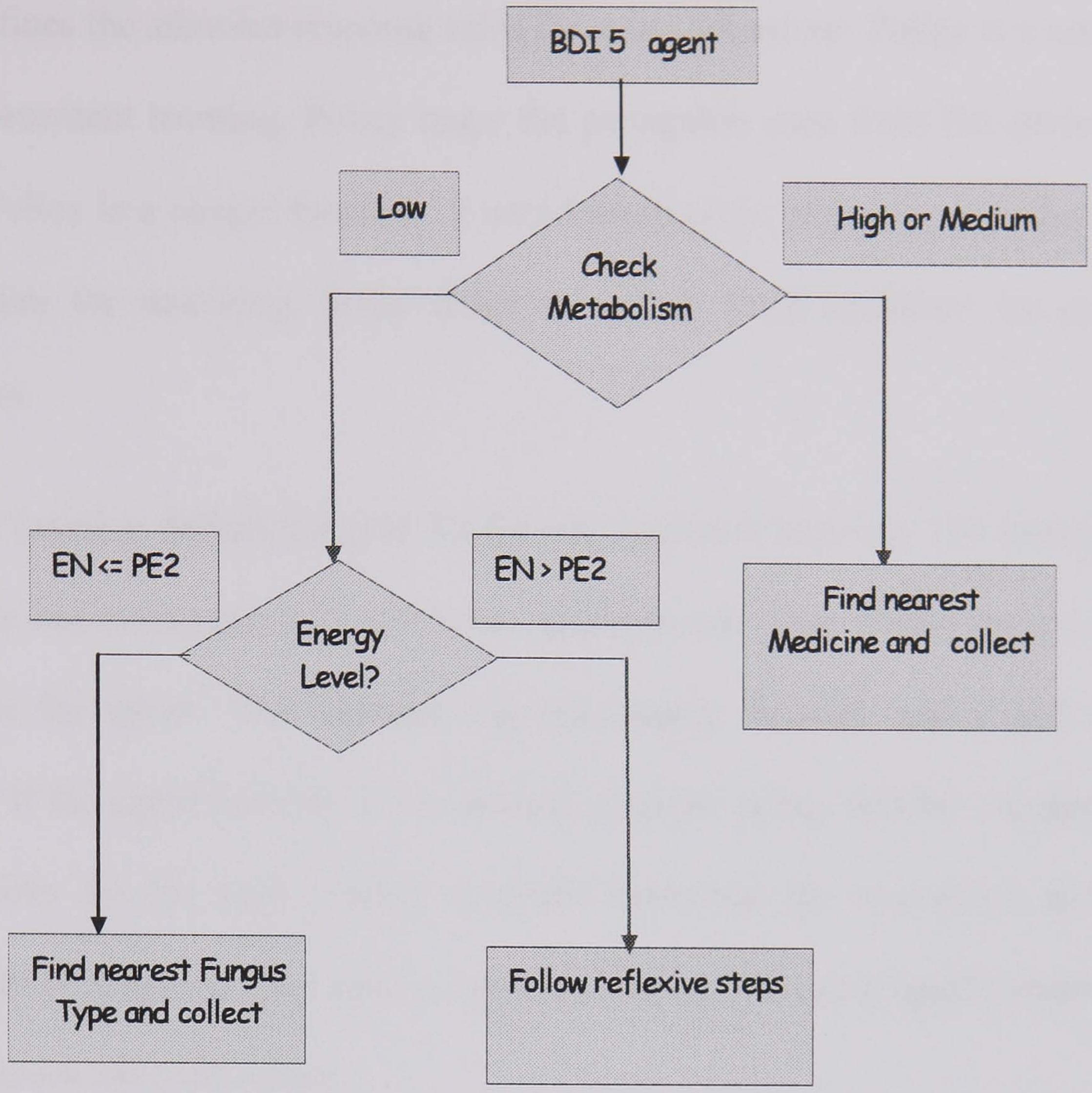


Figure 6.9 Design of BDI-Reflexive (BDI 5)

6.6 Learning Layer

The fourth layer of the architecture is the learning processes layer. Learning changes decision making at one level about actions at another level for tasks defined at yet a further level. This layer is in effect controlled through connections to the metacontrol level. Reinforcement learning calculates based on what and how to map situations to action for maximizing a reward. Q-Learning mechanism finds or tries to find a maximum reward for an action.

Reinforcement learning systems (refer Figure 6.10) In between the agent and environment, there are four main sub elements: (1) policy; (2) reward function; (3) value function; and (4) models.

Policy defines the stimulus-response rules for agent behaviour. Policy is a core element for reinforcement learning. Policy maps the perception state from the environment to action. Policy is a simple function. It uses lookup table, otherwise it makes extensive computation for searching. Some times the policy itself sufficient for determining behaviours.

A reward function defines the goal for the reinforcement learning. The reward function maps state and action pair [Q(s, a)] to the single reward. This defines the good and bad events for the agent. For example the relationship between policy and reward as follows: If the agent's action is low reward, then the policy will be changed to other, and it looks for the high reward. Rewards determine the immediate and intrinsic desirability of environmental states. Any reinforcement learning agent's main objective is to maximize the total reward.

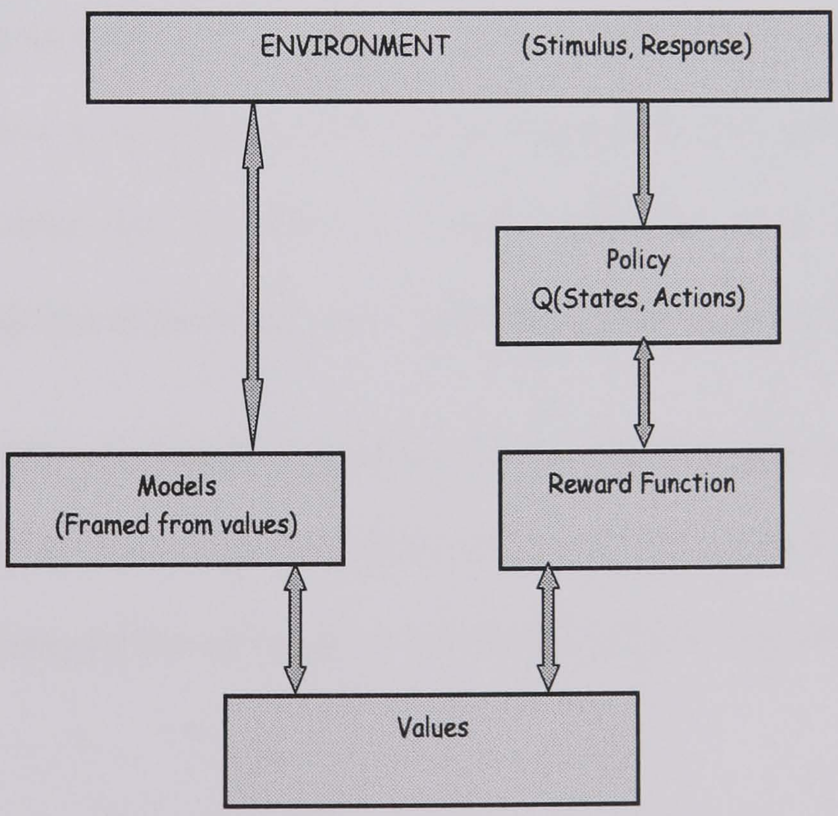


Figure 6.10 Q-Learning Mechanism

The values are predictions of rewards. If there is no rewards their would be no values. Judgment of changing an action is made based on the value. Rewards are easier to calculate then the values. The values are re-estimated and calculated based on the sequence of actions agent made on over its life time. Reinforcement learning algorithm is efficient for estimating the values. The value function decides how to maximize the total rewards. These algorithms try to finds the optimal value function through the iterations.

The final element is a model. Models are used for planning. This will be considered for the future situations. The model of the environment defines how the state and action can occur in the different location of the environment. The best actions for the agent can be learned by trial and error (Sutton, 2004). Models are used for dynamic programming (Sutton and Barto, 1998; Kaelbling et al., 1996).

Q-learning algorithms work by estimating the values of state-action pairs. The value Q(s, a) is defined as the expected discounted sum of future payoffs. This can be obtained by taking an action a from state s. Given the delta value from the current state s, selecting an action a, will cause receipt of an immediate goal unit and arrival at the next move. The rules can be symbolic, fuzzy, neural or other, depending on the direction taken in devising the metacontrol and metacognition part of the architecture.

Algorithm 6.6 explains the interaction between agent and environment with reference to Q-learning. Let Q(s,a) be the expected discount of reinforcement of taking an action in state s, then continuing by choosing actions optimally (McFarland, 1993; Bosser, 1993).

Q-Learning Algorithm:-

Let Q(s,a) be the expected discount of reinforcement of taking action an in state s, then continuing by choosing actions optimally.(McFarland, 1993; Bosser, 1993).

1. Initialize a table f with states S, actions A and the Q (utility or reward) value estimates.

2. Select an action a (where $a \in A$) and execute it.

3. Observe the immediate reward r,Reward defined using some agent relation, for example distance to desired object. Observe the new state s', achieved by action a on state s, where $a \in A$ and $s \in S$.

4. Update the table entry for Q value using an appropriate rule, for example

New(s, a) = Old(s, a) + (r(s) - r(s'))/ r(s).The Q values are nearly converged to their optimal values

5. Update the state: $s \rightarrow s'$.

6. Repeat from 2 until learning finished.

Algorithm 6.6 Learning Agent Design for Fungus world

The delta value is calculated from agent's distance and new distance values:

Delta is 1 / (Distance + 1);

and

Delta is OldQ + ((Distance - New distance)/Distance).

Afterwards the q value for each direction is assigned. For example, Xval = 660, Yval = 220, NewDir = down, and Delta = 0.00439997. From the new direction, the new locations are calculated. For example. Q([660,220],right), gives s'=[680,220]; Q([660, 220],left), gives s'=[640,220]; Q[660,220],up), gives s'=[660,200]; and Q([660, 220], down), gives s'=[660, 240].Agent selects the biggest q value location, for selecting a new direction.

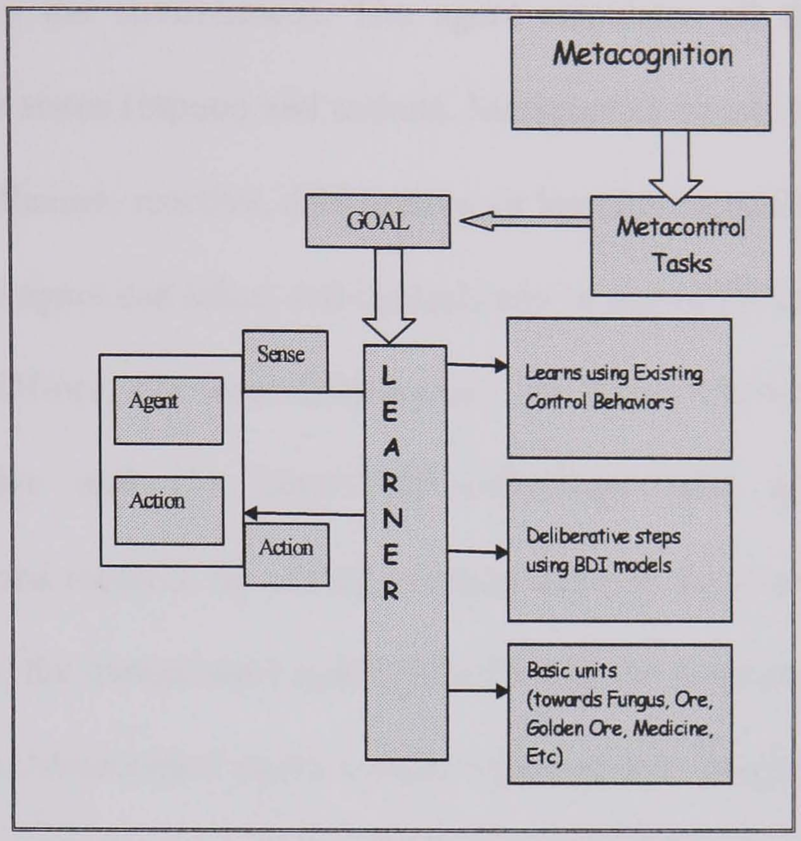


Figure 6.11 Learner

The learning agent (refer Figure 6.11), in effect, changes the goals and deliberative steps according to given rules. This may change to move towards fungus, ore, crystal and medicine at basic reflexive and reactive levels. Learning also can be applied in higher level layers. The learning mechanism can follow according to rules framed in the deliberative, metacontrol and metacognition levels. The metacontrol mechanisms can be viewed in terms of which the agents use existing controllers, learn behaviours (i.e. existing Q(s, a) values) or learn new behaviours by training the agents.

6.7 Metacontrol Level

Metacontrol agent decides which deliberative agents are to be learned and ready to perform in different conditions. The deliberative actions are called control actions. A meta controller determines the relevant control actions.Metacontrol agent's compromises the fifth layer of SMCA as shown in Figure 6.12. The metacontrol agent learns actions upon the environment. The agent calculates all the combinations of deliberative agent's states (inputs) and actions. Metacontrol agents have different levels of skills, such as reflexive, reactive, deliberative, or learning capabilities. As Figure 6.12 depicts metacontrol agent can select and controls any of one of the decision models such as :(1) learned-BDI-ore, (2) learn-BDI-crystal, (3) learned-BDI-ore and crystal, (4) learned-BDI-adaptive and (5) learned-BDI-reflexives. BDI agents should learn themselves by trained method. So adding learning methodology makes more effective. Reward is a goal of the metacontrol agent. This defines the good and bad events for the selected BDI agent. Metacontrol agent's main objective is to maximize the total reward of the running BDI agent. The metacontrol level may be a neural or some neuro-symbolic hybrid, and that allows learning. These rules can be used by the metacontrol part of the SMCA architecture. The rules can be symbolic, fuzzy, neural or other depending on the direction taken in devising the metacontrol part of the architecture.

The metacontrol task level agent does follows :(1) when and what to learn; (2) what decision model to select and (3) when to change between architecture possibilities. Metacontrol agents can select the BDI model. But cannot reason for higher level thoughts. Due to this reason metacontrol agent can not reason and change the BDI models.

Performance (OreBDI) = Ore + Golden_ore + Age

Affect (OreBDI) = Norm (OreBDI)/ Performance (OreBDI)

Performance (Crystal BDI) = Crystal + Age

Affect (Crystal) = Norm (Crystal)/ Performance (Crystal)

Performance (OreCrystalBDI) = Ore + Golden_ore + Crystal + Age

Affect (OreCrystalBDI) = Norm (OreCrystalBDI) / Performance (OreCrystalBDI).

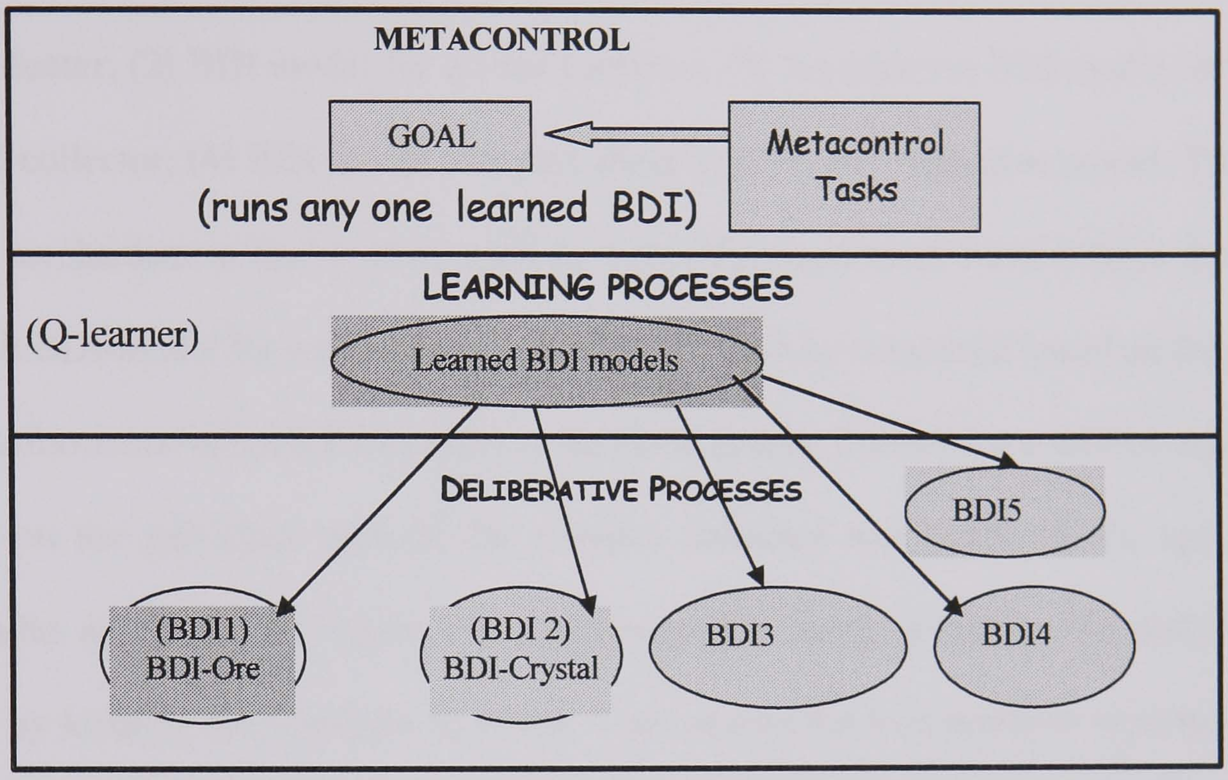


Figure 6.12 Metacontrol Task

6.8 Metacognition Level

This is the final layer of SMCA mind model. This layer uses norms to control the architecture. Metacontrol agents can choose BDI models, but cannot change the deliberative models with reasoning. The metacognition level agent works by comparing the architectural level, and uses (1) learned-BDI-ore, (2) learn-BDI-crystal, (3) learned-BDI-ore-crystal, (4) learned-adaptive and (5) learned-BDI-reflexive. The metacognition agents can change the framework of BDI agents with reasoning. This level works to control and monitor the deliberative models. The deliberative models can be switched off or on based on the norms.

Metacognition (refer Algorithm 6.7) technique in fungus world is decomposed into the different actions as given above algorithm. Each of these actions demonstrates a few specific types of metacognitive tasks. It uses different BDI models: (1) A BDI model for ore collector; (2) BDI model for crystal collector; (3) An adaptive BDI model for ore or crystal collector; (4) BDI model that plan ahead and (5) BDI-reflexive model. This uses affect mechanism to find a need, such as metabolism and food. Next it finds the affect of each BDI-model for each running cycle. Affect will be calculated based on the norms and performance of each BDI model in the environment. Performance will be calculated based on the particular type of the resource collected and based on the age of the particular agent. As in Figure 6.13, metacognition agent compares the architectural level, by keeping all situations in count. Then selects the best possible (optimal) BDI-model for each cycle to achieve the goal efficiently and increase the performance.

Norms controls the cognitive components or BDI models. Norms are responsible for figuring out when and how to execute a metacontrol task and then making sure that the task or set of tasks are done correctly. Metacomponents such as Norms, Affect and higher level rules are reason about the action, reflect upon that reasoning, and assess cognitive activity with respect to meatacontrol task. Given a loaded norm for resources (such as ore, golden-ore and crystal), energy decision boundaries, affect values for medicine, fungus, ore, crystal or ore and crystal. This can be used to decide upon the DESIRE (i.e. when and what resource to collect).

Given a loaded norm for ore, crystal, and one for energy decision boundaries. The Affect values for medicine, fungus, ore, crystal or ore and crystal can be used to decide upon the desire (i.e. what resource to collect). These are the values for the Q-learner to update. These are the values used to decide, for which agent to use for which desire.

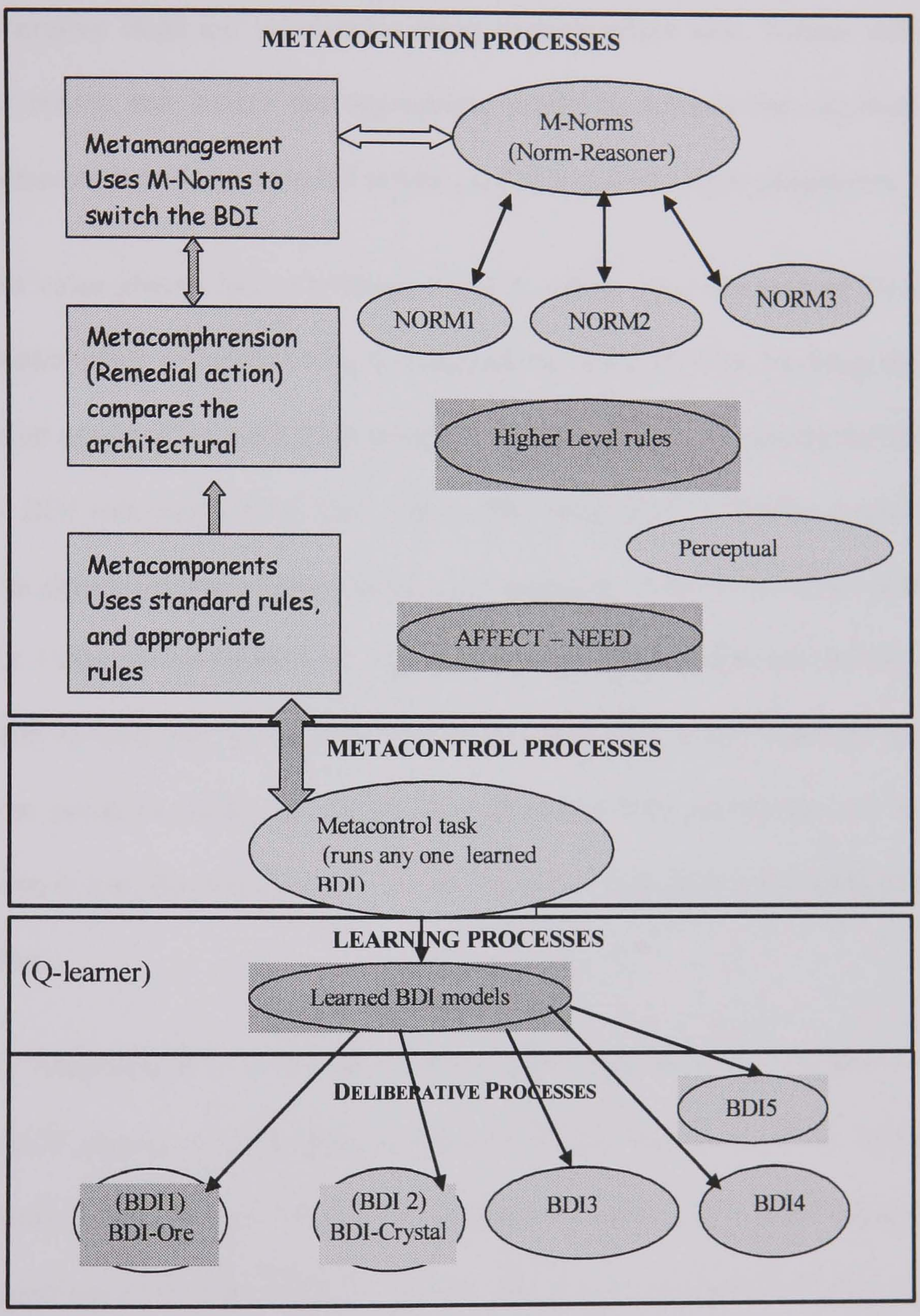


Figure 6.13 Metacognition layer in SMCA

By combining all DESIRE⊗INTENTION from affect values, the agent can choose a goal and the means to achieve it. A different metacomponents such as norm or multi norms, affect, perceptual range, and higher level rules may choose the highest affect goal and then the best (highest affect) method to achieve it.

The deliberative steps are verified by using metacognitive aids. Norms will find the particular reason and switch the appropriate model in a particular situation. This is called metacomphrension (remedial action), and is a part of metamanagement.

