Emergence of Congruent Behaviour by Implicit Coordination of Innate and Adaptive Layers of Software Agents

Ranasinghe Aratchige Chaminda Ranasinghe

Thesis submitted for partial fulfilment of the requirements for the Degree of Doctor of Philosophy

University of Colombo, Sri Lanka.

2008

Declaration

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief it does not contain any material previously published or written or orally communicated by another person except where due reference is made in text.

Signature of Candidate

To the best of our knowledge the above particulars are correct.

Names of Supervisors **Signature**

Dr. A.P. Madurapperuma
Senior Lecturer,
Faculty of IT.
University of Moratuwa,

Dr. N.D. Kodikara
Head / Dept. of Information Systems Engineering,
University of Colombo, School of Computing (UCSC),
University of Colombo.

Abstract

The emergence of behavioural and structural congruence based on simple local interactions of atomic units is a fascination to the scientific community across many disciplines. The climax of behavioural congruence and emergence of behaviour is exemplified by the community life-style of ants. Each individual ant possesses the capability to solve only part of the overall puzzle while aggressively communicating in primitive methods with the spatially related neighbours to produce emergent behaviour. Hence, ant colonies have evolved means of performing collective tasks, which are far beyond the capabilities of their individual structures. The consensus is that comprehension of emergent complexity in insect colonies such as ants would serve as a good foundation for the study of emergent, collective behaviour in more advanced social organisms. As evidence of structural congruence, the realisation of a phenotype from a single genotype during the embryonic development, and some theories of the human mind that describe intelligence as a synergy of mindless constituents provide insight to the emergence theories. These facts argue that there exists a fundamental theory for structural and behavioural congruence that is yet to be discovered.

The primary hypothesis of the research is that the constituent atomic actions of a complex behaviour could be successfully coordinated by collaborative and autonomous agents that are loosely coupled through implicit communication to demonstrate emergent congruent behaviour in dynamic environments. The resulting congruent behaviour could be further optimised by using a hybrid learning approach that models adaptive behaviour on a static foundation of innate elementary behaviour.

The AAANTS model was conceptualised and implemented as a platform to represent the biologically inspired learning model to test the research hypothesis. The model encompasses aspects related to coordination, knowledge representation and adaptation by reinforcements. Two experimental domains were implemented on this platform, related to foraging in a grid-world and robotic arm movements to grab and push an object. The experiments demonstrated relative improvements in achieving behavioural congruence using the AAANTS model in relation to the traditional Monte-Carlo based methods. The research has also identified further improvements to the model that would enhance the capabilities in achieving higher levels of behavioural congruence in heterogeneous application domains.

Dedicated to my family…

My Wife Chiranthi and My Sons Yenula and Savinu

Acknowledgement

It was a difficult choice to decide on an educational path that leads to a Ph.D as an orthogonal objective while continuing on a full-time career in the industry. The numerous challenges along the path were diluted only by guidance, understanding and sacrifice of a few important individuals and institutions. There are many to thank for the last seven years of effort and sincerely hope I have done justification to the deserved in the following note of appreciation.

I should first and foremost state that this research would have never been started without the encouragement from Dr. Ajith Madurapperuma as the supervisor of the research. He had been a pillar of support and encouragement throughout the seven years. It would be impossible to find a supervisor of this calibre that would take personal interest in research activities and I firmly believe that the research would have never realised in void of his continued selfless commitments. Subsequent to the resignation of Dr. Madurapperuma from UCSC, Dr N.D. Kodikara represented my research as the supervisor. His contribution was invaluable in coordinating the research approval and submission process. His guidance and the level of independence given to drive the research are greatly appreciated.

I am in debt to my parents for all the sacrifices and commitments in all aspects of my life as a child and specially the luxury of offering independence of decision and choice as an adult. If I have the energy to be different, unorthodox and risk-taking, then it should be attributed to my parents. I am also in debt to my wife's parents in assisting me throughout the research period in coordinating and assisting in the family related matters. If without their sacrifices it would have been very difficult to allocate personal time to research activities.

As I mentioned earlier, this research was conducted with a full-time involvement within the IT industry. I have taken limited periods of leave with the approval and understanding from the work places during the last 7 years. Hence, I sincerely convey my gratitude to all my superiors, peers and subordinates who have encouraged and assisted in this endeavour in the companies - Millennium Information Technologies Ltd, Interblocks Ltd and Dialog Telekom PLC. Further, I greatly appreciate the assistance given by Dr. Keerthi Premadasa with words of wisdom and always providing refuge and facilities at IIT to conduct research work. After his departure from the institute, Mr. Chandima de Silva assisted in the same calibre which greatly assisted during the latter part of the research. Further, I convey my sincere appreciation to all my friends, colleagues and teachers at the UCSC who encouraged and advised me on numerous occasions. Especially, the critical and independent comments given by Chamath Kappetiyagama on the final drat of the dissertation assisted in improving the standards to a greater extent. Since this is the first computer science PhD to be fully completed within University of Colombo and Sri Lanka, they continuously encouraged the completion of this research.

I would sincerely appreciate all friends that encouraged and helped in numerous occasions for this effort. Chula, Dharini, Kalika, Prasantha, Tehani and Praneeth always motivated the effort. Chula has been my friend for over 30 years and has been there for me in all occasions of my life. Dharini as my wife's school friend came to my rescue in proof reading the whole dissertation in the midst of her busy schedule and examinations. Kalika who I consider as a master in the art of word processing has been

involved in the dissertation structure from the very beginning. I convey my sincere gratitude to all of these friends and wish them a prosperous future ahead.

I have kept my deepest appreciation to be explained in conclusion. This research was started directly after my marriage to Chiranthi. It is indeed selfish to allocate recent years of my life, partly on the industry profession and the rest on this research. It would be correct to state that this research would have ended with no progress without the support from my wife. Hence, with deepest love and appreciation I dedicate this research to my wife and the two sons – Yenula and Savinu.

List of Publications

1. "Learning Coordinated Actions by Recognising State Patterns with Hubs", Eighth International Conference on Human and Computers, September 2005, University of Aizu, Japan, R.A.C Ranasinghe, A.P. Madurapperuma.