The affect value always lies in between 0 and 1. Affect value is assigned for each BDI model (Refer affect Algorithm 6.9). It compares the affect needed, by using the smallest affect value assigned for each BDI model. If the affect value of ore-crystal BDI is less than ore- BDI and crystal-BDI, then it uses ore-crystal BDI. A similar mechanism will be used to determine other affect values. For example, if the metabolism is high, then the affect value of medicine is 1. Agent needs to collect medicine and reduce their metabolism to medium. If the metabolism is medium, the affect value of medicine is 0.5. Agent needs to collect the medicine and reduce their metabolism to low. If the metabolism is low, the affect value of medicine is 0. Agent does not need to reduce their metabolism.

As in the Algorithm 6.9, if the energy level is less than decision boundary, the affect value is 0.75. Agent needs to collect energy (fungus) and reduce their hunger. If the energy level is greater than or equal to decision boundary, the affect value is 0. This means agent has sufficient energy to survive. If the total collection of ore is equal to 0, norm of Ore-BDI is 0.75.If the total collection of ore is more than zero; norm of particular BDI is equal to perceived ore divided by total collected ore. If the total collection of crystal is 0, norm of particular crystal-BDI is 0.75. If the total collection of crystal is more than zero; norm of Crystal- BDI is equal to perceived crystal divided by total collected crystal.

Metacognition agent general structure

Step1:-Map internal states onto Belief set from the perceptual range. The perceptual level increases the agent's belief set for sensing in the environment. Example level 5 returns Sense List = [5-spacefree, 4-agent, 3-spacefree, 2-fungus, and 1-spacefree]

Update Belief Set with perceptions and perceptual range.

Step2: -Use Affect mechanism to find a need of the metabolism and need of a food.

Step3:- Use metacomponents such as Norms or M-Norms (Such as Norm1, Norm2, Norm3, ETC are) to decide which BDI model to choose in write time by using write decision (optimal decision)by comparing resources available and balance the resources in a testbed.

Example Norm 1:

Collected ore is Ore1 + Golden_ore1,

Collected ore > 0,

Norm_oreBDI = Perceived ore / Collected ore.

Perceived ore = No_Ore + NO_Gold,

Norm_oreBDI = Perceived ore / Collected ore.

Collected crystal is Crystal1.

Step4:- (metacomphrension or remedial action) Select appropriate Belief-Desire-Intention combination (BDI-Ore, BDI-Crystal, BDI-Ore Crystal, ETC), by comparing the architectural results.

Step5:- (Metamanagement) Uses M-Norms to switch the BDI Models (Such as BDI-Ore, BDI-Crystal, BDI- Ore Crystal, ETC),

Step6:- (Schema training) Use Q-Learning for Optimal steps taken from agent by using M-Norms and Affect Mechanism (Metacognition level).

Step7:- Repeats the steps (Step1 to Step6) until Simulation ends.

Algorithm 6.7 Metacognition Agent Design for Fungus world

Norm 1

Collected (Ore) = 0,

Norm (OreBDI) = 0.75.

Collected (Ore) > 0,

Norm (OreBDI) = Perceived (Ore)/Collected (Ore).

Collected ore is Ore1 + Golden ore1, Collected (Crystal) = 0,

⇒ Norm (Crystal) = 0.75.

Collected (Crystal) > 0,

Norm (Crystal) = Perceived (Crystal)/Collected (Crystal).

(Collected (Ore) + Collected (Crystal) = 0,

Norm (Ore_CrystalBDI) = 1.

(Collected (Ore) + Collected (Crystal)) > 0,

Norm (OreCystalBDI) = Perceived (Cystal+Ore)/Collected (Cystal+Ore).

Algorithm 6.8 Norm 1

As similar to the above example (Algorithm 6.8), if the total collection of ore and crystal is 0, norm of ore-and-crystal-BDI is 0.75. If the total collection of ore and crystal is more than zero, norm of ore-crystal-BDI is equal to sum of perceived crystal and ore divided by sum of collected ore and crystal. There are different versions of norms based on the different circumstances and reasons. For each decision model, three different norms will be calculated based on their performance. Compare the norms each other to find out the best norm to use for that particular cycle in testbed.

For example, Algorithm 6.9 compares the norms. If the norm of the ore_crystal_BDI is greater than norm of the ore_crystal_BDI1 and ore_crystal_BDI2 then norm of the ore_crystal_BDI is selected. If ore_crystal_BDI1 is greater than norm of the ore_crystal_BDI and ore_crystal_BDI2. Norm of the ore_crystal_BDI1 is selected.

```
                         Affect rule1

If( Metabolism = Low),          Need(Medicine) = 0,
If (Metabolism = Medium),       Need(Medicine) = 0.5,
If( Metabolism = High),         Need(Medicine) = 1.0,
If( Energy_ level < Decision_boundry),
Need(Energy_level) = 0.75,
If(Energy _level >= Decision_Boundry),
Need(Energy_level) = 0,
Need(Medicine) = 0,
Need(Energy_level) = 0,
Performance_Ore_BDI is Ore1 + Golden_ore1 + Age1,
Norm_oreBDI > Norm_oreBDI1,
Norm_oreBDI > Norm_oreBDI2,
Affect_ore_BDI is Norm_oreBDI / Performance_Ore_BDI.
Performance_Ore_BDI is Ore1 + Golden_ore1 + Age1,
Norm_oreBDI1 > Norm_oreBDI,
Norm_oreBDI1 > Norm_oreBDI2,
Affect_ore_BDI is Norm_oreBDI1 / Performance_Ore_BDI.
Performance_Ore_BDI is Ore1 + Golden_ore1 + Age1,
Affect_ore_BDI is Norm_oreBDI2 / Performance_Ore_BDI.
```

Algorithm 6.9 Affect 1

If the ore_crystal_BDI2 is greater than norm of the ore_crystal_BDI and ore_crystal_BDI1, then norm of the ore_crystal_BDI2 is selected. Affect or need of the particular resource is calculated based on the norm of particular BDI and performance criteria. The relationship is as follows:

Affect_ore_BDI is Norm_oreBDI / Performance_Ore_BDI;

Performance_Ore_BDI is Ore1 + Golden_ore1 + Age1.

The perceptual range or perceptual level increases the agent's belief set for sensing in the environment. First, it maps internal states onto a belief set from the perceptual range or perceptual level. This increases the agent's belief set for sensing in an environment. See, for example, level 5: returns Sense List = [5-spacefree, 4-agent, 3-spacefree, 2-fungus, and 1-spacefree]. Second, the agent updates the belief set with perceptions and perceptual range (refer Appendix B).

6.9 Summary

This chapter summarizes the design part of SMCA using extended CAMAL cognitive architecture with extra processing layers. The distributed model of Mind as a "Society of Mind" design includes reflexive, reactive, deliberative level agents, BDI models, General structure of the BDI model and metacognition agent general structure. The design also includes about the metacomponents such as affect, higher level rules for resource set, norms and learning of metacognition steps. The design for metacognition or self-reflection architecture in one specific cognitive architecture. Metacognition is defined as thinking about thinking. It can be viewed as two ways monitoring a group of agents in an intelligent or cognitive or robotic architecture (i.e. self reflection) and making changes by adapting effective strategies in that society of agents, to constitute a "Society of Mind". Combination of agents work to achieve the different goals (cognitive tasks) in three different tasks: (1) collection of ore (2) collection of ore golden ore; and (3) collection of crystals can be performed by society of agents. The different combination of agents is organized for different goals or task.

Chapter 7 Design of Experimental Testbed

7.1 Testbed Setup

The fungus world testbed is implemented using SWI-Prolog 5.4.6 (SWI_Prolog, 2003). The fungus world testbed experiments include cognitive and engineering perspectives on the architecture described in the previous chapters. The fungus world environment has been created to have both dynamic and static (Figure 7.1). The static blocks are more flexible, to create a particular location of the environment. There are different parameters in the environment for an agent's biochemical engine and performance. Resource parameters in the environment are created through the checkbox consisting of: (1) standard fungus; (2) small fungus; (3) bad fungus; (4) ore; (5) golden ore; (6) crystal and (7) medicine. The agents are created in the environment by using Prolog graphics (Figure 7.1 and Figure 7.2). All of the parameters can be changed according to experimental requirements, and are defined in a configuration module. The agent cannot differentiate between standard fungus, small fungus, and bad fungus until it collects or eats them.

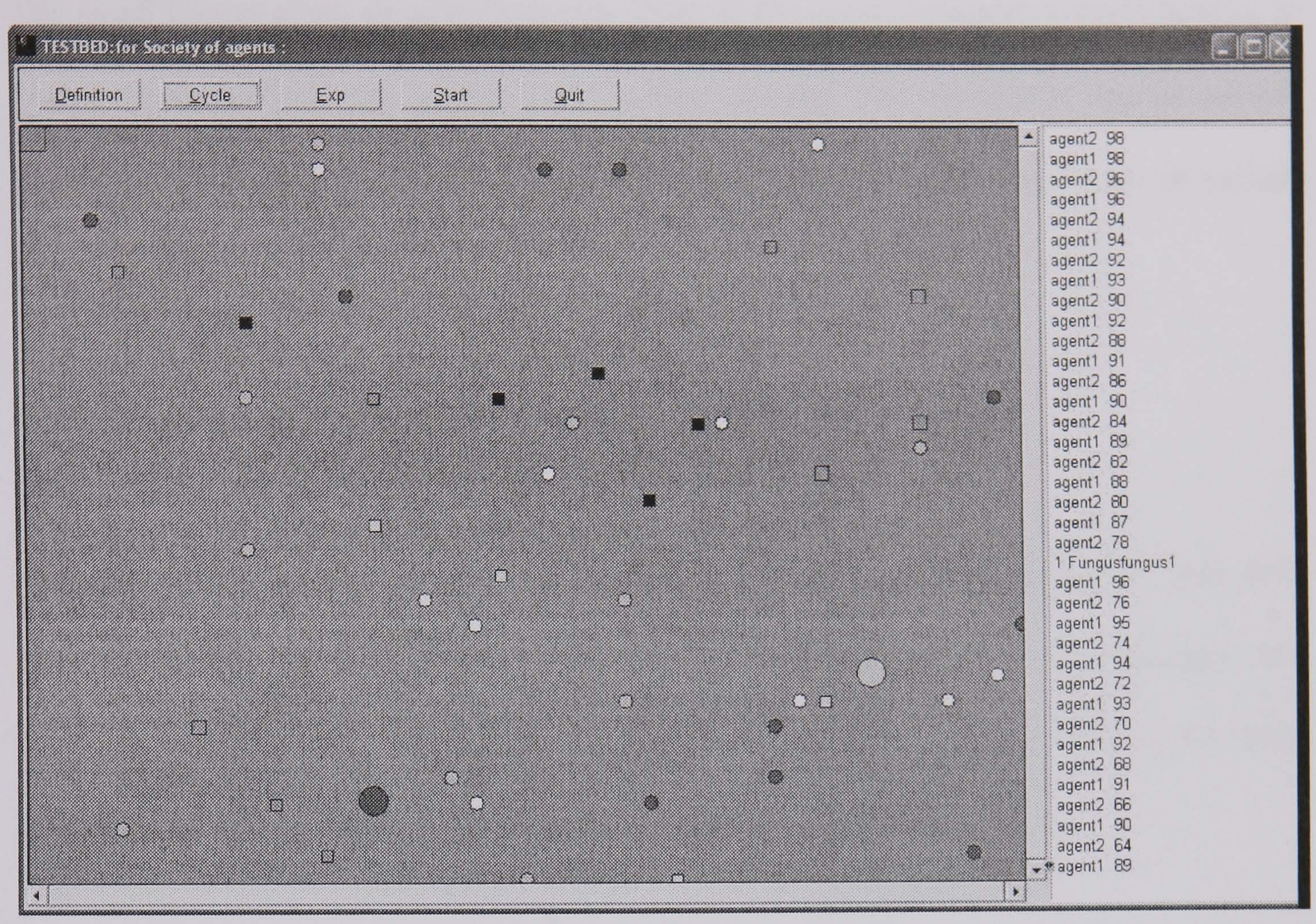


Figure 7.1 Fungus world Testbed

7.1.1 Standard Fungus

Fungus is a nutrient for the agents. Each standard fungus gives an agent 10 energy units. Initially, each agent has predetermined energy units. For each cycle, the agent consumes a fixed number of energy units. If the energy level (nutrients) reaches 0, the agent will die. All parameters are defined in the configuration module. These values can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.1.2 Small Fungus

The small fungus gives an agent 5 energy units. If the agent consumes a small fungus, 5 energy units (default) are added to the energy storage. This parameter can be varied simply by changing the values in the configuration module. The value can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.1.3 Bad Fungus

The bad fungus has 0 energy units. If the agent consumes bad fungus, it gets null energy. Moreover, bad fungus increases the metabolism rate, and changes the metabolism affect. This value can be varied from 0 to 150. Refer the Table 7.4 for their effects.

7.1.4 Ore

The collecting of ore is the ultimate goal of each agent. Each agent group tries to collect as much ore as possible in the environment. At the same time, an agent has to maintain the energy level necessary to live in the environment. Initially, collection is 0, and one value is added after collecting each piece of ore. This value can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.1.5 Golden Ore

Collection of golden ore increases the performance of an agent. One piece of golden ore is equal to five standard ore units. This value can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.1.6 Crystal

Collection of crystal increases the performance of agent by a factor that is double that of ore. This value can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.1.7 Medicine

The medicine affects the metabolism of the agent in the testbed. The collection of medicine decreases the metabolism. The metabolic effect is exactly opposite that of collection of bad fungus. This value can be varied from 0 to 150. Refer to Tables 7.1, 7.2, 7.3, and 7.4 for their effects.

7.2 Experimental setup

This environment supports the running of the various types of agents, where each agent uses a different type of rules and mechanisms. In these experiments, a maximum of 50 agents were defined.

The experiments were conducted for the same number of agents, the same type, the same number of fungi (including standard, small, and bad), ore (including standard and golden ore) and the same objects (including obstacles). The time scale and maximum cycles were kept constant by adding the same type of agent in each experiment.

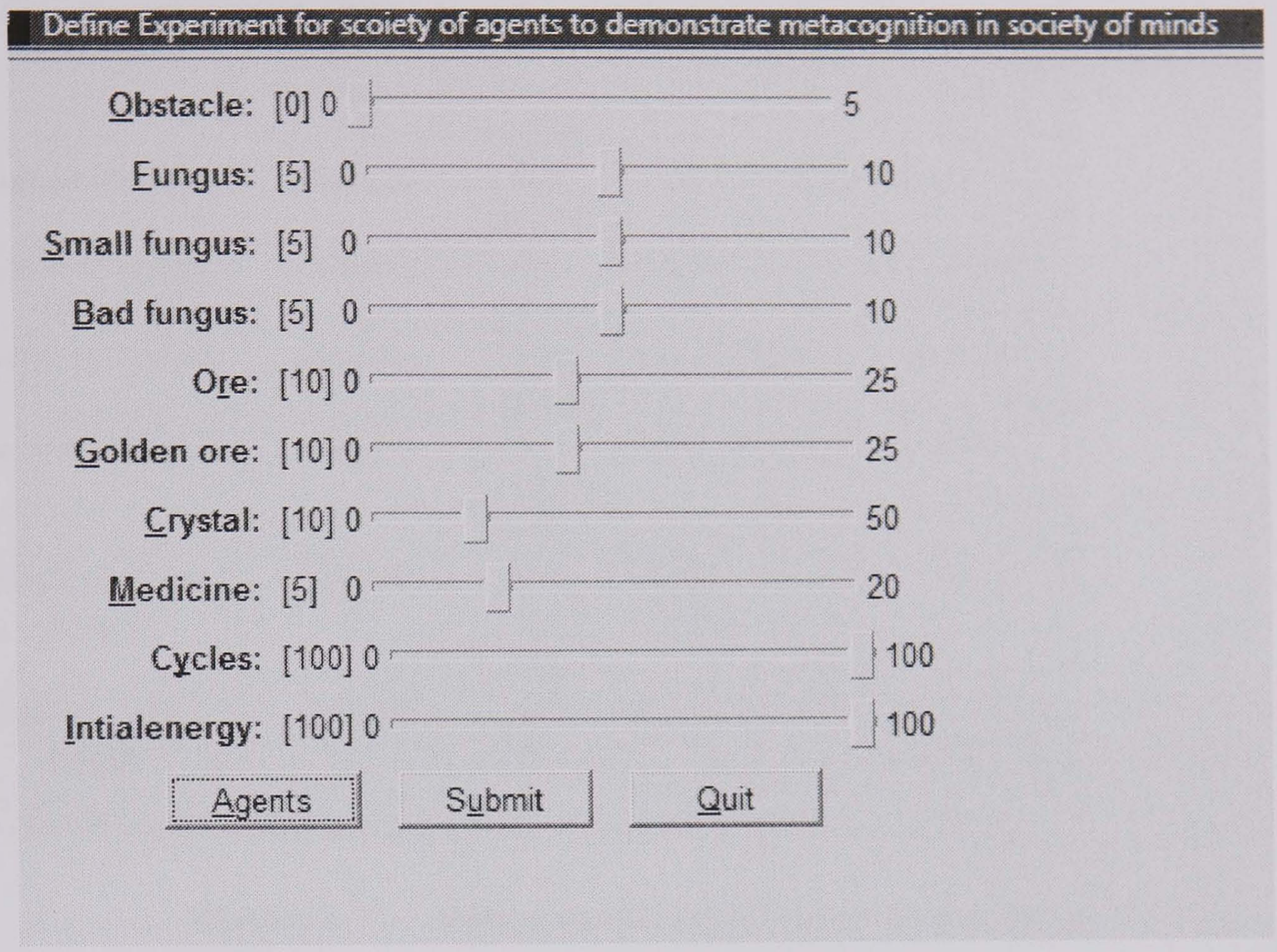


Figure 7.2 Parameters Selection Menu

The same analytical parameters were recorded in each study: energy left after their maximum cycles, ore collected, fungus consumption, and life expectancy of agents. Tables 7.1, 7.2, 7.3, and 7.4 gives brief explanations of experimental setup parameters used, including their types, values, and effects on the experiment.

Table 7.1 gives the details of actors present in the fungus world environment, including their type (numeric or atom), their assigned range of values (0 to n), and their default effects on the environment. For example, numeric value for "Number of Agents" is (0 to 50: 20). The range is 0 to 50, and default value is 20. All the parameters are similarly defined.

Parameter	Type	Value	Default Effect
Number of Agents	Numeric	0 to 50 : 20	Amount of Agents in testbed
Agent Type	Categorical: atom	type1 etc	Defines type of agent in environment
Obstacles	Categorical: atom	None, Static, random	Obstacles present or not
Number of Ore	Numeric	0 to 150: 20	Amount of Ore in Testbed
Number of Golden Ore	Numeric	0 to 150: 10	Amount of Golden Ore in Testbed
Number of Crystal	Numeric	0 to 150: 10	Amount of Crystal in Testbed
Number of Fungus	Numeric	0 to 150 : 20	Amount of Fungus in Testbed
Number of Small Fungus	Numeric	0 to 150 : 20	Amount of Small Fungus in Testbed
Number of Bad Fungus	Numeric	0 to 150 : 20	Amount of Bad Fungus in Testbed

Table 7.1 Parameter for fungus world environment.

7.3 Replenish (Refreshment) Rates

Table 7.2 gives details about the replenishment effects on the fungus world environment.

Replenish rate	Type	Value	Default Effect
Fungus	Numeric	5 : 2	2 fungus created on every 5 cycles
Small Fungus	Numeric	6: 2	6 small fungus created on every 2 cycles
Bad fungus	Numeric	7: 2	7 bad fungus created on every 2 cycles
Ore, Golden Ore, Crystal	Numeric	0 : 0	Null effect
Medicine	Numeric	5 : 2	5 medicine created on every 2 cycles

Table 7.2 Replenish (refreshment) rate

The replenish parameters of fungus, small fungus; bad fungus, and medicine are created with different rates in the environment. An example is "replenish rate of fungus type numeric," and the value (5:2) means that two fungus will be created on every five cycles. Similarly, for every six cycles, two pieces of small fungus will be created. For every seven cycles, two pieces of bad fungus will be created. The replenishing effect of medicine is that for every five cycles, two pieces of medicine will be created.

7.4 Agent Performance Parameters

Table 7.3 gives details about the effect of parameters on agent performance. Each move of the agent consumes some energy. The consumption of energy depends on the metabolic rate or metabolism of an individual agent. The effect of energy usage for each move of an agent is based on the metabolism. If the agent follows Low metabolism, energy usage is one unit per cycle. If the agent follows Medium metabolism, for each move, energy usage will be two units per cycle. For High metabolism, the agent consumes five units for each move. This can be changed, from the configuration module.

Table 7.3 also explains the energy storage level of agents when they consume different types of fungus. If the agent consumes standard fungus, the energy storage level of the agent increases by ten units. For small and bad fungus, consumption increases the energy storage levels by five and zero units, respectively. This also can be changed, from the configuration module. The last category in Table 7.3 is the effect of bad fungus and medicine on the agents. If the agent collects bad fungus, this increases the metabolism. If the agent collects the medicine, it decreases the metabolism.

Parameter	Type	Value	Default Effect
Metabolism	Categorical-atom	Low	Agents use Energy at 1 unit per cycle
Metabolism	Categorical-atom	Medium	Agents use energy at 2 units per Cycle
Metabolism	Categorical-atom	High	Agents use energy at 5 unit per cycle
Fungus	Object : Numeric	10	Increases energy level of agent by 10 units
Small Fungus	Object : Numeric	5	Increases energy level of agent by 5 units
Bad Fungus	Object: Numeric	0	Increases energy level of agent by 0 units
Bad Fungus	Object: categorical	N/A	Increases Metabolism Low to Medium Medium to High High to High
Medicine	Object: categorical	N/A	Decreases Metabolism Low to Low Medium to Low High to Medium

Table 7.3 Parameters Affecting Agent Performance.

7.5 Output Parameters

Table 7.4 gives details about the parameters affecting an agent's performance. The fungus, small fungus, and bad fungus increase the energy storage levels by ten, five, and zero units, respectively. If the agent collects one standard ore, it increases the performance by one unit. Collecting a golden ore increases performance by five times more than a standard ore. Collection of one piece of crystal increases the performance two units. Cycle is a categorical type, and the values are 1, 2, and 5, respectively, for agent usage of energy for each move, based on low, medium, and high metabolism effects.

Parameter	Type	Value	Default Effect
Fungus	Object : Numeric	10	Increases the energy level by 10 energy units, to live in the environment
Small Fungus	Object : Numeric	5	Increases the energy level by 5 energy units, to live in the environment
Bad Fungus	Object: Numeric	0	Increases the energy level by 0 energy units, to live in the environment Decreases the performance by Increasing metabolism
Ore	Numeric	1	Increases the Performance by 1.
Golden Ore	Numeric	5	Golden Ore increases the agent performance 5 times More than an ore.
Crystal	Numeric	2	Crystal Increases the agent Performance 2 Times more than a Ore.
Medicine	Object: Numeric	0	Increases the performance by Decreasing metabolism
ENP (Energy storage)	Object: Numeric	N/A	Stores the energy based on consumption of Fungus, Small Fungus, and Bad Fungus.
Cycle	Object: categorical	1 or 2 or 5 Energy units	Agent consumes the Energy
Low	Categorical-atom	1	Agents use energy at 1 unit per cycle
Medium	Categorical-atom	2	Agents use energy at 2 unit per cycle
High	Categorical-atom	5	Agents use energy at 5 unit per cycle

Table 7.4 Output Parameters Defining Agent Performance

7.6 Society of Agent's Setup in the Experiment.

The Society of Mind approach to cognition and metacognition in a cognitive architecture is divided into six tiers: reflexive, reactive, deliberative, learning metacontrol and metacognition level. Agents are distributed across different layers of architecture, to cover all processing and functioning associated with the adopted model of mind. The following cognition: reflexive (six behaviours), reactive (eight behaviours), and metacognition: deliberative (fifteen behaviours), learning (learning all given behaviours), metacontrol (complex) and metacognitive (complex) (society of agents) are set up in the experiment (refer agents selection menu Figure 7.3). (Refer section 5.4 for proposed cognition and metacognition agents in SMCA and design of agent's Chapter six).

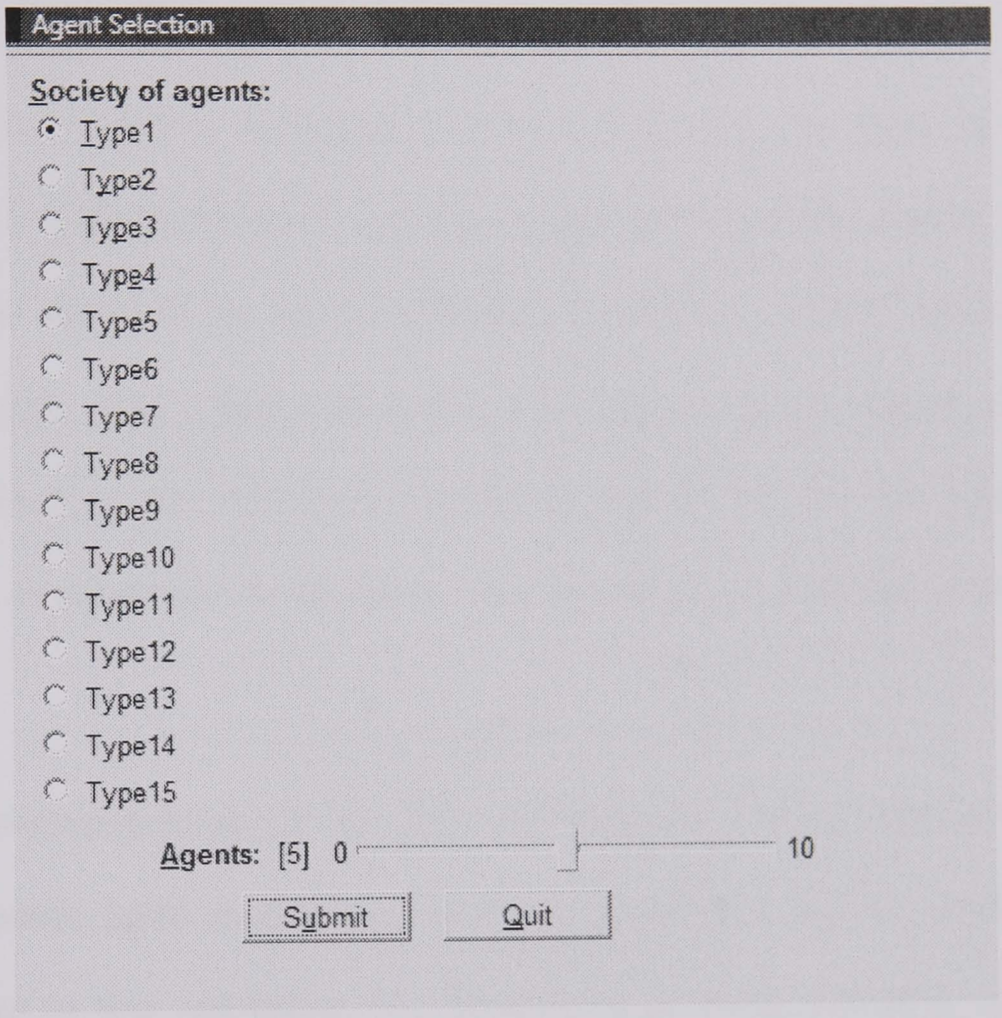


Figure 7.3 Society of Agents Selection Menu

7.7 General structure of the fungus world simulation

As shown in Figure 7.4, the fungus world testbed simulation can be mapped onto physically situated agents in cognitive architectures, to demonstrate "Society of Mind" (SMCA). Agents move and operate in an environment. At each turn, each agent performs two phases. First, the agent moves in an environment and checks the corresponding adjacent positions. Agents determine the regular (any one way) and random directions (up, down, left, and right). If the adjacent position is space free, then it moves to the next corresponding location. Secondly, agents check the parameters and rules. If the rule allows it to collect, it collects the parameter, or moves, based on the direction assigned. The general structure of the simulation can be sketched as follows. Initially (chosen by user), agents are randomly distributed in the environment.

Each agent's initial effort is determined based on its type; for most types of agents, initial effort is determined based on their action and behaviour. Agents' position and effort may be observed on the relative display window. In every round, each agent moves randomly or in a particular direction in order to meet a fungus, ore, crystal or medicine, based on the rules and regulations framed for an each agent. An agent perceives its environment by sensing and acting rationally upon that environment with its effectors.

The agent receives precepts one at a time, and maps this percept sequence into different actions. Dynamic agent morphology allows an ontogenetic process (metabolism) i.e., high, medium or low and aging, i.e., being born, growing, maturing aging, etc. The energy level determines the current hunger condition and thereby triggers eating (metabolism).The control architecture enables the fungus eaters to adapt to the dynamic environment.

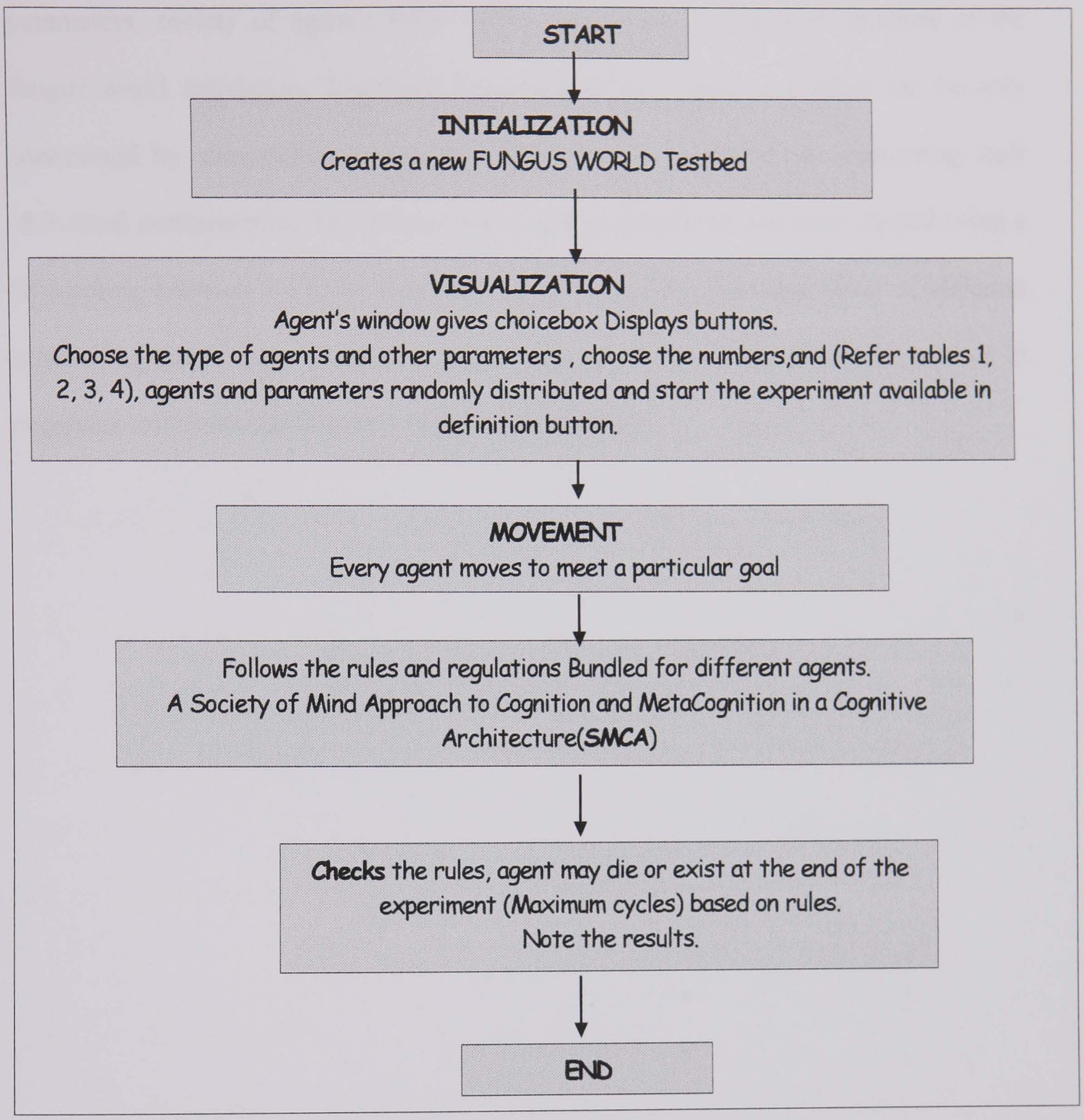


Figure 7.4 Simulation Flowchart (General Structure)

7.8 Summary

This chapter describes the design of the fungus world testbed and development undertaken enabling experimental setup. It gives the experimental setup with parameters for fungus world environment, replenish rates, agent performance parameters, output parameters, society of agent's setup in the experiment and general structure of the fungus world simulation. Intelligent behaviour of an animal or a robot can be only understood by competition among the different types of agents by comparing their individual performances. The fungus world testbed (platform) was been created using a Swi-prolog (version 5.4.6).To test and simulate qualities and capabilities of different types of agents (society of agents) and to demonstrate a Society of Mind approach to cognition and metacognition in a cognitive architecture.

Chapter 8 Experimentation Results

This chapter gives the details of the simulation experiment results using the society of agents. The solutions will demonstrate the effectiveness of a Society of Mind approach to cognition and metacognition in a cognitive architecture. The results of these experiments will provide the basis for solutions or partial solutions for the research issues raised in this thesis.

8.1 Testing Plan

Society of Mind Cognitive Architecture (SMCA) is designed in the perspective of principles of microeconomics in animal minds. Agent behaviours can be analysed using many different metrics. The major metrics are metabolic activity, competition and social interaction with respect to environment and microeconomics. The SMCA results are simulated and presented based on the two metrics. They are (1) fitness function, and (2) benefit or goal. Cost can be measured by considering the fitness of an animal over a period of time, where fitness is defined in terms of future expected reproductive success after this period. The cost function deals with real risks, real costs and the benefits. The simulation platform provides a simple way to study the complex interactions between different types of agents. This simulation demonstrates agent's behaviours with respect to the use of energy and time to makes decisions. The simulated result graphs compares agents performance based on the goal achievement and fitness function. Time scale is fixed for all the agents. Each agent will be experimented for the same time scale and same resources (refer Figure 8.1). For example type2 agents are running for the 25 cycles or time scale. Input value of each parameter is defined in the configuration file. The output file gives details of each agent. This file includes experiment length, experiment number and collection of each parameter. The results of each agent is systematically tested and calculated based on their fitness (life expectancy) and performance.

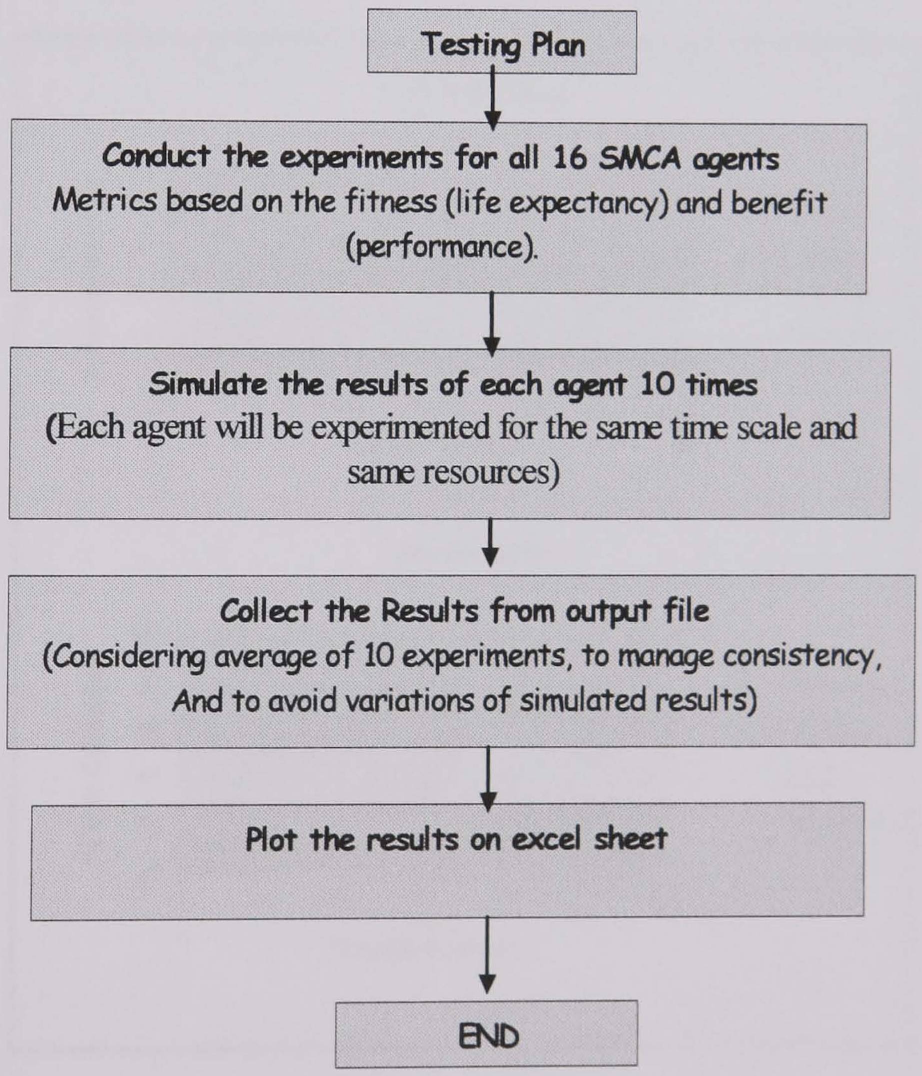
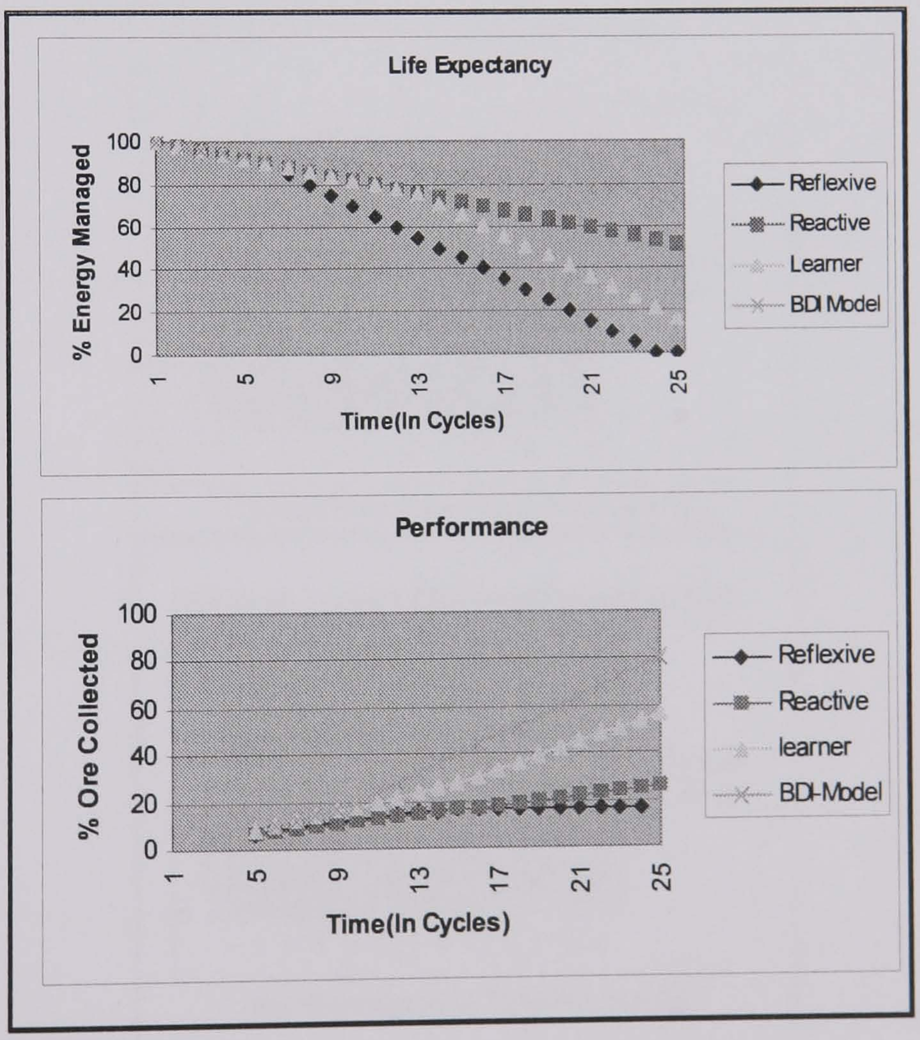


Figure 8.1 Testing Plan Chart

Experiments are conducted based on the assigned stastical data for each agent. To compare a result of each agent, the following statistics were collected: life expectancy, fungus consumption (including standard fungus, small fungus and bad fungus), ore (standard ore and golden ore), crystal collected and metabolism. The life expectancy or age of the agent is noted, along with the agent's death (or age after the end of the maximum cycles or time). The agent's total performance will be calculated by amount of resources (ore, golden ore and crystal) collected, and based on life expectancy. The simulations can be executed several times by considering the same input. The final result graphs are considered by taking an average of ten simulated experiments. The data will be plotted on the excel sheet in order to obtain result graphs.

8.2 Study One (Systematic comparison of multiple agents)



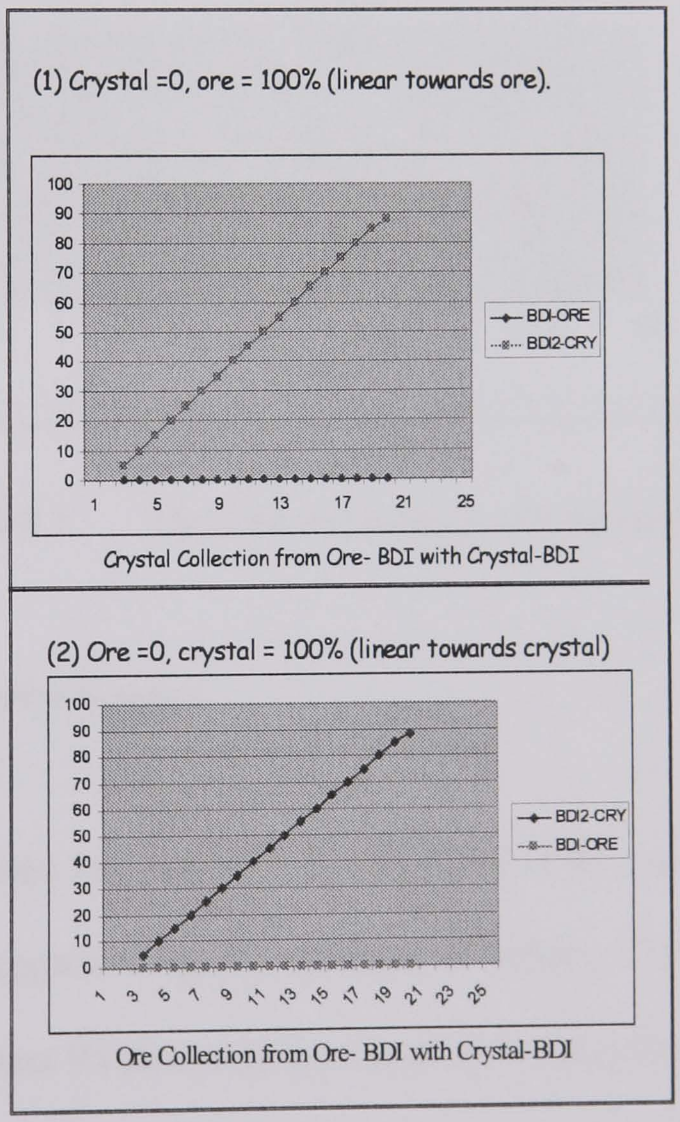
Graph 8.1 Performance of SMCA lower level to higher level agents

The results of this experiment (Graph 8.1) shows that BDI model agent maintains a higher level of life expectancy than other simple agents. Reflexive agents are collected

16% of ore, Reactive agents are collected 26% of ore, simple-learning agents collected 57% of ore and BDI agents are manage to collect 80 % ore BDI agents maintains a higher level of 72.5%, life expectancy than reflexive 26%, reactive 36.5% and learning agents 41% .

8.3 BDI Models to test purposeful actions.

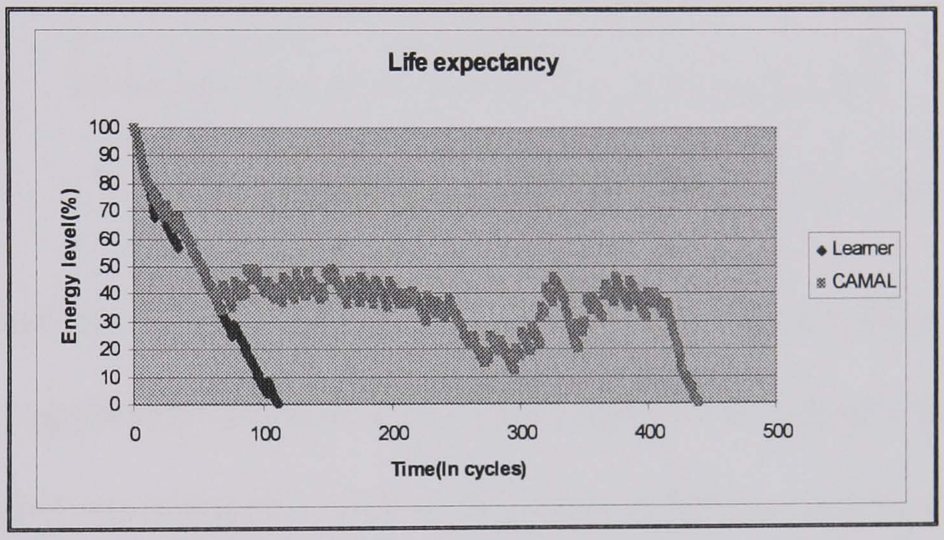
The results of this experiment graphs given below shows the different BDI models will work, based on rewards or resource collection. The graphs are completely linear, based on performance. For example:



Graph 8.2 Comparison of BDI models

8.4 Study Two (Experimentation on BDI models)

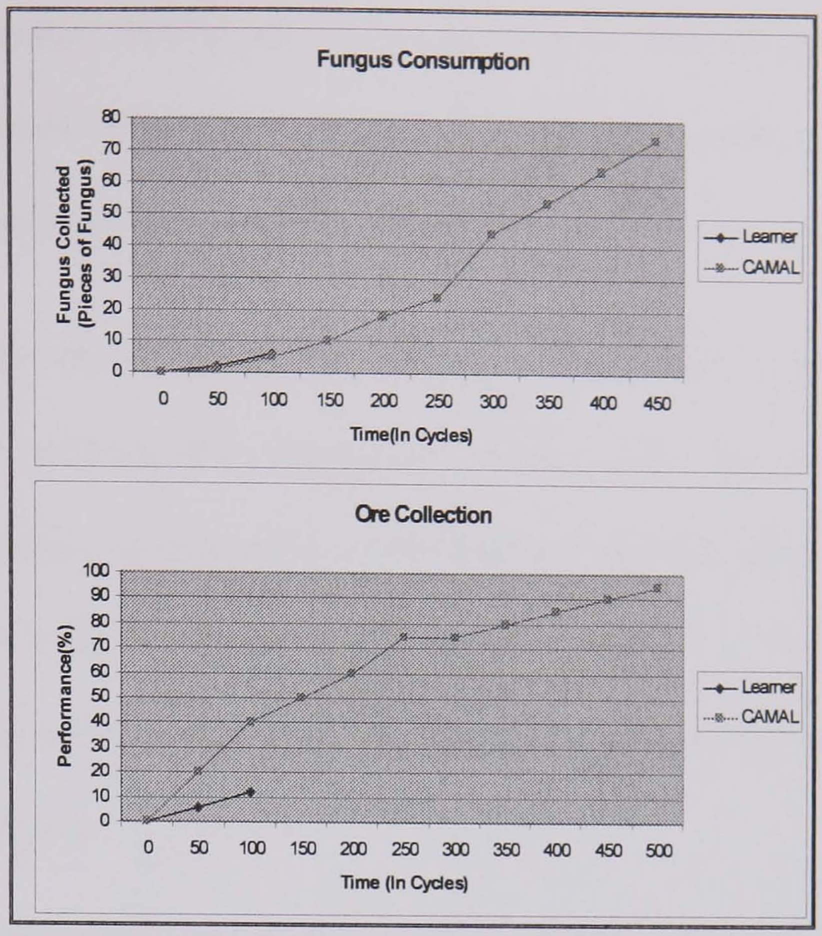
As shown in Graph 8.3, the BDI agent manages to live up to 438 life cycles. The BDI agent (Camal) shows a complete control mechanism in managing an energy level of 40 (assigned threshold or decision variable), and trying to manage the same line for the maximum time of its life cycle. The agents will exhibit optimal decision making capabilities near the decision boundary. The life expectancy of the two types of agents is shown below. The cognition (reflexive-learner) agent manages to live up to 110 life cycles in a fungus world environment.