2. "AAANTS-A Distributed Agent Framework for Adaptive and Collective Intelligence", Mobile Agents for Telecommunication Applications (MATA'2003), IEEE/IFIP, Marrakech, Morocco, R.A.C Ranasinghe, A.P Madurapperuma, 2003.

3. "Enhanced Frame Based Representation for an Intelligent Environment", (KIMAS'2003), International Conference on Integration of Knowledge Intensive Multi-Agent Systems, IEEE Boston Section, Cambridge MA, USA, R.A.C Ranasinghe, A.P Madurapperuma, 2003,

4. "Non-Intentional Cooperative Behaviour for an Agent Based Intelligent Environment", IITC Colombo, Sri Lanka, R.A.C. Ranasinghe, A.P. Madurapperuma, 2003.

5. "AAANTS – Distributed Mobile Component Architecture for an Ant Colony Based Synthetic Ecosystem", MATA-2002, Spain Barcelona, Short Paper, R.A.C. Ranasinghe, A.P. Madurapperuma, 2002.

6. "AAANTS – An Intelligent Synthetic Ecosystem", IITC Colombo, Sri Lanka, R.A.C. Ranasinghe, 2002.

Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
AA	Atomic Action
AAANTS	Adaptive Autonomous Agent Colony Interactions with Network Transparent Services
ABS	Action Breakdown Structure
ACL	Agent Communication Language
AI	Artificial Intelligence
API	Application Program Interface
AT	Action Template
BA	Behavioural Act
BC	Behavioural Concentres
CAP	Credit Assignment Problem
CDT	Colony Definition Tool
CI	Cohesiveness Index
EQ	Exploration Quotient
FCM	Fuzzy C-Means
FS	Food Source
HCF	Hub Cohesiveness Frame
IF	Inclusive Fitness
MAS	Multi-Agent System
MC	Monte-Carlo Learning
MDP	Markov Decision Process
MQS	Message Queue Service
POMDP	Partially Observable Markov Decision Process
RC	Relationship Coefficient
RI	Ration of Improvement
RL	Reinforcement Learning

RSG	Receptor Sensory Grade
SECC	System Execution and Control Components
SMVC	System Monitoring and Visualisation Components
TSF	Temporal Sensory Frame
XML	Extensible Mark-up Language

Chapter 1

Introduction

1.1 Introduction

The survival of an entity in the environment is directly attributed to selecting the most appropriate and refined behaviour with respect to the rapid changes in the environment. Behaviour of this nature could be called as congruent with reference to the current demands of the environment. However, over a period of time due to the changes and demands of the environment, the existing behaviour could become obsolete. Hence, behaviour should adapt and improve, or simply be congruent to the latest changes in the environment.

The adaptive entities in the natural world use emergent models [HAZY04] [SUMP00] [MATA94a] [PAOL97] [PARU97b] to achieve behavioural congruence. These models begin with an innate layer of basic incongruent atomic behaviour, which based on the reinforcements and or supervisions from the environment reaches a level of refinement more aligned with the demands of the environment. Hence, dynamically and stochastically combining atomic behaviour that are either accepted or rejected based on the reinforcements from the environment tends to provide a high level of behavioural congruence in natural systems.

Most of the current artificial models on reaching behavioural congruence focus mainly on top-down disintegration [SUMP00] [MATA94a] [PAOL97] of constituent actions coordinated centrally according to a prescribed algorithm or reinforcements from the environment. Even though many scientists have contributed immensely during the past few decades to the progress of behavioural congruence on artificial substance, this domain still remains void of a major breakthrough in comparison with natural models of emergent behaviour. We are yet to experience intelligent behaviour from a non-biological substance which is at least comparable to most basic levels of behaviour in a primitive biological organism.

The success of the naturally occurring models in delivering abundance of heterogeneous and congruent behaviour using concepts of emergence, innateness and adaptations has inspired the thesis of this research. The primary hypothesis of the research is that the constituent atomic actions of a complex behaviour could be successfully coordinated by collaborative and autonomous agents that are loosely coupled through implicit communication to demonstrate emergent congruent behaviour in dynamic environments. The resulting congruent behaviour could be further optimised by using a hybrid learning approach that models adaptive behaviour on a static foundation of innate elementary behaviour.

The research proposed a conceptualised artificial intelligence model that encompasses aspects related to learning, coordination and knowledge representation. This model was called as AAANTS (Adaptive Autonomous Agent colony interactions with Network Transparent Services). The name of the model was coined based on the initial experiments related to heterogeneous distributed services and the inspiration gained from the insect world. Software agent paradigm was selected as the reference concept due to its inherent capabilities to solve a problem using myriad interactions of simple, automated and co-ordinating entities. The AAANTS model focussed on a special co-ordinating approach that refines the innate capabilities through implicit communication among a community of participating agents.

1.2 Motivation

The age old ambition of creating intelligence on artificial substance that is anthropomorphic in nature is still considered a dream yet to be realised by humans. It was this curiosity that initiated the investigation into the behavioural complexity found in nature, which subsequently became the foundation of this research.

There are several theories, models and paradigms that have given inspiration and direction to the work carried out in this dissertation. Naturally occurring collective systems of individually simple animals such as populations of insects and turtles together with artificial phenomena such as traffic jams suggest that individual complexity is not a necessity for complex intelligent behaviour of colonies of such entities [PARU97a] [RESN94].

The community life-style of ants was an inspiration to this research. It was estimated that the Ants' success story spans over several millions of years preceding the known era of human existence [HOLL94]. Each individual ant possesses the capability to only solve part of the overall puzzle while aggressively communicating in primitive methods with the spatially related neighbours to produce emergent behaviour. Ant colonies have evolved means of performing collective tasks, which are far beyond the capacities of their individual structures. This phenomenon is demonstrated without being hard-wired together in any specific architectural pattern and without central control [PARU97a], hence void of any kind of top-down control. The consensus is that comprehension of emergent complexity in insect colonies such as ants would serve as a good foundation for the study of emergent, collective behaviour in more advanced social organisms, as well as leading to new practical methods in distributed computation [BABA01] [GARC01]. Therefore, the key motivation was to device an artificial learning model that could demonstrate collective intelligence analogous to insects.

The "Society of Mind" theory by Marvin Minsky [MINS86], was another inspiration to this research. This theory portrays the mind as a collection of mindless components that interact and compete to provide intelligent emergent behaviour. Society of agents in the mind is triggered by external sensations where agents act individually but in a cooperative and synchronised manner.