Graph 8.3 The Life Expectancy of Cognition versus BDI Agents

8.4.1 BDI and Reflexive-learner

The resource (ore, golden ore and crystal) collection of the simple cognition and BDI agents is as follows: cognition agents managed to collect 12 pieces of ore, and BDI agents managed to collect 95 pieces of ore. Graph 8.4 illustrates agent decision making capability at the threshold value. If an agent acquires more than the threshold or predicted energy level, then agent tries to collect ore. If the agent has a lack of energy, then it collects fungus, from their hunger condition.



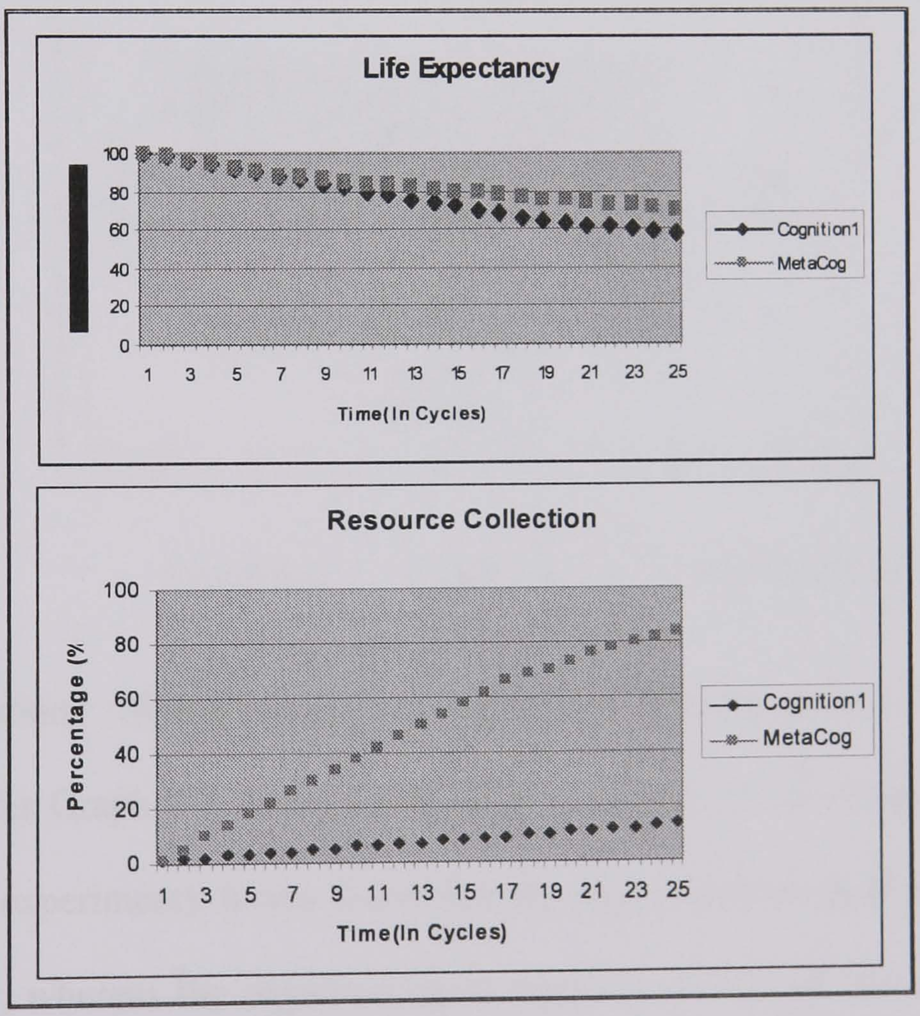
Graph 8.4 Fungus and Ore Collection

Graph 8.4 shows the fungus consumption rate of cognition and BDI agents in their lifetimes. The cognition(reflexive-learner) agent managed to collect 6 pieces of fungus and the BDI agent are managed to collect 74 pieces of fungus. As Graph 8.4 illustrates, in the initial stages, the (reflexive-learner) cognition agent was found to collect more fungus than the BDI agent. The BDI agent was not concerned about fungus in this stage. Agents in the initial stage born energy with medium metabolism. The BDI agent collects the medicine to decrease metabolism. Agents, once they achieved low metabolism by collecting required medicine, then it does not concerned about medicine.

8.5 Cognition, BDI and Metacontrol v/s Metacognition Agents

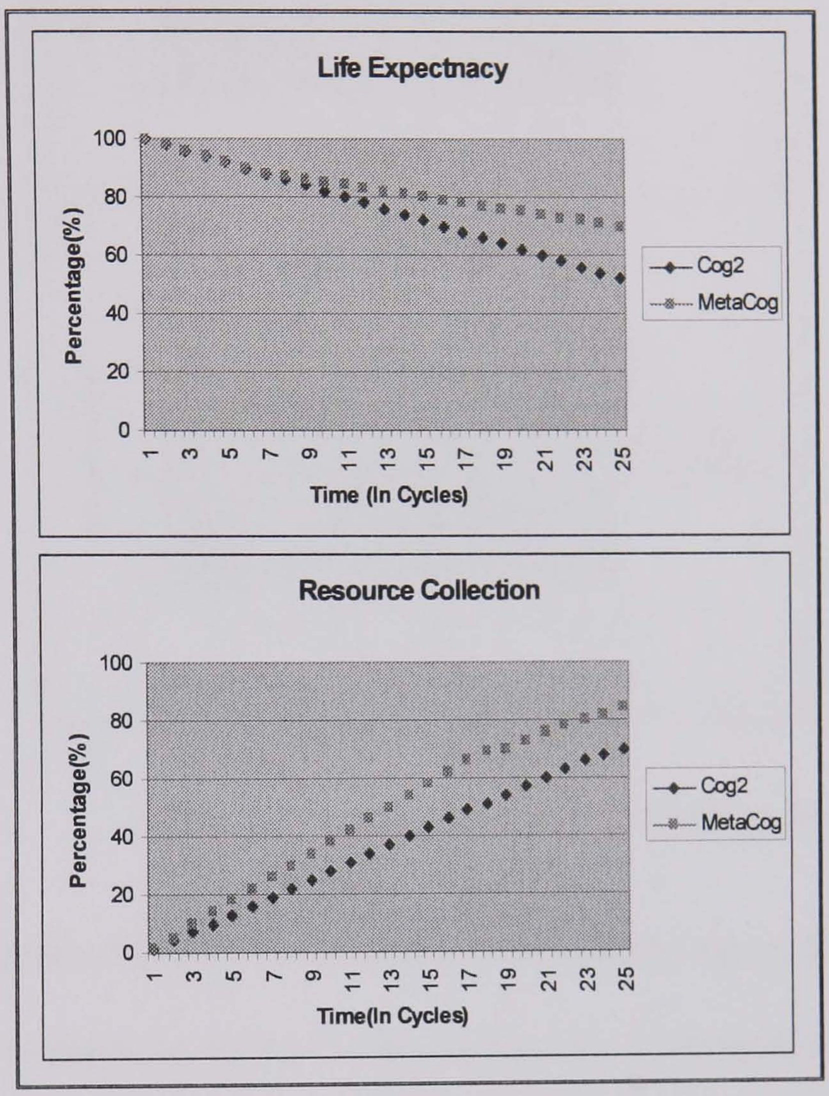
The experiments are conducted for various types of agents based on differentiating the cognitive model qualities including physiological and goal-oriented behaviour. The conceptual level associated with decision making with its cost function and utility behaviour, performance at the microeconomic level. This section presented some experiments in between cognition: (1) reflexive, (2) reactive-ore, (3) reactive-crystal, and (4) reactive-unit; and metacognition agents.

Initially, cognition1 and metacognition agents had the same percent of (100%) life expectancy. After running the experiment (Graph 8.5), the metacognition agent maintained 70% energy, as compared to 58% by the cognition agent. The metacognition agent was able to collect 84% of resources, as compared to 14% collected by the reflexive agent. The metacognition agents thus collect a higher percentage of resources than the cognition agent.



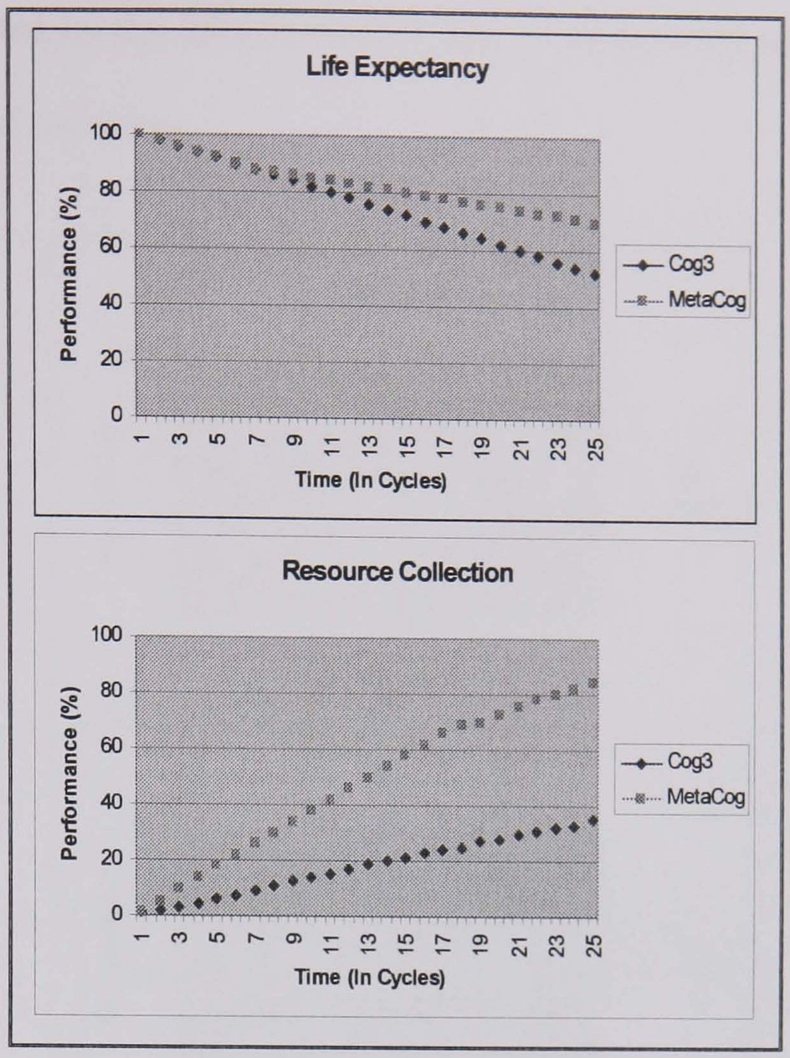
Graph 8.5 Cognition1 v/s Metacognition Agent.

Initially cognition 2 (reactive-Ore) and BDI agents had 100% life expectancy. After running the agent for maximum cycles, the following results were seen(refer Graph 8.6). The metacognition agent maintained 70 % energy, as compared to 52% for the cognition2 agent. The metacognition agent was able to collect 82% of resources as compared to 64% by the reactive (cognition type) agent. The metacognition agent thus collected a higher percentage of resource than the cognition agent.



Graph 8.6 Cognition 2 v/s Metacognition Agent

Initially Cognition3 (Reactive-Crystal) and metacognition agents had 100% life expectancy (refer Graph 8.7). After simulating the agents for 25 cycles (the maximum defined in this experiment), it was found that the metacognition agent maintained 70% life expectancy, whereas the cognition agent maintained 52% of life expectancy. The metacognition agent was able to collect 82% of the crystal as compare to 38% collected by the reactive crystal agent. The metacognition agent thus collected a higher percent of crystal (resource) than the cognition3 agent.

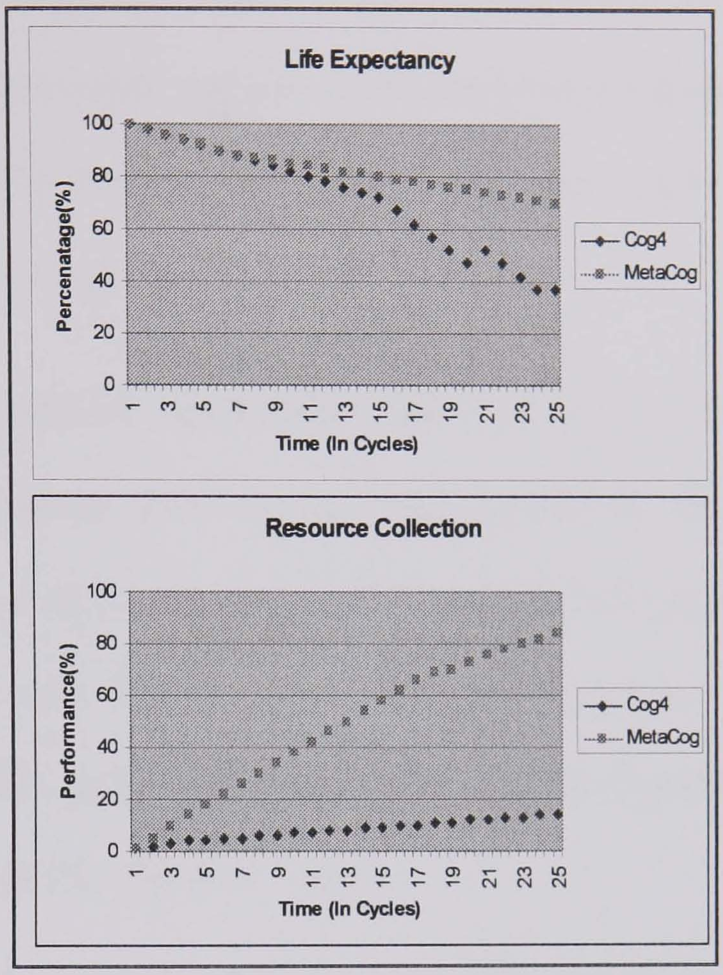


Graph 8.7 Cognition 3 v/s Metacognition agent

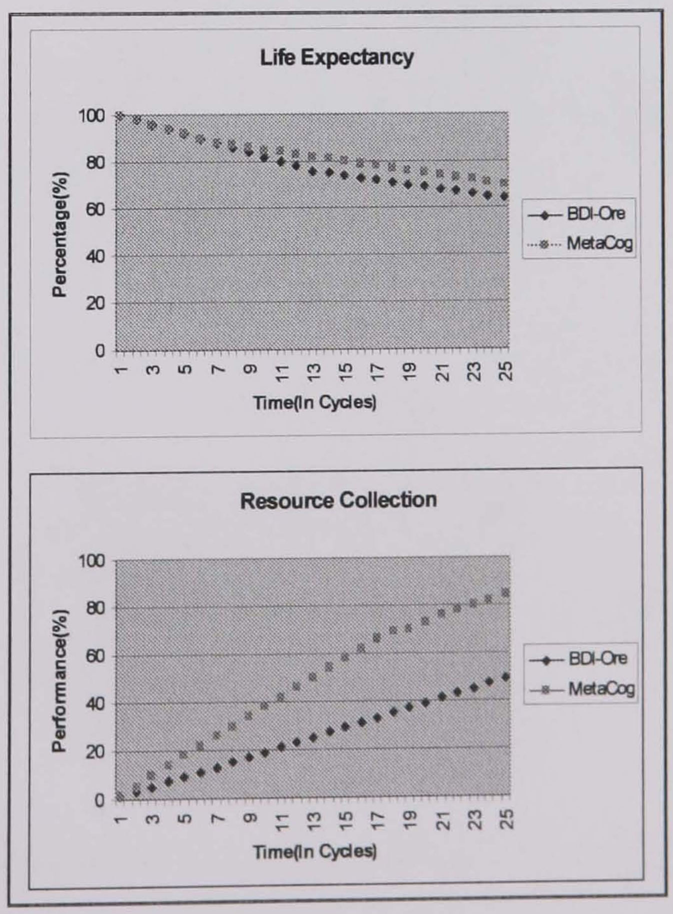
Initially, both cognition 4 (Reactive-unit) and metacognition agents possessed 100% life expectancy. After simulating agents for the maximum 25 life cycles, the metacognition agent maintained 70 % of life expectancy, compared to the cognition agent's 37% of life expectancy (Graph 8.8). The metacognition agent was able to collect 82% of the crystal, compared to 16% of the crystal collected by the reactive crystal agent. The metacognition agent thus collected a higher percentage of crystal (resource) then the cognition 3 agents.

Initially, the cognitive modell (or BDI-ore) and metacognition agents had the same percentage (100%) of life expectancy (Graph 8.9). After the experiment, the metacognition agent maintained 70% of energy, compared to 64% for the BDI-ore agent. The metacognition agent was able to collect 82% of resources, compared to 50%

of resources collected by the BDI-ore agent. The metacognition agent thus collected a higher percentage of resources.



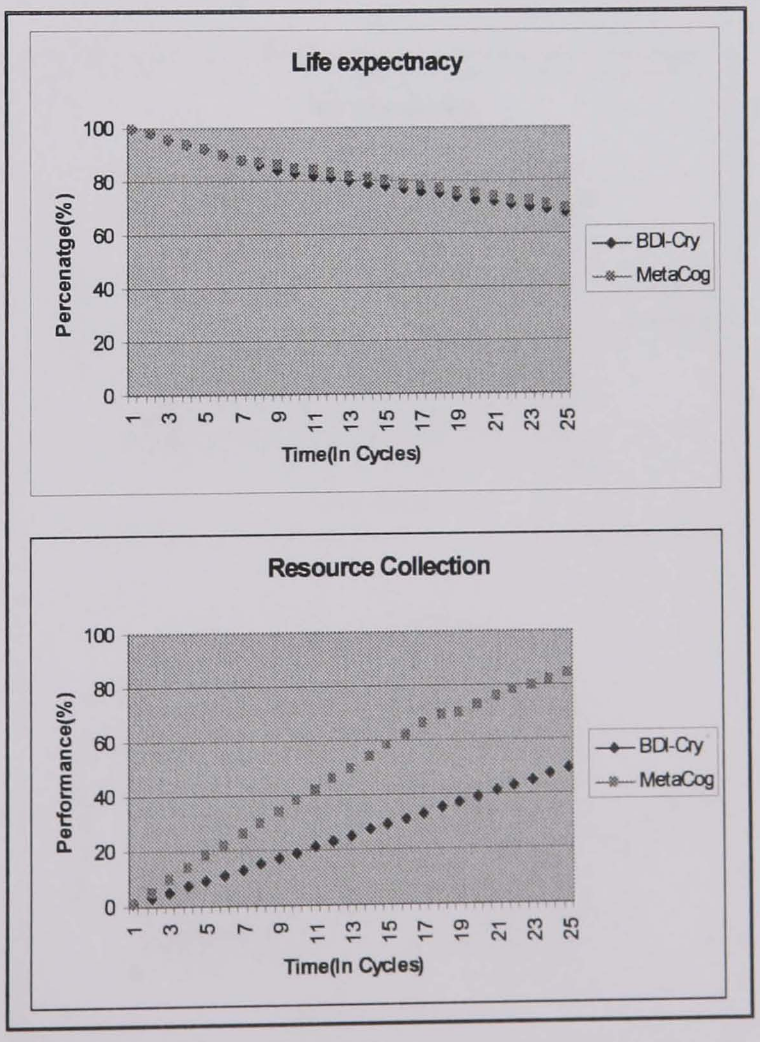
Graph 8.8 Cognition 4 v/s Metacognition Agent



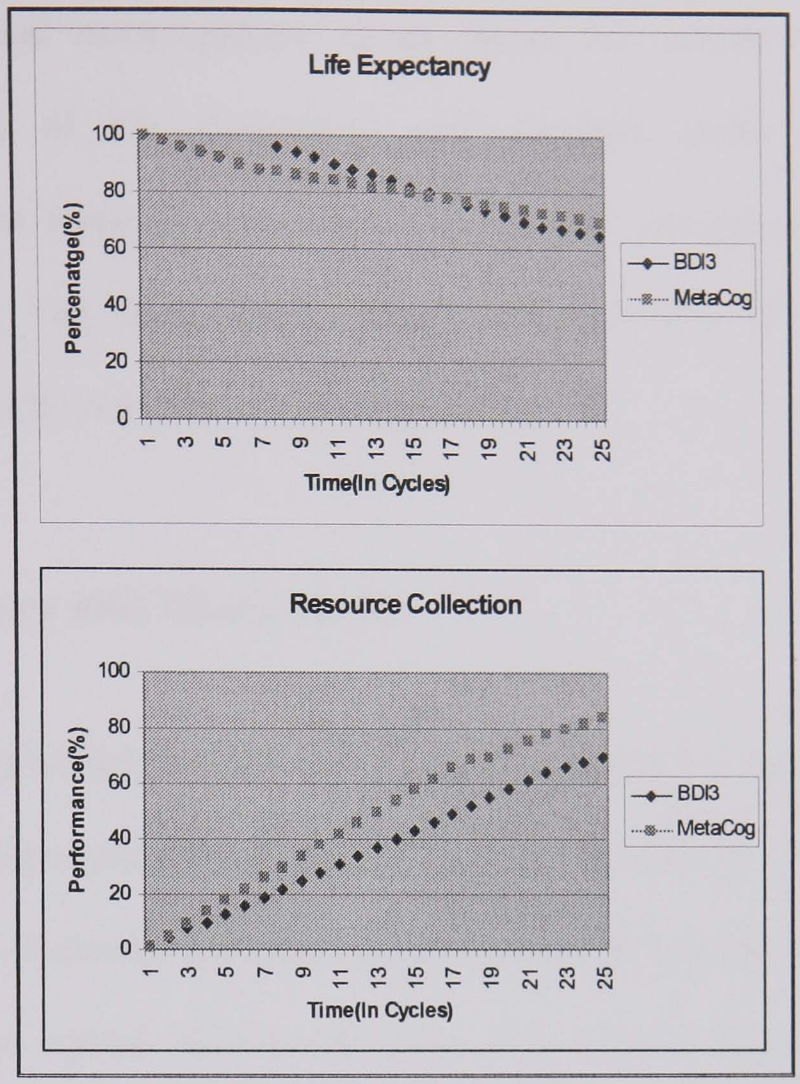
Graph 8.9 Cognitive Model 1 v/s Metacognition Agent

Initially, the cognitive model 2 (or BDI-crystal) and metacognition agents had the same percentage (100%) of life expectancy (Graph 8.9). After the experiment, the metacognition agent maintained 70% of energy, compared to 68% for the BDI-crystal agent. The metacognition agent was able to collect 84% of resources, compared to 49% of resources collected by the BDI-crystal agent. The metacognition agent thus collected a higher percentage of resources.

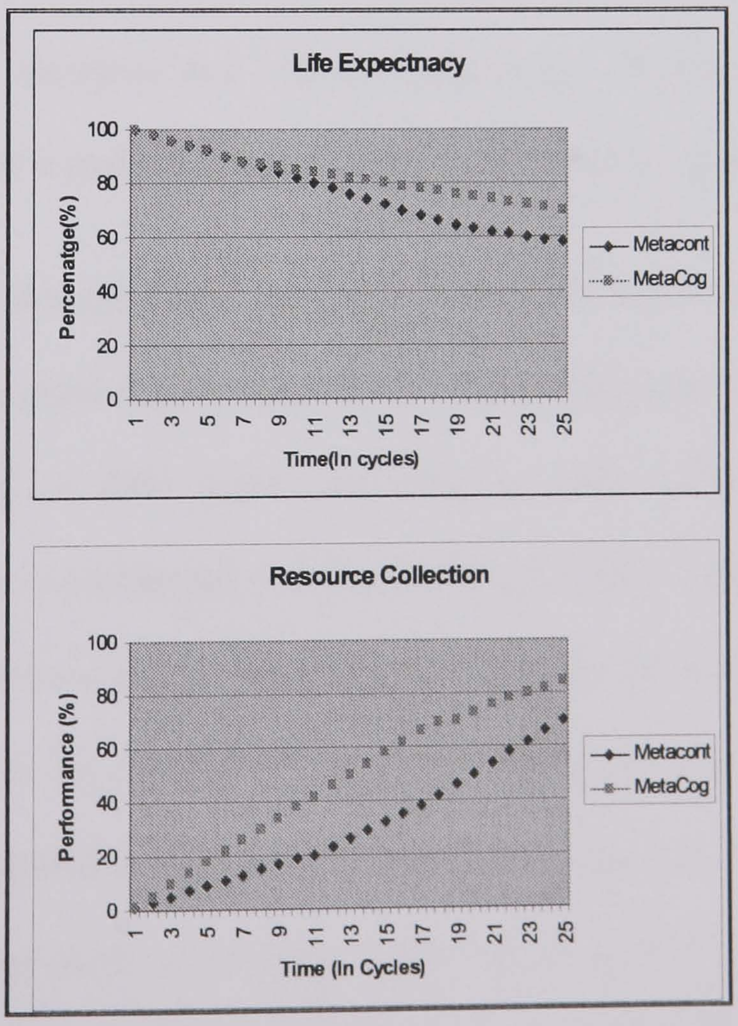
The BDI3 and metacognition agents began the experiment with the same percentage (100%) of life expectancy. After running the experiment, the metacognition agent maintained 70% of energy, compared to 65% for the BDI3 agent (Graph 8.11). The metacognition agent was able to collect 84% of resources, vs. 70% of resources collected by the BDI3 agent. The metacognition agent therefore collected a higher percentage of resources than the BDI 3 agent.



Graph 8.10 Cognitive Model 2 v/s Metacognition Agent.



Graph 8.11 Cognitive Model 3 v/s Metacognition Agent.



Graph 8.12 Metacontrol v/s Metacognition Agent

The metacontrol and metacognition agents began the experiment with the same percentage (100%) of life expectancy, and resources (refer Graph 8.12). The metacognition agent manages an energy level of 72%, compared to 58% from the metacontrol agent. The metacognition agent collected 82% of resources, and the metacontrol agent collected 68% of resources.

8.6 Summary and Discussions

The life expectancy and performance were the metrics used for assessing the efficiency of the agent. Life expectancy was defined as the survival of agents in a testbed for fixed energy or nutrients. Resource collection was defined as the number of resources, such as ore; golden ore and crystal, collected in given a time cycle. The results of all fifteen agents in SMCA (Society of Mind approach to Cognitive Architecture), which have complex behaviours, is a society of mind built by a society of agents, which demonstrates simple, moderate and complex behaviours. Agents demonstrate skills or capabilities like decision making, classification, intentions and commonsense activities.

The BDI models are designed to demonstrate how the metacontrol and metacognition, mechanisms can be applied within the different models (thinking of energy, thinking of metabolism, thinking of their goals, according to their self-reflection or internal conditions). From the experimental sections 8.2 and 8.3, BDI models or simple minds, the results show how two minds together are better than the either one alone. In this experiment, mind can be viewed as involving vastly different types of cognitive processes, such as predicting, repairing, reviewing, comparing, and generalizing, thus simplifying other many ways of thinking. Experiment 2 gives a more in-depth analysis comparing two types of agents. The first experiment was conducted for 100 life cycles.

The second experiment was conducted for 500 life cycles, to find out the in-depth potential of the agents through their lifespan. Cognition agents lived up to 110th of their life cycle. Cognition agents collected 12 pieces of ore and 6 pieces of fungus in their lifetime. BDI agent life expectancy is 438 life cycles, and the managed to collect 74 pieces of ore and 95 pieces of fungus in their life cycle. For the BDI agent, fungus consumption is considerably less, unless it was found to have less energy storage. As in Graph 8.4 depicts, in between the 250 and 300th life cycle, the BDI agent's fungus consumption rate is found be very high. In this stage, the BDI agent is in the hunger condition, and needs more fungus. Hence it switches towards the collection of fungus. This results proves that BDI agents can reason about their change of aims (deliberations), watch their status (self regulation or self control), and achieve their goals. BDI agents manifest decision making and intelligent behaviours (refer section 8.3). BDI agents have a complete control mechanism for managing food and metabolism. These agents' exhibit decision making capabilities near decision variable boundary. BDI agents engaged in activities to utilize their pattern of behaviour with respect to the use of energy and time. The level of decision making when they are hungry (less than the decision making energy level) switch into the fungus consumption and if they normal, switch towards goal-oriented (i.e. collection of ore), demonstrates physiological and goal-oriented behaviour. BDI agent manages the affect mechanisms, such as energy level, based on a given threshold or predicted energy level to manage the decision boundary.

Based on the results obtained in these experiments (refer section 8.4) metacognition agents consistently performs better against the cognition and BDI agents in all experiments. Metacognition agents are more efficient to manage their energy level as well as collecting more resources. Where as the Cognition agents are unable to control their energy levels (internal conditions), and some times comes down to zero energy level and dies before completing their maximum cycles.

Metacognition agents managed to collect more number of pieces of Ore and Crystals (resources) collected with the maximum cycles defined (i.e.25 cycles), as compare to cognition agents. It concludes that, metacognition agents, has the complex and intelligence (optimal) behaviours to constitute a "Society of Mind" for sensing in the environment. The metacognition agent shows complete control mechanism in managing a food and metabolism (Affect), try to balance motivations. The agents are exhibiting optimal decision making capabilities near a decision variable boundary. The metacognitive activity reduces to turning the individual BDI models (BDI-Ore, BDI-Crystal, BDI-Ore and Crystal) to the ON and OFF state, based on norms and affect. At any given point in the cycle, some agents in the Society of Mind are active, while others are static. This combined status of mind constitutes Minsk's view of "partial state of the mind". The society of agents demonstrates the "Society of Mind" with their different Belief-Desire-Intention models. Agents demonstrates "Society of Mind" concept in terms of the arrangement of activities within their planning, reasoning, decision making, self reflection, problem solving and learning capabilities, from different combination of agents. Metacognition agent collects more resource and manages the higher life expectancy than any other agents. This result proved a concept of metacognition is a powerful catalyst for control and self-reflection. Metacognition concept used on BDI models improved the performance. Finally, different combination of agents in SMCA demonstrated task effectiveness, goal achievement, and the ability to perform well in novel situations.

Chapter 9 Conclusions

The work presented in this thesis provides details of the theory, design, implementation and testing of a Society of Mind Cognitive Architecture (SMCA), running in a simulation environment. This Chapter presents solutions or progress and remarks towards questions raised in the Introduction Chapter. This Chapter also details limitations of the SMCA and gives some directions for future research.

9.1 Research questions and solutions

This research project addressed issues associated with the development of a SMCA (Society of Mind approach to a Cognitive Architecture) based on an extended CAMAL cognitive architecture with extra processing layers using a distributed model of mind, and in doing so impacts on the different questions. The questions posed in Chapter1 are answered in the chapters of this thesis. These questions are re-visited to provide solutions or at least some steps or progress with discussions towards answers.

What are the principles used for designing a SMCA?

The SMCA is designed based on the principles of animal cognition. The behaviour of an animal has consequences which depend on situation, energy use and other physiological commodities such as water, weather etc. The important consequence of animal behaviour is energy expenditure. According to Thorndike (1911), the behaviour of animals is predictable and follows the uniformity of nature. He says that "any mind will produce the same effect, when it is in the same situation." Similarly, an animal produces the same response, and if the same response is produced on two occasions, then the animal behaviour for that response must changes. The law of instinct or original behaviour is that an animal in any situation, apart from learning, responds by its inherited nature(McFarland, 1993; Berger's, 1998).

Animal behaviour is not simply a matter of cognition; rather it is product of the behavioural capacity and the environmental circumstances (McFarland, 1993; Bosser, 1993).Charles Darwin in his book Descent of Man (1871), argued that animals possess some power of reasoning. This research is concerned with the principles whereby agents as like animals competent for its resources, and so demonstrates intelligent behaviour in developed testbed of SMCA. Chapter 2 and Chapter 8 answer this question. The principles of artificial minds are given below (McFarland, 1993; Berger's, 1998).

Decision Variables

A decision-making of a person, animal or robot can be described as an activity whereby decision variables are compared to decision boundaries. From the economic point of view, the decision-making unit is the cost or performance. Decision-making with respect to use of energy (food) and benefit (goal) based on given decision variables or decision boundaries. Cognitive modelling design and implementation based on the analogies between animal, person and products. Metacognition, metacontrol and BDI agents in SMCA testbed manages the affect mechanism, such as energy level, based on a given threshold or predicted energy level to manage the decision boundary. Chapter 2 and Chapter 8 answer this question.

Cost and Utility Function

The decision making level in animals is defined in terms of cost functions and utility behaviours - the microeconomic level. Cost functions and utility behaviour in animals operate in such a way that a utility (for example, energy) is maximized or minimized (McFarland, 1993). Metacognition, metacontrol and BDI agents in SMCA testbed manages to maximized the energy level by eating fungus (food).

Learning in Animals

Learning is a part of development. It is a result of adaptation to accidental or uncertain circumstance. When the animal learns environmental situations, it undergoes permanent change. We expect that learning should, in general, bring beneficial results. Animal learning is similar to reinforcement learning in machine learning or robotics (McFarland, 1993; Nason and Laird, 2004).

The learned agents in SMCA, changes the goals and deliberative steps according to given rules. This may change to move towards fungus, ore, crystal and medicine at basic reflexive and reactive levels. Learning also applied on higher level layers in SMCA. The learning mechanism can follow according to rules framed in the deliberative, metacontrol and metacognition levels. The metacontrol mechanisms can be viewed in terms of which the agents use existing controllers, learn behaviours (i.e. existing Q(s, a) values) or learn new behaviours by training the agents. Q-learning mechanisms within SMCA architecture has been discussed previously in Chapter2, Chapter6 and Chapter 8.

Optimal Behaviour in Artificial Minds

Animal behaviour is a trade off between the native courses of action, i.e. physiological, and goal oriented behaviour. Animals are engaged with activities to optimize its pattern of behaviour with respect to the use of energy and time. If the conditions are relevant to two or more activities simultaneously, it chooses the most optimal action among them in terms of its innate and learnt decision boundaries. The mechanisms of designing a machine are different from the animal's kingdom, but the principles remain the same (McFarland, 1993; Bosser, 1993) (refer Chapters 2). The BDI, metacontrol and metacognition agents in SMCA shows complete control mechanism in managing a food, metabolism (affect) and try to balance motivations. The agents are exhibiting optimal decision making capabilities near a decision variable boundary.

What are the metrics for measuring a performance of agents?

Agent behaviours can be analysed using many different metrics. The major metrics are metabolic activity, competition and social interaction with respect to environment and microeconomics. Application of economics on the artificial life to watch adaptive behaviours. This follows the microeconomic regularities such as cost and utility.

Life expectancy and resource collection are two metrics used in the SMCA testbed experimental results. The life expectancy is defined as the survival of agents in a testbed for fixed energy or nutrients. Resource collection is defined as the number of resources such as ore; golden ore and crystal are collected in given a time cycle (refer Chapter2 and Chapter8).

What is the difference between reflexive, reactive, deliberative, learning, metacontrol and metacognition level processes in a cognitive architecture?

The Society of Mind Cognitive Architecture (SMCA), developed in this thesis, extends the CAMAL architecture with extra processing layers as a distributed model of mind, using the "Society of Mind" concept. Minsky (1985) state that, intelligence is a combination of more simple things, and every Mind is a "Society of Mind". Hence Mind consists of a great diversity of mechanisms. The term agent refers to the basic element or simplest individual which constitutes an active element in a "Society of Mind". The SMCA architecture has designed and implemented for six layers: the reflexive, reactive, deliberative (BDI), learning (Q-learner), metacontrol and metacognition processing levels. This leads to the development of many different types of simple agents, with different behaviours. These agents are distributed across the different layers of the architecture. To cover all the processing and functioning associated with the adopted model of mind, requires the development of many different agent behaviours. Presently, SMCA comprises of six reflexive behaviours, eight reactive behaviours, fifteen deliberative behaviours, nineteen perceptual behaviours, fifteen learning behaviours, fifteen metacontrol behaviours and seventy seven metacognitive behaviours. Experiments conducted for different types of agents based on differentiating the agent performance. Metacognition and its relation to metacontrol and learning are discussed in the previous Chapters 4, 6, 7 and 8.

What is Metacogntion? What is the difference between cognition and metacognition? Is deliberation necessary for metacognition?

Cognition is defined as a mental process or activity that involves the acquisition, storage, retrieval, and use of knowledge. The mental processes include perception, memory, imagery, language, problem solving, reasoning, and decision making (Zalta, 2005; Wilson & Kiel, 1999).

Metacognition is often simply defined as "thinking about thinking" (Wilson & Keil, 1999). Metacognition is any knowledge or cognitive process that refers to monitoring and controlling any aspect of cognition. Adkins (2004) defines "metacognition is thinking about knowing, learning about thinking, control of learning, knowing about knowing and thinking about thinking". Minsky (1985) defines "we cannot think about thinking, without thinking about thinking about something". The metacognitive act can be referred to as metacontrol. Metacognition can be viewed in two ways: (1) monitoring a group of agents in an intelligent or cognitive or robotic architecture (i.e. self reflection) and; (2) making changes by adapting effective strategies in the group of agents.

The metacognition concept provides a powerful tool towards developing efficient and quality computational models. This research investigates the concept of metacognition as a powerful catalyst for control, unify and self-reflection. Metacognition is used on BDI models with respect to planning, reasoning, decision making, self reflection, problem solving, learning and the general process of cognition to improve performance. The reactive class of agent, in turn, provides a computational platform for the deliberative agents. The design of deliberation mechanisms for the fungus testbed includes five different types of BDI agents. The BDI determines which of the reactive or reflexive control mechanisms are active according to the goals to satisfy. These goals are either task related or agent-internal resource related, and determine the number of different types of reflexive and reactive agent required for this specific testbed. Chapter 4, Chapter 6 and Chapter 8 explains clear ideas about implementation and results part of the cognition and metacognition.

What are the different parts of metacognition? How can these parts be designed, programmed and verified using a simulated environment?

Chapter 4 and Chapter 6 had given a clear roadmap for researches, to develop a metacognition concept in computational models. Metacognition is used on BDI models with respect to planning, reasoning, decision making, self reflection, problem solving, learning and the general process of cognition to improve performance.

Metacognition concept in a SMCA model is based on the differentiation between metacognitive strategies and metacomponents or metacognitive aids. Metacognitive strategies denote activities such as metacomphrension (remedial action) and metamangement (self management) and schema training (meaning full learning over cognitive structures). Metacomponents are aids for the representation of thoughts. To develop an efficient, intelligent and optimal agent through the use of metacognition requires the design of a multiple layered control model which includes simple to complex levels of agent action and behaviour. This SMCA model has designed and implemented for six layers which includes reflexive, reactive, deliberative (BDI), learning (Q-learner), metacontrol and metacognition layers.

What are the metacomonents? Explain how these metacomponents can be used in society of agents?

Metacognitive aids or metacomponents are used for the representation of thoughts. Metacomponents can be represented with the help of some aids such as: (1) using an abstraction, metasyntactic variable (matching variables) or metacomponents and; (2) goal setting variables for increasing the performance. Metacomponents affects on the agent behaviour from a sense of what is important instead of what to do. Metacognition agents will follow well aligned norms, perceptual range, metarules, and learning and affect values. A well driven agent will maximize its performance as a consequence of learning to maximize its own reward. These executive processes involve planning, evaluating and monitoring the problem solving activities (Zalta, 2005, Adkins, 2004). The term "norm" is an interdisciplinary term, and can be used to refer to a standard principle or a model used for a right action. The executive processes that controls the other cognitive components are responsible for "figuring out how to do a particular task or set of tasks, and then making sure that the task or set of tasks are done correctly". Norms in society of minds can be guided, controlled and regulates the proper and acceptable behaviours.

Metacomponents in metacognition layer reason about the action, reflect upon that reasoning, and assess cognitive activity with respect to meatacontrol task. Given a loaded norm for resources (such as ore, golden-ore and crystal), energy decision boundaries, affect values for medicine, fungus, ore, crystal or ore and crystal. This can be used to decide upon the desire (i.e. when and what resource to collect). Chapters 4 and chapter 6 answer this question.

What are BDI models? How can BDI models plan in different circumstances?

BDI (Belief- Desire-Intention) model has different group of coordinated capabilities to meet a particular intention. Deliberative agents compromise the third layer of the Society of Mind Cognitive Architecture. The design of deliberation mechanisms for the fungus world testbed includes five different types of BDI agents. The BDI determines which of the reactive or reflexive control mechanisms are active according to the goals architecture attempts to satisfy. These goals are either task related or agent-internal resource related, and determine the number of different types of reflexive and reactive agents required for the specific testbed.

BDI models follow the reactive mechanisms. The metacognition agent shows complete control mechanism in managing a food and metabolism. The agents exhibit decision making capabilities near a decision variable boundary. The energy spent (maximum cycles follows low metabolism) in each move of BDI types exhibits minimal or minimized (due to maintenance of low metabolism), and utility is also maximized. BDI agents engaged with activities optimize its pattern of behavior with respect to energy and time. There are different BDI models, for different purposes are used in this experiment. They are BDI-ore, BDI-crystal, BDI-ore-and-crystal, BDI model that plans ahead and reflexive- BDI model. Chapters 6, 7 and 8 discusses about the theoretical issues, design, implementation and results part of the different BDI models.

What is Society of Mind?

Minsky views intelligence as not just a simple recipe or as an algorithm for thinking, but a combined social activity of more specialized cognitive processes. According to Minsky, every mind is a "Society of Mind." The mind consists of a great diversity of mechanisms. Minsky proposes that the mind is made up of simple and smaller entities called micro-agents. Minsky argues that each agent is like a simple piece of code, and can do simple work. The agents can be connected within a larger system called a society of agents. Each individual agent, having a different background, plays a different role in society. The society of mind results from combining more specialized cognitive processes.

In the "Society of Mind", the active agents are in the "on" state, and non-active agents are in the "off" state. The combined activity of active agents constitutes a "total state" of mind, and the subset of the activities represents a "partial state of mind." Minsky claims that K-lines are the selecting most common agents in the Society of Mind theory. The actors (agents) demonstrate the "Society of Mind" concept in terms of the arrangement of activities within a SMCA for their planning, reasoning, decision making, self reflection, problem solving and learning capabilities. SMCA can be viewed as containing vastly different types of cognitive processes, such as predicting, repairing, reviewing, comparing, generalizing, and simplifying and many other ways of thinking. Chapter 5, Chapter 6 and chapter 8 explains how to build an artificial model that combines reflexive, reactive, deliberative, learning, metacontrol and metacognition processes across the "Society of Mind" architecture to demonstrate how intelligent agent can be viewed as a large collection of agents or single agent as collective behaviours (Metacognition agent) as a "Society of Mind. The development of SMCA (Society of Mind approach to a Cognitive Architecture) using extended CAMAL cognitive architecture with extra processing layers demonstrates "Society of Mind".

9.2 Summary of Main Contributions

This thesis has made several contributions to the field of artificial intelligence and cognitive sciences.

This thesis addressed two broader aims and research questions related to development of Society of Mind Cognitive Architecture (SMCA).

This thesis illustrated Society of Mind approach to a Cognitive Architecture. The mind is a control system, and uses the "Society of Agents" metaphor. Simulation demonstrates Minsky's approach of "Society of Mind."

The thesis, Society of Mind Cognitive Architecture is the first architecture, completely viewing cognitive architecture in the perspective of "Society of Mind" and society of agents.

This research has thrown some light on the operation of animal minds and provided rich evidence for McFarland's (1993) theoretical issues of microeconomics and animal cognition. The main principles are: (a) cost and utility function, (b) physiological and goal oriented behaviour and (c) decision-making based on decision boundary or decision variable.

The implementation of cognition and metacognition techniques on a society of agents to demonstrate "Society of Mind" architecture gives rich evidence to Minsky's theoretical issues of "Society of Mind," Minsky's A, B and C-Brain, a Sloman's (2001, 2002) metamangement and Kennedy's (2003) self- reflection concepts.

This thesis, Society of Mind Cognitive Architecture gives the clear difference between cognition and metacognition processes.

This thesis explains how to build an artificial model that combines reflexive, reactive, deliberative, learning, metacontrol and metacognition processes across the SMCA architecture, to demonstrate how an intelligent or optimal agent can be viewed as a large collection of agents or single agent's collective behaviours (metacognition agent) as a "Society of Mind".

The developed SMCA is an example for complex AI systems that have metacognition techniques inbuilt in their architecture.

Finally, this thesis gives a clear road map for researchers, to develop a metacognition concept on AI applications based on metacognitive strategies, such as: (a) metacomprenhsion (remedial action), (b) self-regulation (metamanagement) and (c) schema training (meaningful learning), along with metacognitive aids or metacomponents such as perceptual range, affect, norms and higher level meta-rules.

9.3 Further Research Directions and Limitations

Further research onto develop a Metacognition concept on robots. This can be tested on 'real world' rather than simulations in testbed scenario. There are large extensions and directions can be made to SMCA (Society of Mind approach to a Cognitive Architecture). Presently have around fifteen agents, and one seventy seven behaviours. This can be extended with some more complex skills. For example to develop a human agent (biological agent) as suggested by Sloman in H-CogAff as a part of SMCA.

Another area for future research is the inclusion of perceptual behaviours within SMCA. Agents require communicating each other to compute a particular task. This approach can be fits with Barasolu's perceptual symbol systems. Connectionist model approach can recognize the symbols systems for interaction. For example, common sense computing architecture (refer Singh's EM-One in Chapter 3). SMCA can be extended by adding perceptual behaviours. SMCA architecture needs a simulator that produces limitless simulations to demonstrate perceptual behaviours. Perceptual behaviours are very well explained from Barasalou's theory of a perceptual symbol systems (Barasalou, 1999). Barasolu's perceptual symbol systems explain how the brain can capture images, represent and store. Barasolu's concepts are causal relations between the world and environment (Barasalou, Simmons, Barbey & Wilson, 2003).Goldstone (1994) supports Barasalou's statements and explains how the objects can be simplified and filled, from perception.

SMCA model can be extended further into different application areas such as education, business, entertainment, etc. Society of mind cognitive architecture can be extended by adding some more behaviour for each of the layer. For example, adding SARSA and conceptual learning as part of SMCA. So that metacontroller can get multiple options to choose learning mechanisms. SARSA (State-Action-Reward-State-Action) is a learning algorithm. This works based on the markov decision process policy. As similar to q-learning mechanism SARSA works in terms of agent based simulations. According to Takadama (2004), SARSA agents are better than q-learning mechanism in terms of negotiation.

Conceptual learning is a learning mechanism. This algorithm works based on the understandings of people learning methods. This mechanism works based on contrast to factual knowledge. This can be implemented as part of the SMCA model.

Another Future work is to giving database support for SMCA cognitive model. For example Cyc model (Lenat, 1995) contains more than two millions of facts and rules about the every day world. This ontology level can be added as part of the SMCA. This database support contains the wide area of representations. For example, space knowledge, beliefs, time, social relationships, physical objects, and other domain areas.