The incarnation of a complete multi-cellular being starting from a single fertilised egg seems like a heavenly secret to all of us and certainly a motivation to this research. It is the initial set of genes in a fertilised egg that helps a simple cellular growth to be morphed into a complex combination of organs found in a complete animal. It is amazing that every cell contains a complete footprint of all genes found in the initial cell and each cell only represents a single instance of the overall pattern. This aspect of different cells expressing same genes at different levels could be called as a sub-pattern where most patterns are in fact combinations of a small number of basic patterns [SALA00]. Hence, a gene could be compared to a conductor leading an orchestra; the conductor makes no music on its own but with the proper participants could produce a symphony of enormous beauty and complexity [ELMA99].

1.3 Research Objectives

Congruent behaviour could be achieved through several methods. However, persistence of congruent behaviour in relation to the dynamics of the environment, and further the sustenance of congruence over a considerable period of time is still considered non-trivial based on the current artificial models and architectures [JENN95] [PATT91] [KAEL91] [FERB99] [WEIS00]. This research tends to take a step in the direction of sustaining behavioural congruence using coordination methods from nature based on emergence. The following objectives should be realised to evaluate the accuracy of the hypothesis.

The primary objective of the research is to evaluate whether the bottom-up emergent methodologies [HAZY04] [SUMP00] [MATA94a] [PAOL97] [PARU97b] could provide similar or improved results in comparison to the methodologies that prescribe behaviour composition in a top-down manner to achieve behavioural congruence in dynamic environments.

An objective of this research is to device a learning model that mixes innateness capabilities and reinforcement learning to enhance the overall capabilities of an emergent system. The basic forms of biological life such as insects primarily depend on the innate capabilities that are genetically imprinted in the genome which are later represented by the phenotype. However, more advanced forms of life such as mammals uses the innate layer of capabilities as the basis to build more advanced and environmentally suitable behaviour that are further refined through reinforcements.

An objective of this research is to evaluate whether multiple congruent behavioural instances could result from a static layer of innateness based on the differentiated reinforcements from the environment.

An orthogonal objective is to evaluate whether the accepted learning methods such as the Monte-Carlo algorithm could be further enhanced by using multiple agents that implicitly coordinate through a shared context of information. The outcome should out perform individual additive contribution of multiple agents, hence resulting in some level of emergence.

1.4 Research Contributions

The AAANTS model for Emergent Behaviour

The most important contribution of this research is a model that could simulate the behavioural congruence of elementary actions of a community of software agents. The model encompasses the aspects of the Artificial Intelligence Mix (AI Mix). The congruence in behaviour results from the emergent nature of the model mixed with reinforced adaptations and frame-based representations.

The rationale for emergent behaviour is that the action breakdown of the overall behaviour is not predetermined and could result from simple interactions of elementary actions. In top-down approaches, the designer of the behaviour is responsible for creating the elementary functional breakdown and inter-dependencies.

Mixing Innate and Adaptive Learning Models using Action Templates

A popular method of demonstrating emergence of intelligent behaviour [MATA94a] [PAOL97] is the use of pure reinforcements [SUTT98a] to guide the overall behaviour towards the expected optimum level of congruence. However, another orthogonal approach based on the inspiration from nature is to complement the reinforcement process with an innate layer of capabilities.

The uniqueness of the AAANTS model is the use of a hybrid model (use both innate and adaptive techniques) for implementing adaptability. The model enhances the inherited innate foundation of capabilities using the reinforcements retained from the environment. The key concept that represents innate behaviour is described as Action Templates (ATs). The objective of the adaptive layer is to optimise the coordination of elementary actions within an AT to be congruent with the demands of the environment. It would be evaluated that the introduction of an innate layer would considerably improve the process of simulating congruent behaviour.

The Influence of Hubs

The concept of hubs discussed in complex scale free networks [BARA03], was fused with the AAANTS approach in identifying special environmental states called as hubs. It was found out that the use of these special states improved the capabilities in reaching optimum behaviour.

Distributed Agent Platform for Implementing Emergent Behaviour

The author evaluated a wide range of popular software agent frameworks/platforms to implement the AAANTS model (discussed in section 2.2.5, Table 2.1). However, it was realised that the hybrid nature of the AAANTS model requires considerable customisation to the existing platforms; hence consequently, the decision was taken to develop a generic agent framework to facilitate the objectives of the research. The generic agent platform of AAANTS is also an orthogonal contribution of this research.

The realisation of the AAANTS model resulted in a fully fledged generic agent platform that could be configured to different experimental situations. Three experiments were modelled on this platform, out of which only two experiments were taken for analysis. The AAANTS agent platform could be called as a hybrid platform when compared to other popular platforms based on cognitive and reactive models.

Identifying Situations through Temporal Sensory Frames

Inputs from sensations over a period of time could be separated into discrete temporal collections of data points. Each temporal instance of sensory data values would be introduced as Sensory Frames. A unique pattern for each temporal snapshot of a sensation could be obtained using the limited matrix of hubs and their relationships. Further, a collection of Sensory Frames that define a situation could be identified using a matrix that represents hub values across the temporal frames. This concept could be enhanced further to identify unique patterns in multiple sensory modalities.

Biologically Inspired Heuristics for Congruent Behaviour

As mentioned in the motivation section, the prime fascination and inspiration to this research is the altruistic nature of ant behaviour in the natural world. Research in myrmecology has presented concepts on a wide variety of heuristics used by ants in coordinating behaviour. Some of these heuristics were selected as applicable to the thesis of this research and incorporated to the AAANTS coordination model. The experimental results of this research have disclosed the usefulness of these heuristics to build artificial models of intelligence.

Chapter Contributions

Area of Contribution	Chapters				
	2	3	4	5	6
Insect Behaviour	X				
Multi-Agent Systems/ Software Agent Platforms	X			X	
Emergence	X	X			
Reinforced Learning / Machine Learning	X		X		
Innate Behaviour	X	X	X		
Knowledge Representation / Frames	X	X	X		
Heuristics for Emergence		X			
Reproducing Behaviour			X		
Hubs		X	X		
AAANTS Model		X	X	X	
Fuzzy Clustering Algorithms		X	X		X
Action Templates		X	X	X	X
AAANTS Platform				X	X
Grid World Experiment				X	X
Robotic Arm Experiment				X	X

Table 1.1: Dissertation contribution summary

1.5 Dissertation Overview

A summary of the rest of the chapters are described below.

Chapter 2 – The Foundation: Ants, Agents and Intelligence

The main objective of this chapter is to provide a comprehensive background to the concepts and technologies used in building the AAANTS model. This chapter starts with an introduction to software agents and further continues to discuss the architectures, communication, composition, mobility and ontological commitments related to agent technology. Further, a background to concepts related to artificial intelligence such as reinforcement learning and frame-based knowledge representation is discussed. The chapter concludes with a discussion on the inspiration of natural systems to the research.

Chapter 3 - Behavioural Congruence through Implicit Communication

This chapter contains the essence of the methodologies that helped to formulate the AAANTS coordination model. Emergence based on bottom-up strategies was discussed in detail which resulted in formulating six heuristics which were later used to build the proposed model. Further, the coordination methodology based on action templates and behavioural concentres that dynamically change based on reinforcements is discussed in detail.

Chapter 4 – Reinforced Group Adaptation

This chapter focuses on the methodologies of learning adapted by the AAANTS model. The adaptations are discussed in two broad strategies. Initial discussion relates to the use of reinforcement learning as perturbations to the internal knowledge representations of agents. This is the main mechanism that facilitates the participants of an agent colony to adjust each other for behavioural congruence. The latter part of this chapter focuses on the continuation of existing knowledge to future generations of the colony

Chapter 5 – The AAANTS Platform

This chapter concentrates on the implementation aspects of the AAANTS agent platform. A new agent platform was developed to realise the objectives of the research since the existing platforms required considerable adaptations and enhancements. The AAANTS platform was further moulded to deliver the results of the two experiments to justify the generic nature of the platform. A separate external simulator was developed for the Grid-World experiment and the Lego Mindstorms Robotic kit was used to implement the robotic arm experiment.

Chapter 6 – Simulations and Experiments

This chapter provides an in-depth description of the experiments conducted to evaluate the hypothesis of this research. The experiments span across two experimental domains, the Grid-World Foraging and Robotic Arm Movement.

Chapter 7 – Evaluations and Conclusions

The objectives of the research are discussed with respect to the results of the experiments. A further detailed discussion of improving and extending the AAANTS model, platform and experiments is discussed in the future work section.

Chapter 2

The Foundation: Ants, Agents and Intelligence

2.1 Introduction

During the background study of this research it was identified that basic ingredients of intelligence are related to the aspects of coordination, adaptation and representation. These aspects would be referred to as the "AI Mix" for easy reference in the rest of the discussion. The future success of artificial intelligence projects and research would depend on finding the appropriate AI Mix to create different grades of intelligence. The rest of this chapter is a discussion that journeys through the ingredients of the AI Mix with special emphasis on collective intelligence, agent technology, reinforcement learning and frame-based knowledge representation techniques which would act as the background for this research.

2.2 The Concept of Software Agents

One of the most difficult hurdles to overcome, though surprisingly, is the definition of the term "Software Agent" in a standard manner using one or more phrases. This term is very casually used in the industry to even describe systems that does not possess accepted agent characteristics. According to Michael Luck [LUCK99], this vagueness in the definition has attributed to both progress and confusion within the community. However, a clear definition to the term "Software Agent" should be given before proceeding to rest of the discussion. Appendix A gives a near exhaustive list of software agent definitions by the authoritative authors within the agent research community. These definitions could be used to derive the common characteristics expected by the software agent designers. With the assistance of these listed definitions in Appendix A, software agents could be broadly defined as software artefacts that are intelligent, autonomous, adaptive, goal-driven, cooperative and reactive to the environment in order to assist and act on behalf of humans.

2.2.1 Software Agent Characteristics

There are number of characteristics that should be prevalent in a software entity for it to be recognised as a software agent. These characteristics could be segmented as compulsory and orthogonal, though there still prevails a controversy among the authorities in the field of such a demarcation. The continuation elucidates each of the important agent characteristics found in the software agents such as autonomy, adaptability, cooperation, rationality and mobility.

Autonomy is an important and compulsory characteristic expected out of a software agent. Autonomy is defined as an independent and purposeful existence of states that does not directly rely on any external entity [INVE96] [FRAN96]. This independence could be explained as the ability of an entity to act without direct intervention of humans or other agents maintaining control over its own actions and internal state [JENN98] [JENN95a] [BRAD97] [FERB99]. This requires aspects of periodic actions, spontaneous execution, and initiation; in that the agent must be able to take pre-emptive actions that would eventually benefit the beneficiary [LEON97] [PETR96]. Further, autonomous systems decide for itself on how to relate its sensor data to motor commands in such a way that its goals are attended successfully [PATT94]. Therefore, autonomy simply means the ability of an entity to determine goals on its own.

In the context of a Multi-Agent System[1], there could be agents with different degrees of autonomy. Agents could be categorised as goal generating and goal adapting agents [LUCK96]. Goal generating agents are autonomous, since they do not depend on the goals of others, but instead possess goals, which are generated from internal motivations. Goal adapting agents achieves harmony by aligning goals with other agents that have common objectives. Therefore, an autonomous agent possess motivations that could be evaluated with reference to the environment, hence its behaviour is determined by both external and internal factors.

The above facts, though offering explanation regarding autonomy, introduced some controversy with reference to different agent paradigms such as the cognitive and the reactive. Most of the facts are more appropriate for cognitive agents[2], however; goal generation and achievements are performed in a collective manner in reactive models[3].

If a system is able to improve over a period of time and become better at achieving its goals with experience [BRAD97] [FRAN96], then it demonstrates Adaptive or Learning capabilities which is

[1] Multi-Agent Systems in a basic sense could be described as a collection of software agents that interact for cooperative behaviour.

[2] Cognitive agents are characterised by an internal symbolic reasoning model used to engage in planning and negotiation with other agents.

[3] The reactive agents do not have any internal symbolic models of their environment, and they act using a stimulus-response type of behaviour.

considered as a compulsory characteristic of a software agent system. Therefore, if an autonomous agent is able to adjust its goals in terms of what it perceives in the changing environment, it could be called as adaptive.