There are several limitations in SMCA.The developed SMCA model is much limited to test certain behaviours, and not up to the demonstration of complex behaviours of human. Developed SMCA does not much supports for communication in between the society of agents. SMCA cognitive model testbed presently works on simulation scenario. These simulations are very difficult to test on real world robots. SMCA model is very much limited in representation. SMCA only uses BDI (belief-desire-intention) models for knowledge representation.

9.4 Summary

This Chapter summarized solutions or progress and remarks towards questions raised in this thesis. This Chapter also gives limitations of the SMCA and some directions for future research. The developed Society of Mind Approach to cognition and metacognition in a cognitive Architecture is an example for complex AI systems that have metacognition techniques inbuilt in their architecture. There are large extensions and directions can be made to SMCA (Society of Mind approach to a Cognitive Architecture). Developed SMCA does not much supports for communication in between the society of agents.

References

Adkins J, 2004, *Metacognition*, university of Saskatchewan, Canada.

Anderson J R & Bower G H, 1973, Human associative memory, Washington, DC: Winston & sons.

Anderson J R, 1993, *Problem solving and learning*, American Psychologist, 48, 35-44.

Anderson J R, 1993, *Rules of the Mind*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Anderson J R, John B E, Just M A, Carpenter P A, Kieras D E and Meyer D E, 1995, *Production System Models of Complex Cognition,* In Proceedings of the Seventeenth Annual Conference of The Cognitive Science Society, Hillsdale NJ, Lawrence Erlbaum Associates.

Anderson J R, 1996, ACT: *A Simple Theory of Complex Cognition*, American Psychologist, 51, 355-365.Available:http://actr.psy.cmu.edu/publications[Accessed 5th November 2005].

Anderson J R, Lebiere C, 1998, *the Atomic Components of Thought*. Available: http://act-r.psy.cmu.edu/publications [Accessed 6th November 2005].

Baars B J, 1988, A Cognitive Theory of Consciousness. New York: Cambridge University Press.

Baars B J, 1998, In The Theatre of Consciousness: Global workspace theory, a rigorous scientific theory of consciousness, Journal of Consciousness Studies, 4, 292-309 Oxford: Oxford University Press..

Baars B J, and **Franklin S,** 2003. How conscious experience and working memory interact. Trends in Cognitive Science 7:166-172

Barsalou L W, 1999, *Perceptual Symbol Systems,* Behavioral and Brain Sciences, *22*, 577-609.

Barsalou L W, 1999, *Perceptions of Perceptual Symbols*, Author's response to Commentary on "Perceptual symbol systems", Behavioral and Brain Sciences, *22*. 633-660.

Bartsch K and **Wellman H,** 1989, *Young Children's Attribution of Action to Beliefs and Desires.* Child Development, 60.946-964.

Bechtel W and **Abrahamsen** A, 1991, Connectionism and the Mind: An Introduction to Parallel Processing in Networks, Blackwell.

Bedau M A, 2003, *Artificial life: Organization, Adaptation, and Complexity From the Bottom Up, Trends In Cognitive Science*, Volume 7, Issue 11, Pages 505-512, [Online] Available at www.sciencedirect.com [Accessed 22nd April 2007]

Bickhard M H and Terveen L, 1995, Foundational Issues in Artificial Intelligence and Cognitive Science, Elsevier.

Beaudoin L P, 1994, Goal processing in autonomous agents, PhD thesis. School of Computer Science, the University of Birmingham, England.

Benes V, 2004, *Metacognition in Intelligent Systems*, PhD study report, University of west Bohemia in Pilsen, Czech Republic.

Berger's J, 1980, *'why look at animals'* Available at http: www. Vivaria.net/experiments/why [Accessed at 15th, August, 2006].

Boden M A, 1990, (Editor), the Philosophy of Artificial Intelligence, Oxford University Press, 1990.

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

Bratman M E, 1987, Intentions, Plans, and Practical Reason Harvard University Press, Cambridge, MA.

Brown A L, 1987, Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In F.E.Weinert & R.H.Kluwe (Eds.), Metacognition, motivation, and understanding (pp.65-116).Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Brustoloni J C, 1991, *Autonomous Agents: Characterization and Requirements*. Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University.

Budiu R, 1998, About ACT-R, Available at ACT-R Home page Available: http://actr.psy.cmu.edu/about/. [Accessed 2nd May 2005].

Byrne D M, 2002, Cognitive architecture, Department of Psychology, Rice University, USA.

Byrne D M, 2005, Cognitive Architectures in HCI: Present Work and Future Directions, 11th International conference on Human computer Interaction, Las Vegas, NV.

Carr Kurtz B E Schneider W, **Turner L A**, and **Borkowski J G**, 1989.Strategy acquisition and transfer among German and American children: Environmental influences on metacognitive development. *Developmental Psychology, 25,* 765-771.

Clark A, 1939, *Metacognition Philosophy*, Cognitive science and Parallel Distributed Processing.

Cohen P R and Levesque H J, 1990, "Intention is choice with commitment" *Artificial Intelligence*.

Copeland J, 2000, what is Artificial Intelligence, Available at http://www.alanturing.net/turing_archive/pages[Accessed 7th November 2005].

Chantemargue M, Courant and Hirsbrunner B, 1999, Autonomous Agents and Collective Robotics "citeseer.ist.psu.edu/108919.html".

Choi D., Kaufman M., Langley P., Nejati N., and Shapiro D, 2004, An architecture for persistent reactive behavior.*Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems* (pp.988-995). New York: ACM Press. Available at http://cll.stanford.edu/research/ongoing/icarus/ [Accessed 18th August 2006].

Cox M T, 1994, Metacognition, problem solving and aging, Cognitive Science Tech. Rep. No. 15, Atlanta, Georgia Institute of Technology, College of Computing.

Cox M T, 1995, Representing mental events (or the lack thereof), in: M.T. Cox, M. Freed (Eds.), Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms, AAAI Press, Menlo Park, CA, 1995, pp. 22-30. Available as Technical Report SS-95-08.

Cox M T 2005, *Metacognition in Computation: A selected research review*, BBN Technologies, 10 Moulton St, Cambridge.

Dennett D C, 1978, Current Issues in the Philosophy of Mind." *American Philosophical Quarterly* (October 1978), 15(4):249-261.

Dennett D C, 2001, Are we explaining consciousness yet? *Cognition*, 79, 221-237, p. 221.

Darwin C, 1872, Expression of The Emotions In Man And Animals, 1872, Paul Ekman (ed), Oxford, 1998, ISBN 0195112717

Davis D N, 1996, Reactive and Motivational Agents (ECAI-96), Budapset, and 1996, In Agent Theory, Architecture and Language III, Springer-Verlag, 1997.

Davis D N, 1997, Reactive and motivational agents. In: Intelligent Agents III: Agent Theory, Language and Architectures, Springer-Verlag.

Davis D N, 2002, Computational Architectures for Intelligence and Motivation, *17th IEEE International Symposium on Intelligent Control*, Vancouver, Canada.

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

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

Davis D N, 2004, Ed. *Visions of Mind: Architectures for Cognition and Affect*. Idea Group, Hershey, PA, Information Science Publishing.

Davis D N, 2007, *Linking Perception and Action through Perception, Experimental & Theoretical Artificial Intelligence*, and JETAI is available online at: http://www.tandf.co.uk/journals/tf/0952813X.html.

Davies I, 2005, Perception and cognition, University of Surrey, Guildford Available http://www.surrey.ac.uk/Psychology/RGPerceptionCognition.htm [Accessed 2nd September 2006].

Davidson G, 1997, Chambers Pocket dictionary, Caledonian International Book manufacturing Ltd, Glasgow, Great Britain.

Dillmann R, 2004, Benchmarks For Robotics Research, European Robotics Network, and University of Karlsrube.

Ernst G & Newell A, 1969, GPS: A Case Study in Generality and Problem Solving. New York: Academic Press.

Feigenbaum E A & Simon H A, 1984, EPAM-like models of recognition and learning. Cognitive Science, 8, 305-336.

Flavell J H, 1979, Metacognition and cognitive monitoring: A new area of cognitive developmental inquiry. *American Psychologist, 34,* 906-911.

Flavell J H, 1987, Speculations about the nature and development of metacognition In Weinert F E and Kluwe R H (Eds.), *Metacognition, Motivation and Understanding* (pp.21-29).Hillside, New Jersey: Lawrence Erlbaum Associates.

Fodor J L, 1975 *The language of thought*, Thomas Y.Crowell Co. (Paperback, Harvard University Press).

Fodor J A, 1981, *Representations*, Cambridge, Mass: The MIT Press.

Fodor J A, 1982, "Cognitive Science and the Twin-Earth Problem," *Notre Dame Journal of Formal Logic* 23 (1982): 98-118.

Fodor J A, 1983, *Modularity* of Mind, Bradford Books.MIT Press, Cambridge, MA.

Franklin S, 1995, Artificial minds, A Bradford book, MIT press.

Franklin S and **Graesser,** 1997 "Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents," Proceedings of the Agent Theories, Architectures, and Languages Workshop, Berlin: Springer Verlag, 193-206.

Franklin S, 1997, *Autonomous Agents as Embodied A,I,* Cybernetics and Systems, 28:6(1997) 499-520 Available at http://www.cs.memphis.edu/~franklin/AAEI.html [Accesses on 16th October 2006]

Garner R, 1990, When children and adults do not use learning strategies: Toward a theory of settings. *Review of Educational Research, 60,* 517-529.

Gershenson C, ***2004,*** Cognitive paradigms: which one is the best? Cognitive Systems Research, *2004.*

Gat E, 1998, Three-layer architectures. In Kortenkamp, D., Bonasso, R. P., and Murphy, R., editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, pages 195-210. MIT Press, Cambridge, MA.

Gibson J, 1986, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates. New Jersey.

Goldstone R L, 1994, Precepts and Concepts Laboratory experiments, Indiana University Cognitive Science Program, Indiana, Available: [Online] www.psych.indiana.edu [Accessed 21st May 2005].

Goldstone R L and Barsalou L W, 1996, Reuniting Perception and Conception, the Perceptual Bases of Similarity and Rules.

Goldstone R L and Barsalou L W, 1998, Reuniting Perception and Conception, *Cognition*, 65, 231-262.

Griffiths R, 1995, *SOAR,* lecturer notes, University of Brighton.

Halpern D F, 1996, *Thought and knowledge: An introduction to critical thinking.* Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

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

Halpern D F, 1996, Thought and knowledge: An introduction to critical thinking. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Harnad S, 1990, The Symbol Grounding Problem.*Physica D* , 42:335--346.Hinton G.E and Sejnowski T J, 1986, Learning and relating in Boltzmann machines in parallel Distributed processing, ed.D.E.rumelhart, J.L.mcClelled, and the PDP Research group, 282-317.Cambridge,MA: MIT press.

Halter J, 2000, Metacognition, Available at http://coe.sdsu.edu/eet/Articles/metacognition/start.htm [Accessed 23rd August, 2007]

Hayes-Roth B A, 1993, Satisfying Cycle for Real-Time Reasoning in Intelligent Agents.

Hayes-Roth B, Lalanda P, Morignot, P. Pfleger, K. & Balabanovic, M.Plans and Behaviour in Intelligent Agents.1993.

Hochstein L, 2002, ACT-R, ACT-R Research Group at Carnegie-Mellon University.

Hoskins on R L and Hess and Fink R K, 1992, "A Decision Support System for Optimum Use of Fertilizers"

Jimmy W, 1996, *Animal cognition*, Available at http://www.en.wikipedia.org/wiki/Animal_cognition[Acessed t 16th, august, 2006].

John S and Edmund W, 1989, *Oxford English Dictionary (OED),* Second edition, Ed, Clarendon Press, twenty volumes, hardcover.

Kaelbling, 1986 Architecture for intelligent reactive systems. In Georgeff, M. P. and Lansky, A. L., Ed, *Reasoning about Actions &Plans - Proceedings of the 1986 Workshop*, pages 395-410. Morgan Kaufmann Publishers: San Mateo, CA.

Kaelbling, 1991, A situated automata approach to the design of embedded agents. *SIGART Bulletin*, 2(4):85-88.

Kaelbling and **Rosenschein, 1990,** Action and planning in embedded agents. In Maes, P., editor, *Designing Autonomous Agents*, pages 35-48. The MIT Press: Cambridge, MA.

Kaelbling L P, Littman M L and Moore A W, 1996, *Reinforcement learning: A survey,* Journal of Artificial Intelligence Research, 4:237 - 285.

Kennedy C M, 2003, "Distributed Reflective Architectures for Anomaly Detection and Autonomous Recovery", PhD thesis, University of Birmingham.

Kent M C, 2000, Functionalism available: http://web.grinnell.edu, [Accessed 20th May 2005].

Klingel H, 1990, Camels In Grzimek's Encyclopaedia of Mammals, Ed, S.P.Parker.New York: McGraw-Hill. **Volume 5,** pp.85-96.42.

Konolige K, 2002, *Saphira Robot Control Architecture*'s Internationals.

Laird J and **Rosenbloom S**, 1996, the evolution of the SOAR cognitive architecture. In D. M. Steier & T. M. Mitchell, (Eds.), *Mind Matters: A tribute to Allen Newell*. pp. 1-50, Mahwah, NJ: Erlbaum.

Laird J, Rosenbloom P and **Newell A**, 1986, chunking in SOAR: the anatomy of a general learning mechanism. *Machine Learning*, 1(1), 11-46.

Langley P, 1994, Cognitive Architectures and General Intelligent Systems Computational Learning Laboratory Stanford University, Stanford. Available at http://cll.stanford.edu/~langley/papers/arch.aimag.pdf. [Accessed 2nd June 2005].

Langley P, 2004, Cognitive architectures and the construction of intelligent agents. Pr*dings of the AAAI-2004 Workshop on Intelligent Agent Architectures* (pp.82).Stanford.http://cll.stanford.edu/research/ongoing/icaru[Accessed 16th, August, 2006].

Langley P, Arai S, & Shapiro D, 2004, Model-based learning with hierarchical relational skills. Proceedings of the ICML-2004, Workshop on Relational Reinforcement learning, Banff, Alberta, Available at http://cll.stanford.edu/research/ongoing/icarus/, [Accessed 17th August, 2006].

Langley P, Roger S, 2004, Cumulative Learning of hierarchical skills, Proceedings of the Third International Conference on Development of learning, SanDiego, CA, Available at http://cll.stanford.edu/research/ongoing/icarus/[Accessed 16th, August, 2006].

Lebiere C, Byrne M D, John R, Anderson, Qin Y, Bothell D and Douglass S A, 2004, *An integrated theory of the mind*.

Lees M, 2002, a History of the Tile world agent Testbed, School of Computer Science and Information Technology, University of Nottingham, Nottingham.

Lenat D, 1995, CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33-38.

Lewis S C, 2004, Computational Models of Emotion and Affect, PhD Thesis, Computer Science, University of Hull.

Livingston J A, 1996, Effects of metacognitive instruction on strategy use of college students. State University of New York at Buffalo.

Livingston J A, 1997, *Metacognition: An overview.* State University of New York at Buffalo. Available: http://www.gse.buffalo.edu/fas/shuell/cep564/Metacog.htm

Lugar G F and William A S, 1998, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", Addison Wesley Longman.

McCarthy J, 1959, ``Programs with Common Sense'', in *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, Her Majesty's Stationery Office, London. Reprinted in [McCarthy, 1990].

McCarthy J, 1962, ``Computer Programs for Checking Mathematical Proofs'', *Amer. Math. Soc. Proc. of Symposia in Pure Math.*, Vol. 5.

McCarthy J, 1995, making robots conscious of their mental states. *Machine Intelligence*, 15, 3-17.

McClelland J L, & Jenkins E, 1991, Nature, nurture and connections: Implications of connectionist models for cognitive development. In K. VanLehn (Ed.), Architectures for intelligence. Hillsdale, NJ: Erlbaum.

McFarland D, Bosser T, 1993, Intelligent behavior in animals and robots, MIT press, Cambridge, Massachusetts, London, England.

Miller G A, 1956, the magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological review, 63, 81-97.

Minsky M, and Oliver G. Selfridge, 1961, Learning in random nets, Proc. Fourth London Symposium. On Information Theory, Butterworth, Ltd., London. 1961.

Minsky M, 1980, K-lines, a theory of memory, Cognitive Science, 4, pp117-133, A theory of how structured hierarchies of knowledge may be represented in terms of networks that activate "partial brain states".

Minsky M, 1985, *The Society of Mind*, Cofounder of the artificial intelligence, MIT.

Minsky M, 1986, The Society of Mind, Simon & Schuster, 1986, ISBN: 0671657135.

Minsky M, 1990, "Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy", in Artificial Intelligence at MIT, Expanding Frontiers, Patrick H. Winston (Ed.),**Vol1**,,MIT Press, 1990, Available at http://web.media.mit.edu/~minsky/papers/SymbolicVs.Connectionist.html [Accessed 20th March 2007]

Minsky M, 1991, "A Response to Four Reviews of the Society of Mind", Artificial Intelligence, April 1991, **Vol. 48**, pp. 371-396.

Minsky M, 2000, "Attachments and Goals," in Hatano, G., Tanabe, H., and Okado, N., eds. Affective Minds. Amsterdam: Elsevier 2000.

Minsky M, 2006, The Emotion Machine, Draft book available at *http://web.media.mit.edu/~minsky/*

Nason S and Laird J E, 2004, SOAR_RL integrating reinforcement learning with SOAR. Available at http://www.sciencedirect.com/science[Accessed at 20th, august, 2006].

Newell and Simon, 1957, General Problem Solver RAND Corp/Carnegie Mellon University.

Newell A and Simon H, 1972 Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

Newell A, 1990, Unified Theories of Cognition. Cambridge, MA: Harvard University Press.

Newell A, 1992, *Unified theories of cognition and the role of SOAR*. In J.A.Michon & A.Akyurek (Eds), *SOAR: A Cognitive Architecture in Perspective*, 25-75.Dordrecht, the Netherlands: Kluwer Academic.

Newell A, Shaw J C, and Simon H A, 1957, "Empirical Explorations of the Logic Theory Machine", *Proceedings of the Western Joint Computer Conference*, pp. 218-239.

Neumann V J, 1946, "The Principles of Large-Scale Computing Machines", reprinted in *Ann. Hist. Comp.*, Vol. 3, No. 3, pp. 263-273.

Norman D A, 1980, *Twelve issues for Cognitive science*, Cognitive Science, 4, 1-32.

Nunes H A, 2001, Investigation of Motivation in Agents Using Five-Aside Football, MSc Research Thesis, Computer Science, University of Hull, November, 2001

Pearsall J, 1998, New Oxford Dictionary of English, Oxford.

Peirce W, 2003, Metacognition: Study Strategies, Monitoring, and Motivation, Available at: http://academic.pgcc.edu/~wpeirce/MCCCTR/metacognition.htm [Accessed 23rd August, 2007]

Pfeifer R, 1988, Artificial Intelligence models of emotion. In V. Hamilton, G. H. Bower & N. H. Frijda (Eds.), *Cognitive perspectives of emotion and motivation* (p. 287-321). Dordrecht: Kluwer.

Pfeifer R, 1993, The new age of the Fungus Eater. *Second European Conference on Artificial Life, ECAL 93*, University Libra de Bruxelles.

Pfeifer R, 1994, "The fungus eaters Approach" to emotion a view from artificial intelligence, AI Lab Institute for Informatics. University of Zurich-Irchel, Switzerland.

Pollack J B, 1990, *Recursive Distributed Representations*, Columbus.

Raja A, 2003,Meta-Level Control in Multi-Agent Systems, PhD Thesis, University of Massachusetts Amherst, http://www.sis.uncc.edu/~anraja/thesis.pdf, September 2003, [Accessed 15th May 2005].

Raja A, 2006, *Meta-Cognition in Coordinators*, North Carolina, USA.Available http://dair.sis.uncc.edu/ResearchWeb/Coordinators.htm, [Accessed 30th August 2006].

Rao A S and Georgeff M P, 1993,"A model-theoretic approach to the verification of situated reasoning systems" In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chamberey, France.

Raymond J, 2002, Dictionary of Psychology, Brynmor Routledge, New York and London.

Reynolds C 1986, Boids Background and Update, Available at http://www.red3d.com/cwr/boids/ [Accessed on 25th September 2006].

Rich E and Knight K, 1991, "Artificial Intelligence", International edition, Carnegie Mellon University.

Richard Forsyth, 1998, *Expert Systems Principles and Case Studies*, Chapman and Hall Computing.

Ritter F E and Shiskowskia N, 2003, frequently Asked Questions List Available http://ritter.ist.psu.edu/projects/act-r-faq/act-r-faq.html (Accessed 5th Nov 2005).

Robert S A, Freed R A, Ritter F E, 2004, *Specifying ACT-R Models of User Interaction With A Goms Language*, North Carolina State University, Raleigh, NC 27695, USA.

Roberts M J and Erdos G, 1993, *Strategy Selection and Metacognition,* **Educational Psychology**, *13,259*-266.

Russel and **Norvig,** 1995, Artificial intelligence: a modern approach, University of California, Berkeley.

Rutledge-Taylor M F, 2004, Can ACT_R Realize "Newell's Dream", Ottawa, Canada.

Ryder J, 2004, developing a Multitask Cognitive Agent Using the COGNET/iGEN Integrative Architecture, Lower Gwynedd, PA.

Saphira, 1996, the Saphira Architecture A Design for Autonomy.

Scheid K, 1993, *Helping students become strategic learners: Guidelines for teaching.* Cambridge MA Brookline Books.

Scruffy, Patrick H.Winston (Ed.), Vol.1, MIT Press, 1990.Reprinted in AI Magazine, Summer 1991.

Selfridge O, 1959, Pandemonium: A paradigm for learning. In Symposium on the mechanization of thought processes.

Sharkey N E, 1997, The new wave in robot learning, Robotics and Autonomous Systems, Dec 1997.

Singh P, 2002a, the Panalogy Architecture for Commonsense Computing. PhD Thesis Proposal. Department of Electrical Engineering and Computer Science, MIT. Available at: http://web.media.mit.edu/~pus[Accessed 10th November 2005].

Singh P, 2002b, The public acquisition of commonsense knowledge. Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access. Palo Alto, CA, Available at: http://web.media.mit.edu/~pus[Accessed at 20th Nov 2005].

Singh P, 2003b, A preliminary collection of reflective critics for layered agent architectures. Proceedings of the Safe Agents Workshop (AAMAS 2003). Melbourne, Australia, Available at: http://web.media.mit.edu/~pus[Accessed 20th Dec 2005].

Singh P and Minsky, M, 2003, Architecture for combining ways to think. Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems. Cambridge, MA.

Singh P, 2003, Examining the Society of Mind, Computing and Informatics, 22(5):521-543. MIT Media lab. [Accessed 5th Feb 2006].

Singh P, Minsky M and Eslick I, 2004, Architecture for computing common sense. BT Technology Journal, 22(4), 201-210, [Accessed 5th March 2007].

Singh P, 2004, Computing Commonsense, MIT, Media lab. Available at: http://web.media.mit.edu/~pus [Accessed 10th Oct 2005].

Singh P, 2005, EM-ONE: Architecture for Reflective Commonsense Thinking PhD Thesis. Available at http://web.media.mit.edu/~pus. [Accessed 22nd Nov 2005]

Sloman A, 2001, *Varieties of Affect And The Cogaff Architecture Schema*, University Of Birmingham. Available http://www.cs.bham.ac.uk/research/cogaff/. [Accessed 22nd July 2005].

Sloman A, 2001, *Beyond Shallow Models of Emotion*, University Of Birmingham. Available at http://www.cs.bham.ac.uk/research/cogaff/. [Accessed 22nd July 2005].

Sloman A, 2002, *How To think About Cognitive Systems: Requirements And Design*, Available at http://www.cs.bham.ac.uk/research/cogaff/. [Accessed 21st July 2005].

Sloman A, 2003, *The Architecture Of Brain And Mind, University* Of Birmingham Available [Online] http://www.cs.bham.ac.uk/research/cogaff/. [Accessed at 21 July 2005].