Three basic types of adaptations could occur in software agents, namely, learned, evolved and pre-programmed [GARC01]. In learned adaptation, agent behaviour is improved over a period of time using a learning process, and in evolved adaptation the improvement is complemented by an inherited set of behaviour that undergoes natural selection [GARC01]. Pre-programmed adaptation is the easiest to achieve, as they are usually in the form of hard-coded logic by the creator [GARC01]. The work introduced in this dissertation focuses on learning techniques related to learned and evolved adaptation.

Cooperation through discourse is another important characteristic of an agent system. The agents could be designed for discourse to achieve objectives similar to that of humans. The prerequisites for a discourse are participants that need to communicate, a media, a language for expression and a defined ontology understood by the participants [LEON97]. Practically, an ontology is an agreement to use a vocabulary in a way that is consistent with respect to the theory specified by it [MIZO95] [GRUB93]. During a discourse, the participants generally require two-way feedback, in which both parties make their intentions and abilities known and mutually agree on something resembling a contract about what is to be done, and by whom [JENN98]. In software agent systems, the media for discourse is usually provided by information transfer among network and memory based transport channels.

An Agent Communication Language (ACL) represents language of expression which is a facilitator for communication with respect to the software agent terminology [JENN95a] [BRAD97] [FRAN96] [FERB99]. With reference to ACLs there are few accepted protocols within the cognitive agent paradigm such as KQML (Knowledge Query Mark-up Language) and FIPA ACL [LACE00a] [LACE00b] [DIMI98].

The above discussed characteristics are considered essential for an agent-based system by the academia. However, there are other orthogonal characteristics that could be accepted as complementary to software agent systems such as mobility [JENN95a] [BRAD97] [FRAN96], reactivity [JENN95a] [MICH99] [FRAN96] [WOOL00], rationality [JENN95a], personalisation [MICH99] [FRAN96], veracity [JENN95a], benevolence [JENN95a] and goal-directedness [JENN95a] [JENN98] [FRAN96]. Out of these characteristics, rationality and mobility are considered important [JENN95a] [BRAD97] [FRAN96], hence discussed below.

The characteristic of Rationality relates to making the right decisions and producing successful behaviour where social rationality should enable agents to select behaviour that would maximise the expected utility in a social context as opposed to being individualistic [HOGG97]. A rational agent could be viewed as a system continuously receiving perceptual input from the environment in which it is embedded and responding by taking actions that affect that environment [KINN92]. A rational agent is assumed irrational if world models and the models of other agents are non-existent [MATA94b]. According to the Principle of Social Rationality [HOGG97], if a member of a responsible society could perform an action whose joint benefit is greater than its loss, then it may select that action.

Mobility enables an agent-based software artefact to be transferred to another host location with the execution state preserved during the movement [MFAC97]. This capability permits agents that must conduct a high-bandwidth conversation to move to a common processor, as a result that the network as a whole is not burdened with the traffic between them. Movement to a common processor using mobility also permits local communities of agents to interact with each other even when the processor on which they are located is disconnected from the rest of the network [PARU98]. However, mobility in heterogeneous platforms may become complicated due to authorisation and authentication mechanisms.

2.2.2 Software Agent Classifications

There is a considerable amount of applied and pure research done in the field of software agent technology by the industry and academia. Consequently, there are many varieties of software artefacts labelled as "Software Agents". To overcome possible confusions some researchers have prescribed a nomenclature that spread across several dimensions. Some of these dimensions are discussed in continuation.

Software agents could be classified using a subset of the characteristics they possess. As per the discussion in the previous section it is clear that a basic agent should at least possess properties such as autonomy, goal-orientedness, adaptability and cooperation. The mobile agents, learning agents and mobile-learning agents could be described as some of the popular categories of software agent systems [FRAN96]. Another classification is derived based on the discussed characteristics: collaborative agents, interface agents and smart agents [NWAN96a]. Apart from these characteristics, software agents could be classified based on the function such as information gathering, filtering or depending on the control mechanism such as algorithmic, rule-based, planner, fuzzy, neural nets and machine learning [FRAN96].

Prior to describing the next type of agent classification, several definitions of knowledge representation based on symbolic and numeric methods needs to be clarified. Symbolic and numeric paradigms offer two different description languages for defining intelligent systems. In order for any system to serve the role of a representation, it must include an encoding process that maps the physical state of the external environment into an internal state and a decoding process that maps an internal state into a physical state of the external environment [HONA94].

A definition for deliberative agents could be derived from the cognitive thinking paradigm, and is characterised by an internal symbolic reasoning model used to engage in planning and negotiations [NWAN96a] [FERB99]. When the internal specification of an agent is defined by a language, such agents are commonly referred to as cognitive, rational, deliberative, or heavyweight [WEIS00]. According to Jacques Ferber [FERB99], the cognitive agents could be called "intentional", due to the presence of goals and explicit plans that allow them to achieve long term objectives. Further, cognitive systems need to cooperate with each other to achieve community wide goals that are usually too complex to accomplish individually.

The reactive agent concept, which could also be called as an emergent organisation, claims that it is not necessary for agents to be individually intelligent, for a system to demonstrate intelligent behaviour [COLO93]. The reactive agents do not have an internal symbolic model of their environment, and they act based on stimulus-response type of behaviour by responding to the present state of the environment in which they are embedded [NWAN96a] [FERB99] [COLO93]. Therefore, due to the absence of an internal model of the environment and possible actions within the state space, reactive agents cannot be expected to individually implement goal-directed behaviour in respect of cognitive models.

The combinations of the above discussed list of dimensions may result in a myriad of agent typology. However, for the sake of clarity and understanding, this space could be reduced to a list containing agent types such as collaborative, interface, mobile, information, reactive, hybrid and smart agents. Out of these types, the logic based approaches are elegant, and have clean semantics [WEIS00] [NWAN96a]. The disadvantages of logic based approaches being the inherent computational complexity of theorem proving makes it questionable whether agents as theorem proving could operate effectively on time constrained environments [WEIS00].

2.2.3 Agent Theory and Architecture

The agent-based research activities have become common and widespread within the Artificial Intelligence research community. The research projects spread across a wide spectrum of agent capabilities and many specialises on different combinations of agent-based features. According to N. Jennings and M. Wooldridge [JENN95a] [WOOL94], the software agent research could be broadly segmented into areas based on Theory and Architectures.

Agent theory is regarded as a specification of an agent where agent theorists develop formalisms for representing the properties of agents, and using these formalisms, try to develop theories that capture desirable properties of agents [JENN95a]. Further, it is expected that a realistic agent theory would be represented in a logical framework that combines these various components and a complete agent theory, expressed in logic with properties, must define how the attributes of agency are related [JENN95a]. According to Mark d' Inverno et al [INVE97], formal agent theories are agent specifications, not only in the sense of providing descriptions and constraints on agent behaviour, but also in the sense that one understands the term "specification" from main stream software engineering, namely that they provide a foundation from which to design, implement and verify agent systems.

The agent architectures embark on the practical aspects of agent technology. An agent architecture simply specifies how an agent could be decomposed into the construction of a set of modules and how these modules should be made to interact [PATT91] [KAEL91]. Also, for architecture to be useful, the total set of modules and their interactions should provide an answer to the question of how the sensor data and the current internal state of the agent determine the actions and future internal state of the agent. This is further described in a more implementation related manner by Gerhard Weiss [WEIS00], as a mapping of the internal capabilities of an agent, its data structures, the operations that may be performed on these data structures, and the control flow of these data structures.

Agent architecture is broadly segmented into deliberative, reactive hybrid and layered architectures. Deliberative architectures contain an explicitly represented symbol model of the world where decisions are made via logical reasoning, based on pattern matching and symbolic manipulation [JENN95a]. According to N. Jennings et al [JENN95a], a drawback of this approach is that it is difficult to build useful symbol manipulation algorithms that would guarantee to terminate with useful results within an acceptable fixed time period and further, though this approach is very attractive in theory, it currently seems to be unworkable in practice.

On the other hand, a reactive architecture does not maintain a symbolic model of the world either in a complete or incomplete manner and therefore lacks symbolic reasoning found in cognitive agent architectures. The Agent Network Architecture [PATT91] is an example of a reactive system. Here the agents are defined as a set of competence modules and each module is specified by the designer in terms of pre and post conditions with an actuation level, which gives a real-valued indication of the relevance of the module to a particular situation [PATT91].

Hybrid architecture uses a blend of both deliberative and reactive approaches. An obvious approach is to build an agent out of two or more subsystems: a deliberative aspect containing a symbolic world model, which develops plans and makes decisions in the way proposed by the mainstream symbolic AI; a reactive aspect, which is capable of reacting to events that occur in the environment without engaging in complex reasoning [JENN95a]. Often the reactive component is given same kind of precedence over the deliberative, therefore that it could provide a rapid response to important environmental events. The concept of flexible agents could be regarded as of hybrid nature; reactive when responding to changes in the environment and deliberative when planning ahead of time [RUI02].

Another important characteristic of many architectural models is the nature of being layered. Layered architecture is currently the most popular general class of agent architecture available because being layered facilitate natural decomposition of functionality. The advantages of layered approaches are evident when analysing many systems found in nature from animal anatomy to eco-systems. According to G. Weiss [WEIS00], there could be two types of control flows within layered architecture, namely, horizontal and vertical. In horizontal layered architecture, the software layers are each directly connected to the sensations and actuations and in vertical layered architecture sensations and actuations are processed by a specialised layer. Subsumption architecture [VIDA02] is a good example for layered approach for connecting perceptions and actions. This is accomplished by building a series of incremental layers, each layer connecting perceptions to actions.

According to G. Weiss [WEIS00], the main problem with layered architectures is that while they are arguably a pragmatic solution, they lack the conceptual and semantic clarity of non-layered approaches. In particular, while logic based approaches have clear logical semantics, there is no clear consensus when aligning such semantics to layered architecture [WEIS00]. The other issue is related to the interaction between layers [WEIS00]. If each layer is an encapsulation of a defined independent function, then it is necessary to consider all possible combinations that the layers could interact with one another. However, layered architecture had been very successful in many research and industrial applications based on agents, mainly due to inherent advantage of been able to use a divide-and-conquer strategy.

2.2.4 Multi-Agent Systems (MAS)

Multi-agent systems, in a basic sense could be described as a collection of software agents that interact for cooperative behaviour. They could also be defined as a tuple of three elements: a set of agents, an environment, and a coupling between agents and the environment [PARU96]. According to Peter Stone [STON98], a multi-agent system is a subfield of artificial intelligence that aims to provide both principles for construction of complex systems using multiple agents and coordination mechanisms for agent behaviour.

The multi-agent approach lies at the crossroads of several disciplines such as Distributed Artificial Intelligence and Artificial Life [FERB99], hence represents some vagueness in definition. Therefore, to overcome the obscurity, common characteristics should be identified. There are several common characteristics found in a typical multi-agent system [WEIS00] [BABA01] [FERB99] [AYLE98] such as the availability of a communication infrastructure, the lack of central design, distributed-independent processes and the applicability to inherently distributed problems.

Generally, multi-agent systems offer a way to relax the constraints of centralized, planned, sequential control, though not every multi-agent system takes full advantage of this potential. They offer production systems that are decentralized rather than centralized, emergent rather than planned and concurrent rather than sequential [PARU94]. The multi-agent systems are more impressive compared to single agent systems due to space distribution, parallelism, time distribution, divide and conquer, cost, reliability and robustness [RUI02].

A multi-agent system is identified as a concept and is clearly separated from mechanisms and technology. Hence, multi-agent systems could use object-oriented expert systems and distributed computing technologies to implement applications and toolkits [AYLE98]. Multi-agent system is a highly diverse field and there are many domains of applicability. However, the applicability could be broadly classified into five categories: distributed problem solving, collective robotics, simulations, construction of hypothetical worlds and design of programs [FERB99]. The concept of MAS is very important to this research due to that fact that the implemented AAANTS platform is designed based on this paradigm.

2.2.5 Software Agent Applications and Platforms

Software agents are not a panacea for industrial software [JENN96], and like any other technology they are best used for problems whose characteristics require their particular capabilities. Software agents are appropriate for applications that are modular, decentralised, changeable, ill-structured, and complex [JENN96]. Hence, problems that naturally exhibit the above characteristics are suitable for software agent implementations.