Sternberg R J, 1984, What should intelligence tests test? Implications for a diarchic theory of intelligence for intelligence testing. *Educational Researcher, 13* (1), 5-15.

Sternberg R J, 1986a, Inside intelligence. *American Scientist, 74*, 137-143.

Sternberg R J, 1986b, *Intelligence applied.* New York: Harcourt Brace Jovanovich, Publishers.

Stillings N, 1995, *Cognitive Science.* Second edition. Cambridge, MA: MIT Press

Susca M, 2006, Measure of cognition and affect, National library of medicine, United States. Available at http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve [Accessed 2nd September, 2006].

Sutton R S and Barto A G, 1998, Reinforcement Learning: An Introduction, the MIT Press Massachusetts.

Sutton R S, 2001, Frequently Asked Questions about Reinforcement Learning, Available at http://cat.mm.rpi.edu/~rsun/RL-FAQ.html [Accessed 18th August 2006].

Takadama K, Fujita H, 2004, Toward Guidelines for Modelling Learning Agents in Multiagent-Based Simulation: Implications from Q-Learning and Sarsa Agents. MABS 2004: 159-172.

Takadama K, Yutaka L, Suematsu, Sgimotuo N, Nawa **N E, Shimohara K**, 2003, Towards Verification and Validation in Multiagent-Based Systems and Simulations: Analyzing Different Learning Bargaining Agents, 26-42.

Takadama K, Kawai T, Koyama Y, 2006, Can Agents Acquire Human-Like Behaviours in a Sequential Bargaining Game? - Comparison of Roth's and Q-Learning Agents, 156-171.

Taylor S, 1999, Better learning through better thinking: Developing students' metacognitive abilities. *Journal of College Reading and Learning, 30*(1), 34ff. Retrieved November 9, 2002, from Expanded Academic Index ASAP.

Thorndike E L, 1911, Animal intelligence: Experimental studies. New York: Macmillan.

Toda M, 1962, *Design of a Fungus-Eater, Behavioral Science*, 7, 164-183. (Reprinted in Toda, 1982, 100-129).Toda, M. (1982).Man, robot and society. The Hague: Martinus Nijhoff Publishing.

Trafton J G, & Bogacz S, 2005, Understanding Dynamic and Static displays: Using Images to Reason Dynamically. Cognitive Systems Research. 6(4), 312-319

Turing A, 1950, Computing machinery and intelligence. Mind, 59(236), 433-460.

Vijayakumar M V and **Davis D N,** 2007, Metacognition, Agents, Animats, and Society of Mind, AISB 2007.

Vijayakumar M V and Davis D N, 2008, Metacognition in a Society of Mind, IASTED (AIA 2008), Innsbruck, Austria.

Wade N J and **Swanston M T,** 2001, Visual Perception, An Introduction 2nd edition.

Wahl S and **Spada H, 2000,** *Children's Reasoning about Intentions,* Beliefs and Behaviour. *Cognitive Science Quarterly*, 1, 5-34.

Waldrop M, 1988, *Toward a unified theory of cognition. Science, 241*, 27-29; SOAR: a unified theory of cognition? *Science, 241*, 296-298

Wang P, 2005, Introduction to Artificial Intelligence, Temple University. Available: [Online] http://www.cis.temple.edu/~pwang/203-AI/CIS203.html [Accessed 7th November 2005].

Wehrle T, 1994, New fungus eater experiments. In P.Gaussier, & J.D.Nicoud (Eds.), From perception to action (pp.400-403).Los Alamitos IEEE Computer Society Press. (http://www.unige.ch/faps e/emotion/members/wehrle/fungus.htm000 [Accessed 15th April 2006]

Wilson R A and **Keil F C,** 1999 Editors, the MIT Encyclopaedia of the Cognitive Sciences, MIT Press.

Winograd T & Flores F, 1986, Understanding Computers and Cognition, Addison-Wesley, 1986.

Young R M, 2001, Production Systems in Cognitive Psychology, In N. J. Smelser & P. B. Baltes (eds.), *International Encyclopaedia of the Social and Behavioural Sciences.* Pergamon, in press

Zalta E N, 2005, *Stanford Encyclopaedia of Philosophy [online],* available: www.platostanford.edu, [Accessed 5th May 2005].

Appendices

Appendix A

A. 1. Reactive agents design (Design Continued)

Reactive agent design algorithm

Goal based behaviour towards Fungus

Find the nearest Fungus by distance formula,

Select the direction towards nearest fungus,

Move towards Fungus type direction | left| right |Up| down.

Goal based behaviour towards resource

(ore, golden ore and crystal)

Find the nearest resource by their distance,

Select the direction towards nearest resource,

Move towards resource direction | left| right |Up| down.

Reactive agent design algorithm

Goal based behaviour towards Crystal

Find the nearest Crystal by distance formula,

Select the direction towards nearest Crystal,

Move towards Crystal type direction | left| right |Up| down.

Goal based behaviour towards resource

(ore, and golden ore)

Find the nearest ore type by their distance,

Select the direction towards nearest ore,

Move towards oree direction | left| right |Up| down.

Reactive agent design algorithm

Goal based behaviour towards Medicine

Find the nearest Medicine by distance formula,

Select the direction towards nearest Medicine,

Move towards Medicine direction | left| right |Up| down.

Goal based behaviour towards Unit

(ore, and golden ore)

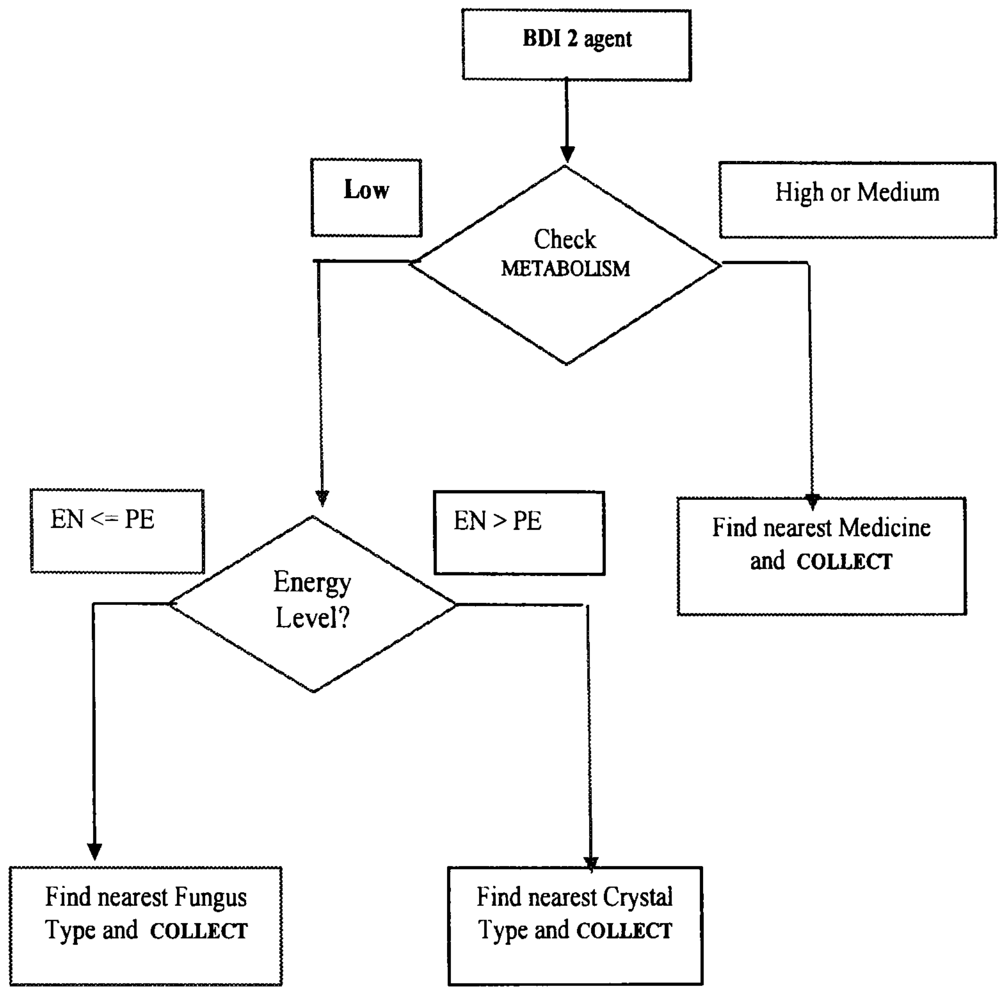
Find the nearest Unit type by their distance,

Select the direction towards nearest unit,

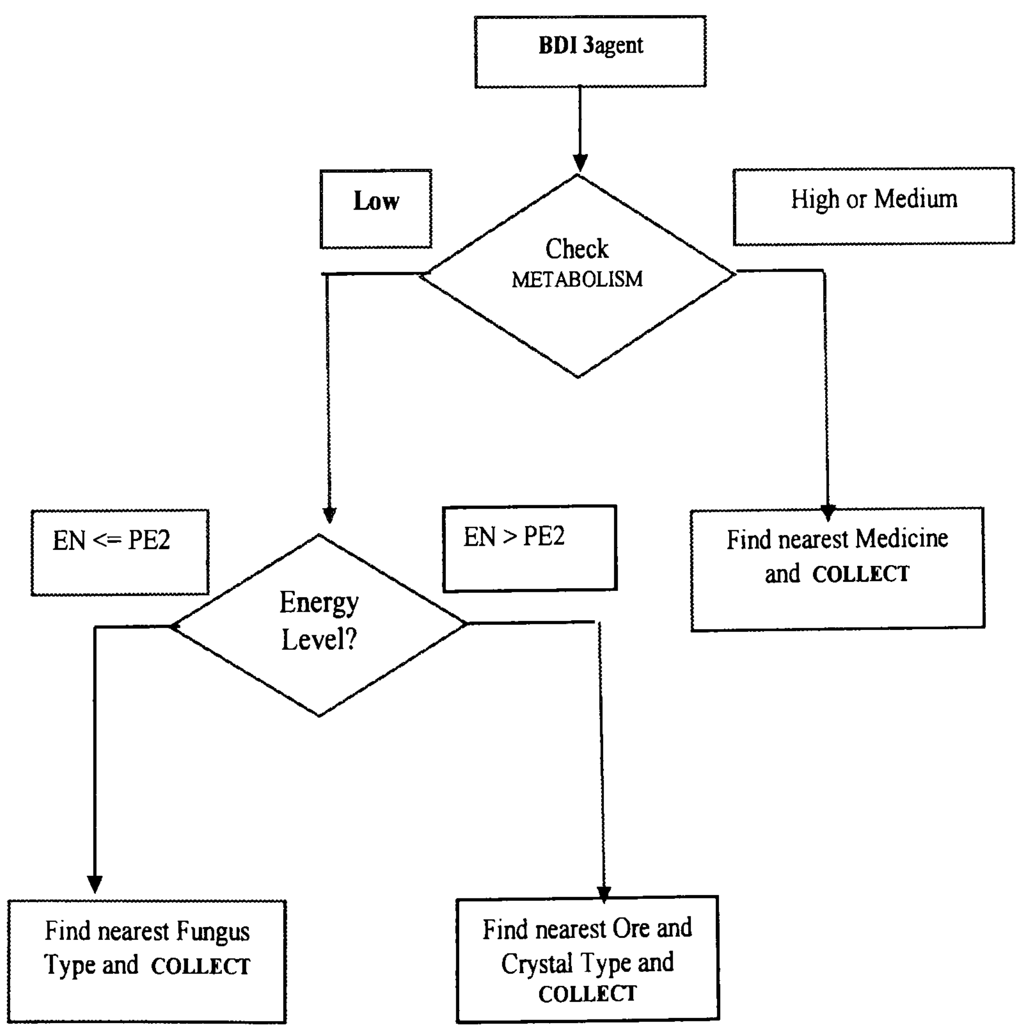
Move towards unit direction | left| right |Up| down.

A. 2. Second BDI Model Structure

(Expanded from the BDI model, different structures)



A. 3. Third BDI Model Structure



BDI 3(Third) Model

(1)Metabolism > Low,

Then searches the nearest medicine to collect to lower the metabolism by their Reactive mechanism. Uses the Reactive Medicine

Uses FSM and up|left|right|Down,

Find the nearest Medicine by their distance,

Select the direction towards nearest Medicine,

Move towards Medicine direction | left| right |Up| down.

(2)Energy Level <= Predict_think2_energy,

The agent desire to move towards to fungus to avoid the hunger condition or their death (Physiological oriented) uses the Reactive Fungus

Uses FSM and up|left|right|Down,

Find the nearest Fungus by distance formula,

Select the direction towards nearest fungus,

Move towards Fungus type direction | left| right |Up| down,

Predict_think2_energy = Predict_think1_energy + distance of nearest fungus/20 * energy use use. (Two levels ahead thinking of energy level).

(3)Energy Level > Predict_think2_ energy (Dynamic)

Reactive Ore (Goal based behaviour towards Ore),

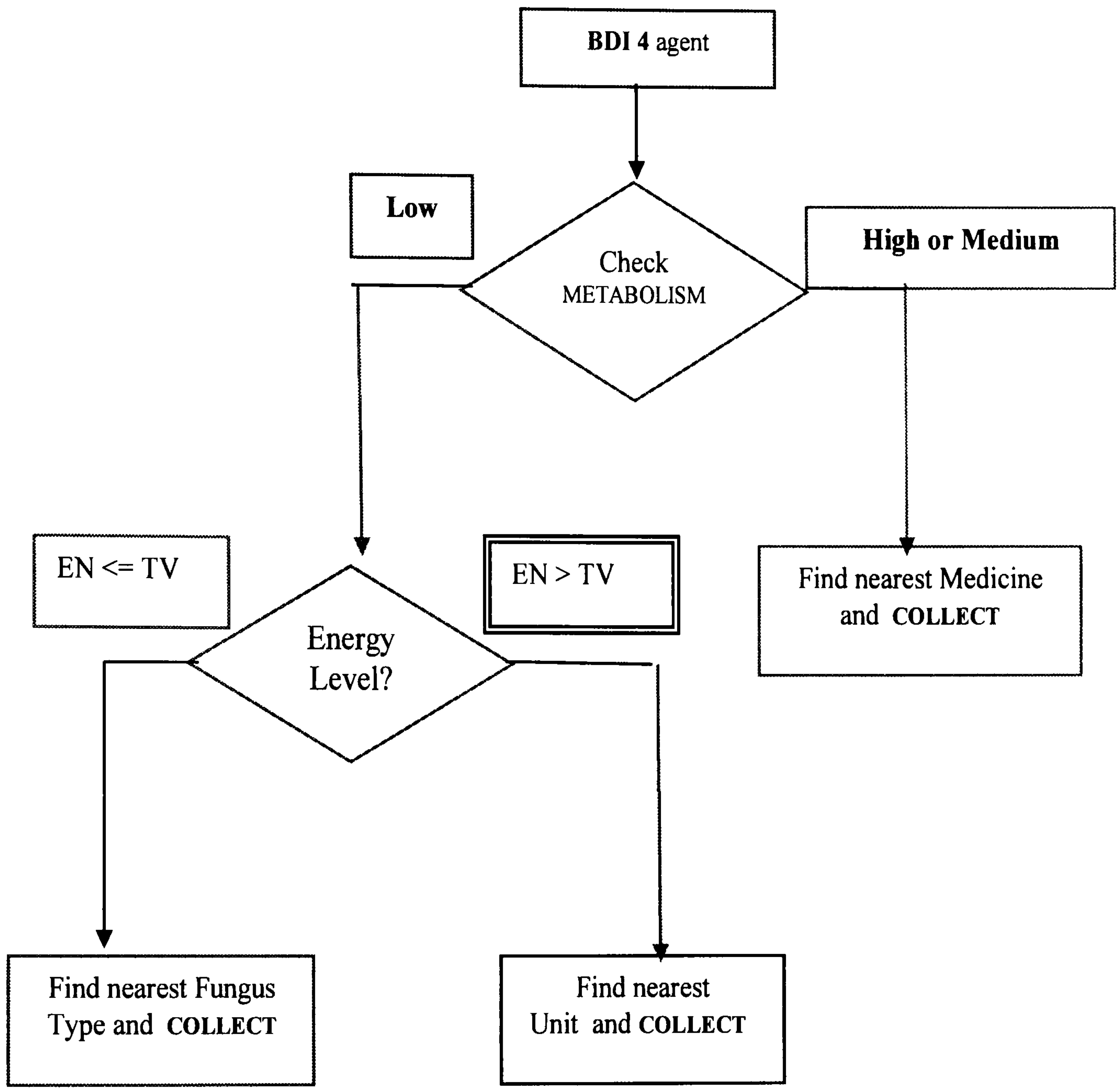
Uses FSM and up|left|right|Down,

Find the nearest Ore type by their distance,

Select the direction towards nearest Ore and crystal type,

Move towards Ore and crystal type direction | left| right |Up| down.

A. 4. Fourth BDI Model Structure



BDI 4(Fourth) Model

(1)Metabolism > Low,

Then searches the nearest medicine to collect to lower the metabolism by their Reactive mechanism. Uses the Reactive Medicine

Uses FSM and up|left|right|Down,

Find the nearest Medicine by their distance,

Select the direction towards nearest Medicine,

Move towards Medicine direction | left| right |Up| down.

(2)Energy Level <= 30 (threshold energy)

The agent desire to move towards to fungus to avoid the hunger condition or their death (Physiological oriented) uses the Reactive Fungus

Uses FSM and up|left|right|Down,

Find the nearest Fungus by distance formula,

Select the direction towards nearest fungus,

Move towards Fungus type direction | left| right |Up| down,

(3)Energy Level > 30

Reactive Ore (Goal based behaviour towards Ore),

Uses FSM and up|left|right|Down,

Find the nearest Ore type by their distance,

Select the direction towards nearest unit

Move towards nearest unit type direction | left| right |Up| down.

A. 5. Finding Smart Energy Levels

Finding decision boundaries for different smart models

Predict energy (Finding the decision boundary):-

Find the nearest fungus,

Find the nearest distance between agent and Fungus,

Find the agent needed energy by using their metabolism state and

Energy required, Predict energy is Distance/20 * Energy Use.

predict_think1_energy (Predict_thinkl_energy):-

Find the nearest fungus, Find the nearest distance between agent and

Fungus, Find the agent needed energy by using their metabolism state and

Energy required Predict_thinkl_energy is Predict energy + Distance/20 *

Energy Use (One level ahead thinking of energy level).

predict_think2_energy (Predict_think2_energy):-

Find the nearest fungus,

Find the nearest distance between agent and Fungus,

Find the agent needed energy by using their metabolism state and

Energy required, Predict_think2_energy is Predict_thinkl_energy +

Distance/20 * EergyUse (Two level ahead of thinking energy level).

A. 6. Q-learning Agent Structure

Q-Learning Mechanism in Fungus World Testbed

Store_qvalues

Calculate X, Y if perform move in Direction,

Calculate distance to Resource. Reactive Fungus, Ore, crystal

Or medicine from NewX, NewY,

Get an old Q_value,Update locations that will get different Q values,

(Q value may increase or decrease)

By using the relation Delta = Old + ((Distance - New distance)/

Distance),Stores a Q values with respect to locations by order (sorting).

A decision on what move can be made by using following design

Get list of Value-Action pairings for current state

%findall (Q-Dir, qvalue ([X, Y], Dir, Q), List),

Order them - list Order will have smallest Q at Head

%order (List, Order),

Reverse list Order to Get Largest Value at top,

The associated Direction is best,

According to current Reward policy,

Direction to go in given by BestDir,

% reverses (Order, [BiggestQ-BestDir|_restOfList]).

Norm 2

Collected (Ore) = 0.

Norm1 (OreBDI) = 0.75.

Collected (Ore) > 0,

Norm1 (OreBDI) = Perceived (Ore)/Collected (Ore).

Collected ore is Ore1,

Collected (Crystal) = 0,

Norm1 (Crystal) = 0.75.

Collected (Crystal) > 0,

Norm1 (Crystal) = Perceived (Crystal)/Collected (Crystal).

Collected (Ore) + Collected (Crystal) = 0,

Norm1 (Ore_CrystalBDI) = 1.

(Collected (Ore) + Collected (Crystal) > 0,

Norm1 (OreCystalBDI) = Perceived (Cystal+Ore)/Collected (Cystal+Ore).

Norm 3

Collected (Ore) = 0,

⇒ Norm2 (OreBDI) = 0.75.

Collected (Ore) > 0,

Norm1 (OreBDI) = Perceived (Ore)/Collected (Ore).

Collected ore is Ore1 + golden ore /2,

Collected (Crystal) = 0,

⇒ Norm2 (Crystal) = 0.75.

Collected (Crystal) > 0,

Norm2 (CystalBDI) = Perceived (Crystal)/Collected (Crystal).

Collected_Cystal is collected Crystal /2,

(Collected (Ore) + Collected (Crystal) = 0,

Norm2OreCrystalBDI) = 1.

(Collected (Ore) + Collected (Crystal) > 0,

Collected (Ore Crystal + Ore) is collected Ore + collected _Golden Ore + Collected Crystal /2.

Norm2 (OreCystalBDI) = Perceived (Cystal+Ore)/Collected (Cystal+Ore).

Resource Rule set

Performance (OreBDI) = Ore + Golden ore + Age.

Affect (OreBDI) = Norm (OreBDI)/ Performance (OreBDI).

Performance (Crystal BDI) = Crystal + Age.

Affect (Crystal) = Norm (Crystal)/ Performance (Crystal).

Performance (Ore_CrystalBDI) = Ore + Golden ore + Crystal + Age.

Affect (Ore_CrystalBDI) = Norm (Ore_CrystalBDI) / Performance (Ore_CrystalBDI).

Affect rule2

If(Metabolism = Low), Need(Medicine) = 0,

If (Metabolism = Medium), Need(Medicine) = 0.5,

If(Metabolism = High), Need(Medicine) = 1.0,

If(Energy_ level < Decision_boundry),

Need(Energy_level) = 0.75,

If(Energy _level >= Decision_Boundry),

Need(Energy_level) = 0,

Need(Medicine) = 0,

Need(Energy_level) = 0,

Performance_crystal_BDI is Crystal1 + Age1,

Norm_crystalBDI > Norm_crystalBDI1,

Norm_crystalBDI > Norm_crystalBDI2,

Affect_crystal_BDI is Norm_crystalBDI / Performance_crystal_BDI.

Performance_crystal_BDI is Crystal1 + Age1,

Norm_crystalBDI1 > Norm_crystalBDI,

Norm_crystalBDI1 > Norm_crystalBDI2,

Affect_crystal_BDI is Norm_crystalBDI1 / Performance_crystal_BDI.

Performance_crystal_BDI is Crystal1 + Age1,

Affect_crystal_BDI is Norm_crystalBDI2 / Performance_crystal_BDI .

Affect rule 3

If(Metabolism = Low), Need(Medicine) = 0,

If (Metabolism = Medium), Need(Medicine) = 0.5,

If(Metabolism = High), Need(Medicine) = 1.0,

If(Energy_ level < Decision_boundry),

Need(Energy_level) = 0.75,

If(Energy _level >= Decision_Boundry),

Need(Energy_level) = 0,

Need(Medicine) = 0,

Need(Energy_level) = 0,

Performance_ore_crystal_BDI is Ore1 + Golden_ore1 + Crystal1 + Age1,

Norm_ore_crystal_BDI > Norm_ore_crystal_BDI1,

Norm_ore_crystal_BDI > Norm_ore_crystal_BDI2,

Affect_ore_crystal_BDI is

Norm_ore_crystal_BDI / Performance_ore_crystal_BDI .

Performance_ore_crystal_BDI is Ore1 + Golden_ore1 + Crystal1 + Age1,

Norm_ore_crystal_BDI1 > Norm_ore_crystal_BDI,

Norm_ore_crystal_BDI1 > Norm_ore_crystal_BDI2,

Affect_ore_crystal_BDI is

Norm_ore_crystal_BDI1 / Performance_ore_crystal_BDI .