Application domains in which software agent solutions have been applied or researched includes workflow management, network management, air-traffic control, business process re-engineering, data mining, information management, electronic commerce, education, personal digital assistants, e-mail filtering, digital libraries, command and control, smart databases and schedule management [NWAN96a].

According to Tihamer Toth-Fejel [TOTH00], MRP (Manufacturing Resource Planning) and ERP (Enterprise Resource Planning) software products are too limited and complex and require a multi-million dollar and multi-year implementation effort, with no guarantee of full success. In addition, scaling these complex software systems is unthinkable. It is also understood that small-grained software agent based systems promise an alternative way of handling the complexity and could be specially used to solve scheduling related issues [TOTH00].

Another application area for software agents is in Robotics. The marriage of robotics with software agents could be used for wide variety of applications such as extraterrestrial exploration [GARC01], manufacturing plants [BADI04] [PARU96b], supply chain management [MEHR97], submarine exploration [GARC01], bomb deactivation and rescue [KOES06] and entertainment [NICO06].

A vast range of multi-agent platforms are prevalent in the industry and academia. They have helped to explain the complexity, advantages and disadvantages of different concepts, models and theories. Some of the popular agent platforms evaluated during this research as generic agent implementations were Aglets [VENN97], FIPA-OS [NORT01] [PEDD02], ADE [MEHR97] [ANDR03], AARIA [BAKE99], MadKit [RICO00], AgentBuilder

[RICO00], JACK [RICO00], dMARS [INVE97b], JADE [BELL99], Grasshopper [BAUM99] and Zeus [RICO00]. Each platform implements a mixture of agent characteristics described earlier and some are limited to specialised application domains. During the early stages of this research the author experimented and prototyped with platforms, such as Aglets, JADE, Grasshopper and Zeus to evaluate the alignment to the research objectives. A more elaborative discussion on related agent platforms is found in Appendix B.

Characteristics	Aglets (ASDK)	FIPA-OS	Grasshopper	JADE	Zeus
Standard Compatibilities	MASIF, CORBA, Java	FIPA, Java, CORBA	Java, MASIF, FIPA, CORBA	Java, FIPA, CORBA	FIPA, Java 2
Communication support	Sockets, Message passing agents	ACL, IIOP, RMI, XMK.	ACL, Sockets, RMI, IIOP	ACL, RMI, IIOP, WAP, XML	KQML, ACL
Mobility	Weak mobility through Java serialisation	Not in-built, may be modified.	Weak mobility	Weak Mobility	No
Security Policy	Roles, Context/Server security, Proxy, Java security.	RMI over SSL.	External – X.509, SSL; Internal – Java security	Connection Auth, user Validation, RPC encryption, Object Manager	ASCII encoded, Safe Tcl scripts, MIME compatible, PKI standards.
Availability	Free source	Free source	Free source	Free source	Free source
Usability & Documentations	Clear, simple, good GUI and good documentation	Average GUI and documentation	Good GUI, very well documented	Very good GUI, Docs and acceptance.	Very weak docs,
Development Facilities	ASDK development with Java	Java	Java 2, Windows CE, Web plug-in	Java, LGPL open source	BT product
Implementations	Electronic Air Tickets	Applications related PDA etc.	IT management, Mobile agents, process integration.	eBusiness, Wireless applications,	e-commerce, work-flow applications

Table 2.1: Comparison of agent platform characteristics

The analysis and comparison of the five important agent platforms evaluated during the initial stages of the research are depicted in Table 2.1. Most of the listed comparative dimensions were extracted from industry analysis [NGUY02] [LESZ04] [BURB04]. Some of the platforms are tightly coupled to an agent model such as in the case of AgentBuilder and JACK platforms being based on BDI agent model [RICO00]. Most of the agent platforms are compatible to acceptable agent standards such as FIPA and MASIF.

2.3 Machine Learning Techniques

The following is a discussion of the techniques adapted by most of the current software agent implementations for the purpose of learning. The selection of a learning technique is of prime importance to an agent system and is usually decided based on dimensions such as the application domain, expected temporal feed-back latency (on-line and off-line), knowledge structures utilised and limitations on resources. The distributed machine learning techniques are given special emphasis during the discussion because of their applicability to the AAANTS research.

2.3.1 Background on Machine Learning

According to A. Drogoul [DROG98], distributed machine learning could be broadly segmented into individual agent learning, group of agents learning, and organisational learning. Individual agents in a multi-agent system could employ different learning techniques such as functional learning, direct learning and model based learning (myopic-learning and strategic-learning) where model-based learning requires more complex knowledge representations than those used in reinforcement learning [DROG98]. During organisational learning, system internal structures are evaluated on the usefulness to the whole group [DROG98] where class of adaptive algorithms called genetic programming [WHIT00] could be used to evolve these structures.

The nomenclatures used by Van Parunak et al [PARU97a], for machine learning techniques include categories such as ontogenetic, phylogenetic and sociogenetic techniques. Classical artificial intelligence learning could be called as ontogenetic which takes place within a single agent during the course of its existence. Phylogenetic mechanisms such as genetic programming could improve the behaviour of a species of agents over successive generations. Sociogenetic mechanisms that construct markers in the environment could enable an agent community as a whole to learn even if individual agents are not modified. Each of these mechanisms demands different requirements on the behaviour of the agents. According to Van Parunak et al [PARU97a], phylogenetic learning is not nearly as demanding as the ontogenetic mechanisms developed in classical artificial intelligence, and sociogenetic mechanisms could be even simpler.

Layered learning is a machine-learning paradigm defined as a set of principles for the construction of a hierarchical, learned solution of a complex task which allows for a bottom-up definition of subtasks at different hierarchical levels [STON00]. Similar to other layered approaches, learned subtasks are organised into layers [STON00] where each layer is complemented by layers above and below. The overall learned output of a layered artificial entity involves the contribution of all concerned adaptive layers. Hence, the learned capabilities are not central but distributed across all layers.

Another interesting theory based on the specialisation and generalisation of information was presented by Marvin Minsky in his much sought after book, The Society of Mind [MINS86]. According to his theory, there are two types of strategies to learn from the environment: Uniframers and Accumulators. The approach of Uniframers is to disregard discrepancies in favour of regularities where they tend to be perfectionists but also tend to think in terms of stereotypes. However, this may sometimes lead to recklessness because they have to reject some evidence in order to produce Uniframes. On the other hand, accumulators are less extreme since they keep collecting evidence, hence are much less prone to mistakes, but with the deficiency of making less discoveries. When aligning these two strategies to a typical human learning experience, it is evident that many of us use a mix of these strategies in different learning situations. However, the ratio of application of Uniframes and Accumulators vary based on the human personality.