Performance_ore_crystal_BDI is Ore1 + Golden_ore1 + Crystal1 + Age1,

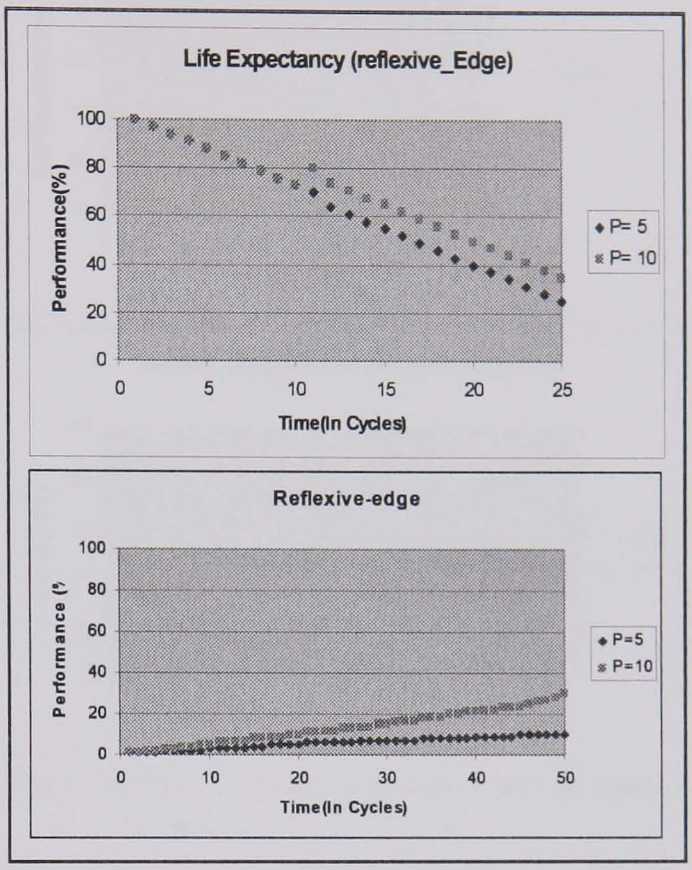
Affect_ore_crystal_BDI is

Norm_ore_crystal_BDI2/ Performance_ore_crystal_BDI.

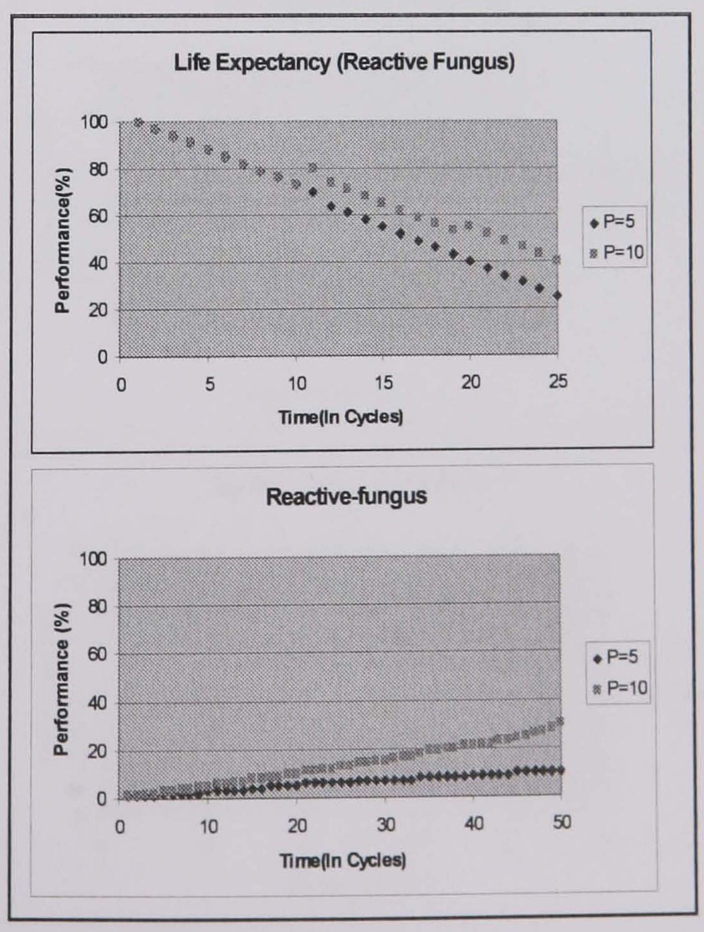
Appendix B

B. 1. Perception of 5 and 10 level with 5 agents

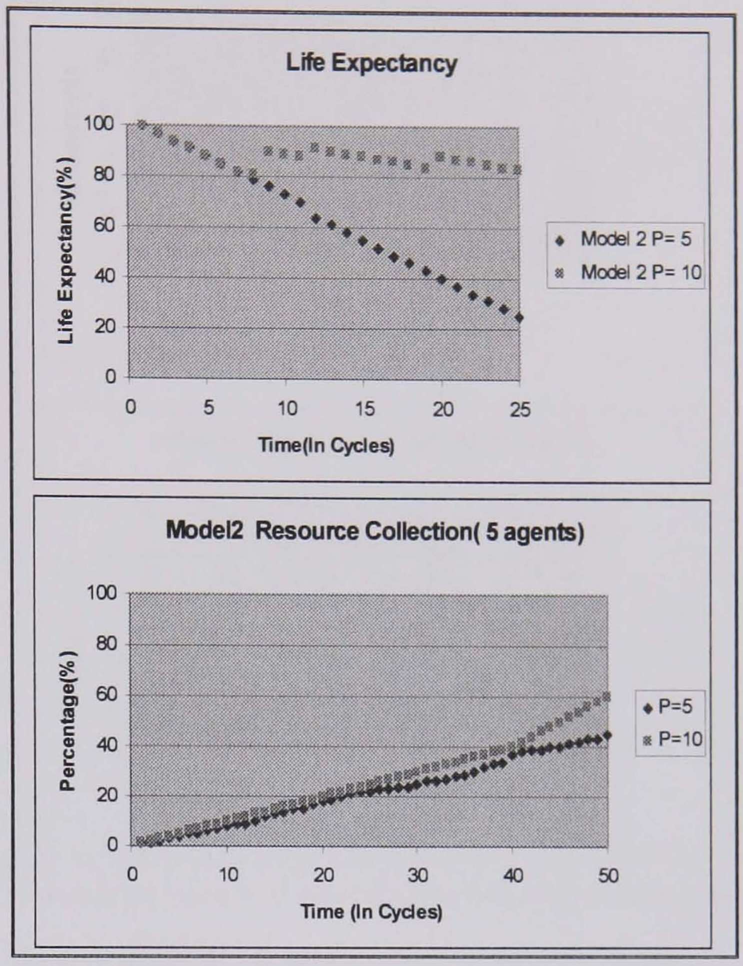
B 1.1 Reflexive-Edge with 5 agents comparing Perceptual level 5 V/s 10



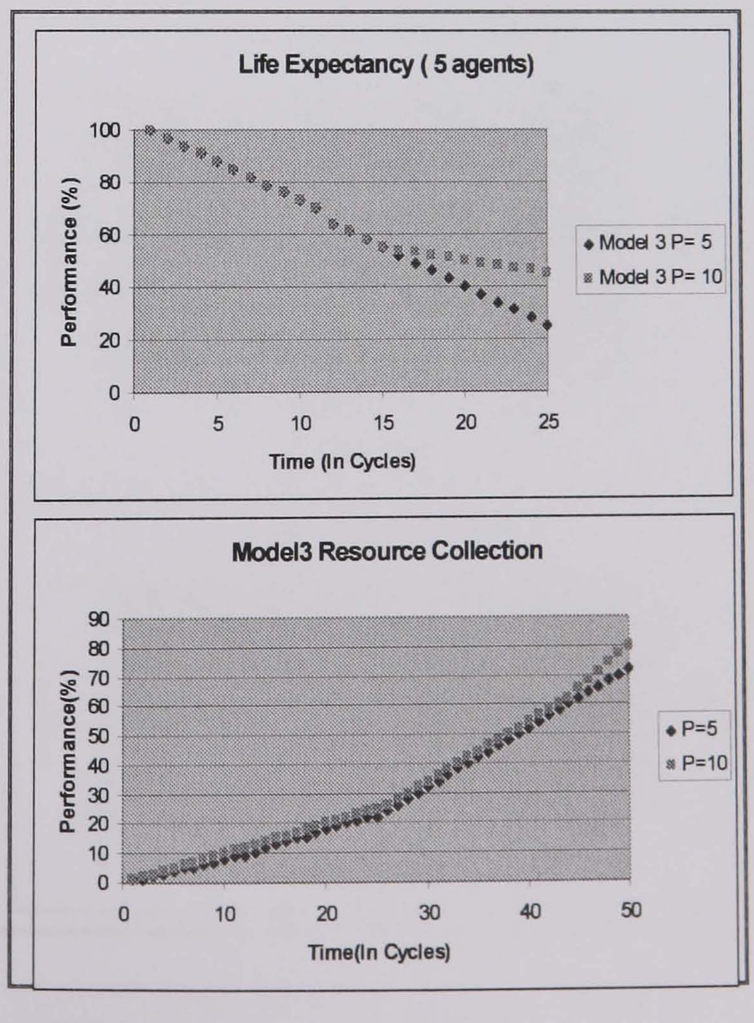
B 1.2 Reactive fungus with 5 agents comparing Perceptual level 5 V/s 10



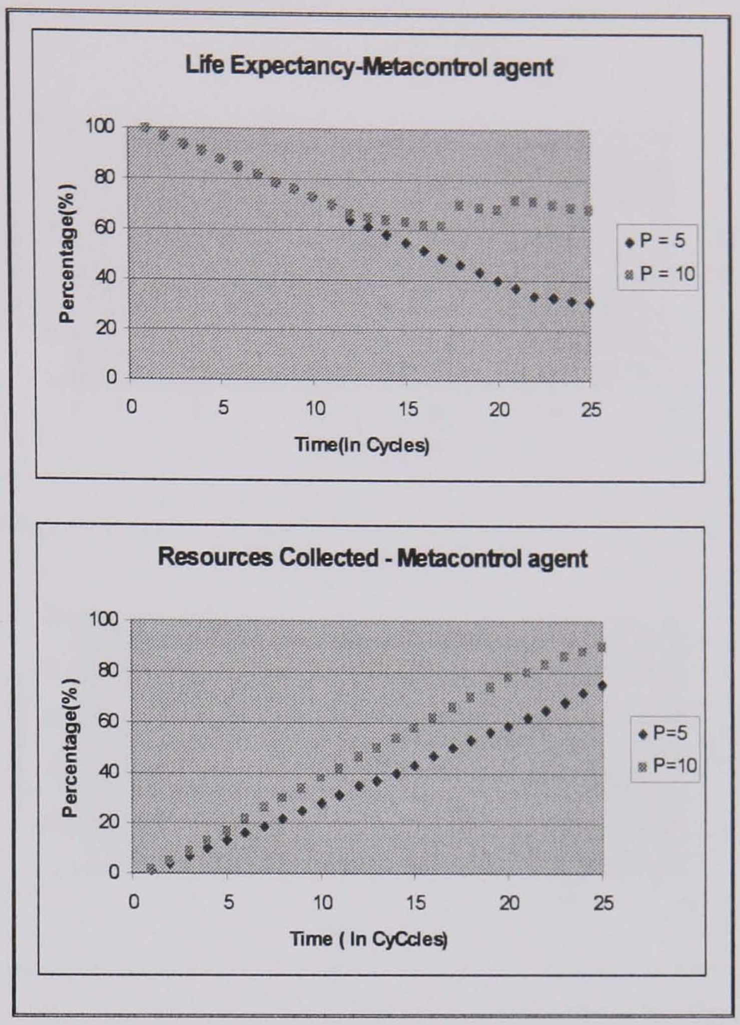
B 1.3 Model2 (BDI 2) with 5 agents comparing Perceptual level 5 V/s 10.



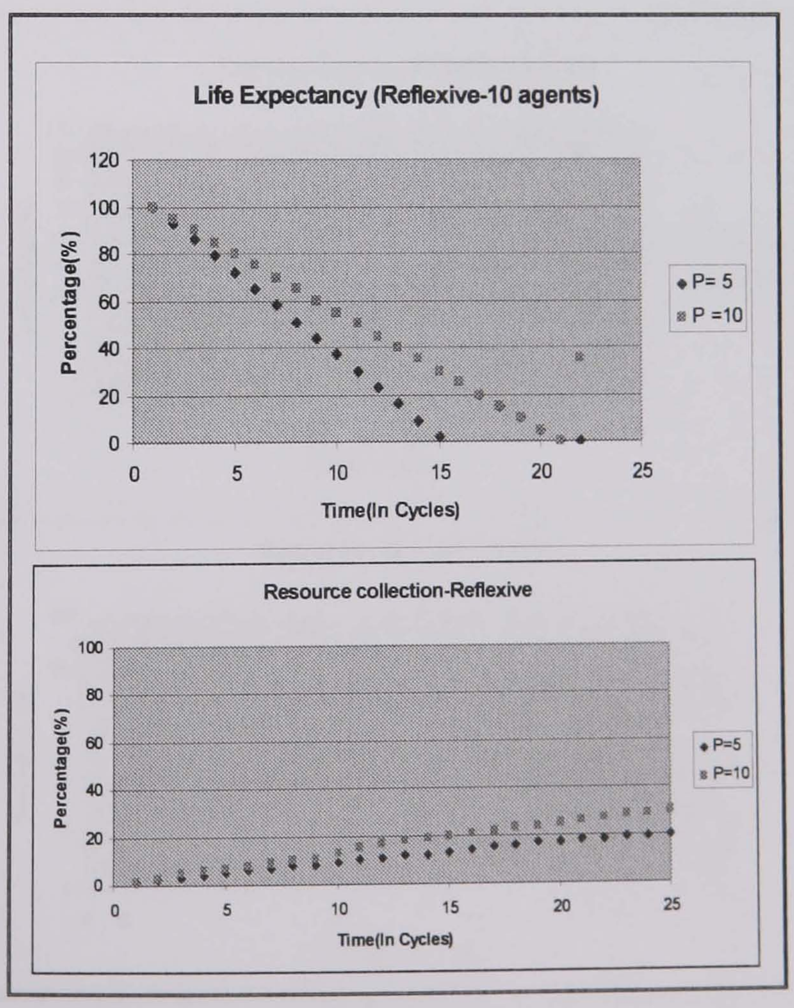
B 1.4 Model3 (BDI 3) with 5 agents comparing Perceptual level 5 V/s 10.



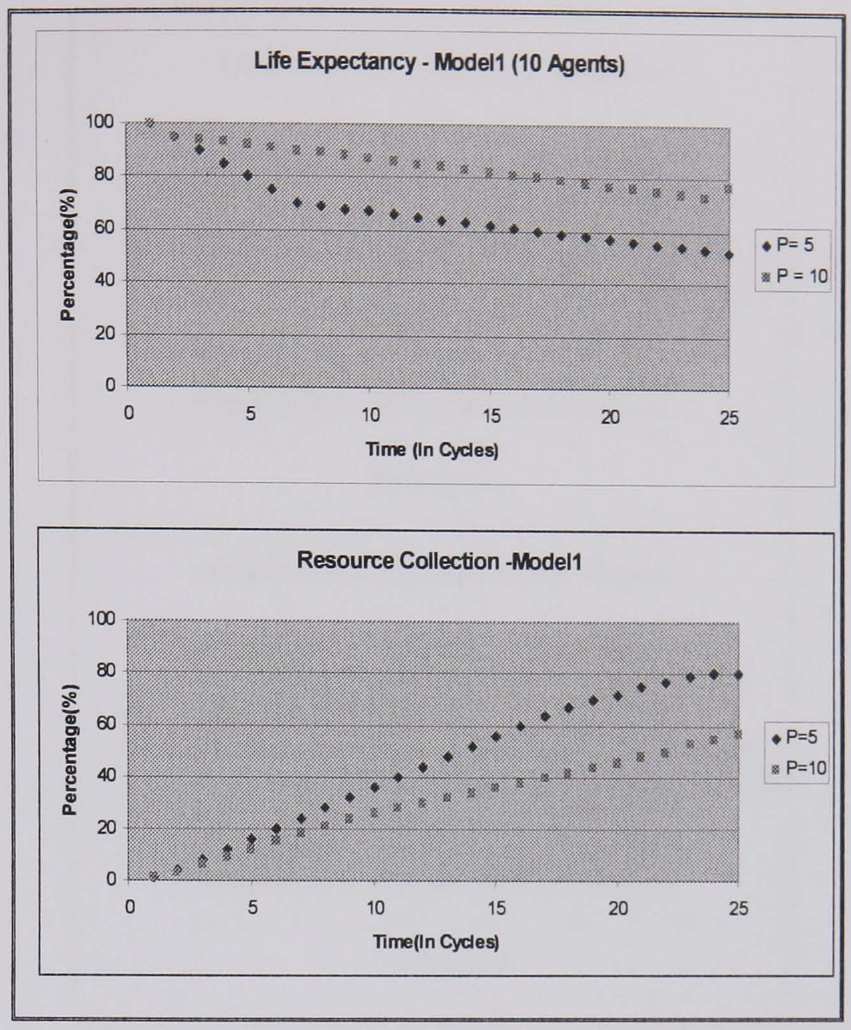
B 1.5 Metacontrol with 5 agents comparing Perceptual level 5 V/s 10.



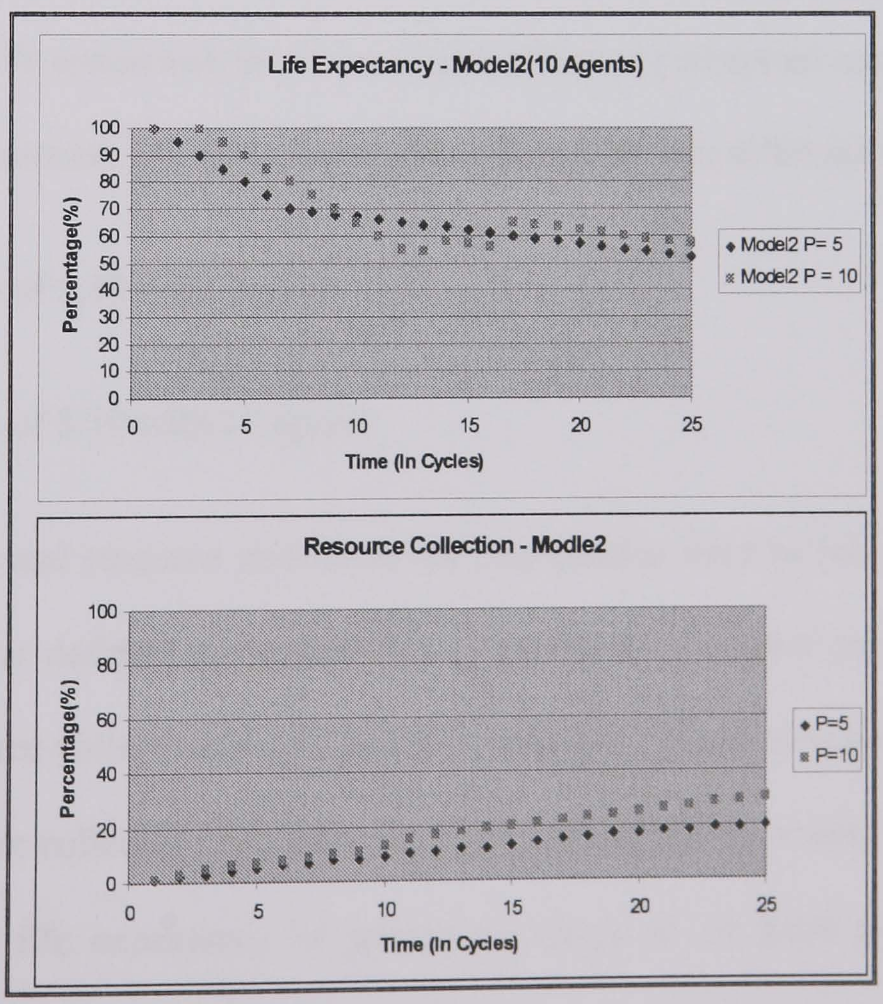
B. 2. Perception of 5 and 10 level with 10 agents



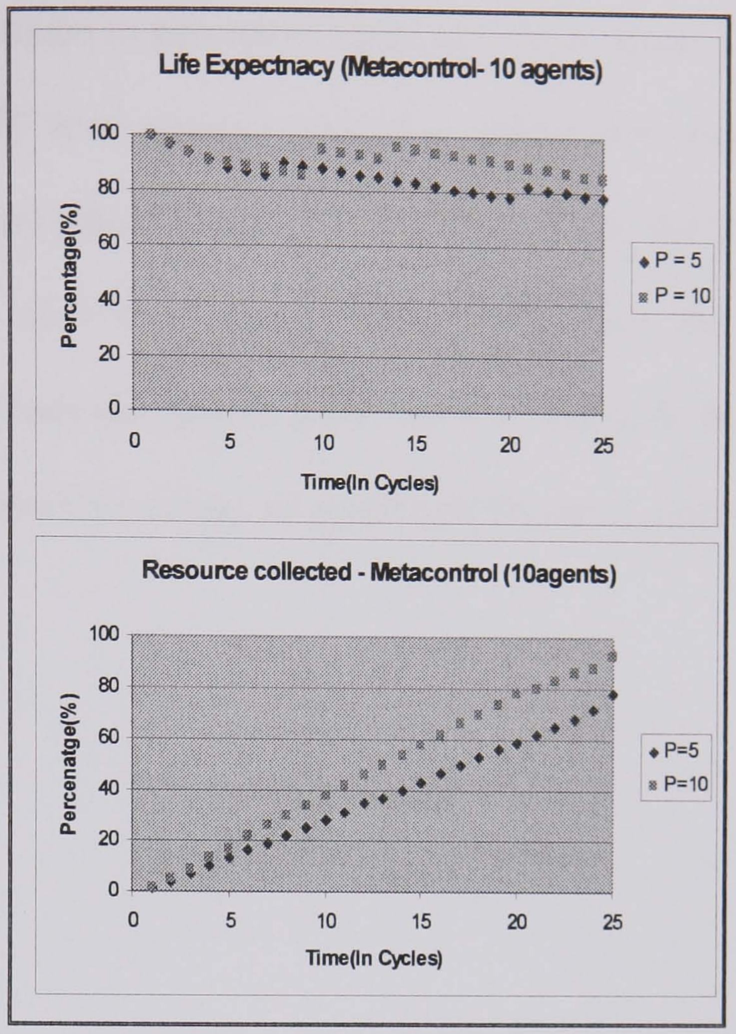
B 1.6 Model1 (BDI 1) with 10 agents comparing Perceptual level 5 V/s 10.



B 1.7 Model2 (BDI2) with 10 agents comparing Perceptual level 5 V/s 10.



B 1.8 Metacontrol with 5 agents comparing Perceptual 5 V/s 10



B. 3. Conclusion

This experiment is conducted for observing an effect of perceptual range on each agent or agent's performance. The experiment is conducted for two different situations.

Perceptual range of 5:10 with 5agents

Perceptual range of 5:10 with 10 agents

Life expectancy and resource collection are two metrics used in this experiment. The life expectancy is defined as the survival of agents in a testbed for fixed energy or nutrients. Resource collection is defined as the number of resources such as ore; golden ore and crystal are collected in given a time cycle. Based on the results obtained in these experiments the life expectancy of perceptual range of 10 level agents (society of agents) has higher expectancy than perceptual range of 5 level agents. Perceptual range of 10 level agents managed to have higher energy level after the maximum cycles (i.e.25 cycles), as compare to perceptual range of 5 level agents (10 to 20% more).The perceptual range of 10 level agents managed to collect more number of pieces of ore and crystals (resources) collected after the maximum cycles (i.e.25 cycles), as compare to perceptual range of 5 level agents. It concludes that, the perceptual range or perceptual level increases the agent's belief set for sensing in the environment. If the perceptual level increases the society of agent's performance also increases.