Learning could also be performed by analysing differences, recording cases, managing multiple models and by training neural nets [WINS92] [RUSS95]. Learning by analysing differences could be done by using induction heuristics that enable procedures to learn descriptions from positive and negative experiences [WINS92]. Learning by recording cases uses consistency heuristics to identify or recognise a new object with reference to some earlier known situation [WINS92]. Learning by managing multiple models uses positive and negative examples to create a version space, which could be used to find solutions to a specific problem [WINS92]. A version space is a representation that enables to keep track of all the useful information supplied by a sequence of learning examples, without remembering any of the examples [WINS92]. Lastly, neural nets explain how neural like elements, arranged in nets, could be used to recognise instances of patterns [WINS92] [RUSS95].

The learning based on the feedback given from the environment is generally called as reinforced learning. This feedback could fall into a dimension that is limited at two ends by means of purely evaluative feedback and purely instructive feedback [SUTT98a]. Purely evaluative feedback provides a value that indicates the suitability of the action taken and purely instructive feedback indicates the correct action to take [SUTT98a]. Therefore, instructive feedback is also called supervisory since it instructs the correct action to take [SUTT98a] in contrast to giving freedom for selecting an action which the agent evaluates as suitable. In other words, supervised learning could be discussed as a general method for training a parameterised function approximator, such as a neural network where it requires sample input-output pairs for the function to be learnt [HARM96] [RUSS95]. Therefore, supervisory learning techniques do not control the environment but rather behave as instructed by the environment [SUTT98a].

According to L.P. Kaelbling [KAEL96], there are several differences between reinforcement learning and supervised learning. Among them the absence of input-output pairs and requirement for online performance with reference to reinforcement learning are considered as major differences. Another key difference is that, in reinforcement learning the software agent is never told the correct action to take in a situation, but only some measure of relative suitability is indicated [REYN02], consequently, it is up to the software agent to independently select the most suitable behaviour based on the feedback from the environment.

A complex task could be achieved through the execution of elementary actions. The sequential execution of a collection of elementary actions to implement a complex task is of trivial nature. However, in most behavioural contexts, tasks require overlapped concurrent execution of actions by individual control units. This is similar to the operation of the brain where many parts of the brain are activated in parallel and through competition, these concurrent regions of the brain attract those inputs they handle particularly well, and they are recruited for those tasks which require a particular form of computation [ELMA99]. The key issue with concurrency is related to identifying the optimum coordination strategy. Whenever the state and action space is large, a distributed approach to perform the computation is desirable because it makes computational speedups from coarse-grain parallelism possible [SCHN98].

The progress indicators [MATA94b] could be effectively used to measure progress during a behaviour composed of elementary actions. These indicators were evaluated with reference to experiments done on foraging behaviour of ants. While immediate reinforcement is not available in many domains, intermittent reinforcements could be provided by estimating the progress relative to its current goal and weighing the reward accordingly. The progress indicators diminish brittleness of a learning algorithm by decreasing sensitivity to noise, encourage exploration in the behaviour space, and decrease fortuitous rewards [MATA94b].

Social or observational learning is the process of acquiring new behaviour patterns in a social context, by learning from conspecifics [MATA94b]. Social learning could be implemented through imitation and mimicry [MATA94b]. Though both mimicry and imitation observe and repeat the behaviour of another agent, in mimicry the mimicking agent does not understand the goal of the behaviour or the internal state of the agent being mimicked [MATA94b]. The social facilitation is another social learning method which refers to the process of selectively expressing behaviour which is already part of an animal's species-specific repertoire [MATA94b]. A society could develop social rules based on individual learning if the agents are able to estimate other agents' reinforcement and their individual reinforcement is positively correlated with their conspecifics [MATA94b].

2.3.2 Reinforcement Learning

Reinforcement Learning is described as a computational approach that could be used to understand and automate goal-directed learning and decision making through trial-and-error in a dynamic interactive environment [RUSS95] [HARM96] [TUNG01]. This is a computational approach that study learning from interaction with the environment [SUTT98b]. Therefore, the two most important and distinguishing features of reinforcement learning is trial-and-error and delayed reward. It is distinguished from other computational approaches by its emphasis on learning by an individual from direct interaction with its environment, without relying on exemplary supervision or complete model of the environment [SUTT98a]. In summary, a reinforcement learner interacts with its environment by adaptively choosing its actions in order to achieve definite long-term objectives [TUNG01].

Reinforcement learning is not a type of neural network, nor is it an alternative to neural networks, but rather, it is an orthogonal approach that addresses a different, more difficult question [HARM96]. The early connection between neural networks and reinforcement learning may have led to the persistent misconception that the latter is a subfield of the former [RUSS95]. Therefore, there seems to be a dependency of reinforcement learning on neural networks due to early relationships [RUSS95] and was confirmed by Mance Harmon [HARM96] by declaring that reinforcement learning combines the fields of dynamic programming and supervised learning.

The two of the most highly researched methods in reinforcement learning so far by the research community were Monte-Carlo (MC) and Temporal Difference (TD) [SUTT98a]. These methods could be further enhanced and also combined in flavour with the use of techniques such as approximation, eligibility, models, and active/passive learning [SUTT98a].

2.3.3 Elements of Reinforcement Learning

Reinforcement Learning requires a basic framework consisting of two elements: Agent[4] and Environment [SUTT98a]. These two elements interact in a continuous or discrete manner. The environment presents the agent with current state information and the agent should select appropriate actions to transfer from one state to another, subsequently being rewarded by the environment for the appropriateness of the action taken. An agent's prime objective is to maximise rewards from the environment over a period of time. However, the term agent is referred to any entity that could be of monolithic or distributed form, sharing a distinct set of objectives. The agent interaction with the environment could be considered as discrete through time. These discrete interactions could be described as either episodic or continuous [SUTT98a]. Episodic interactions naturally divide the agent actions to segments or episodes where as in continuous interactions, this demarcation is absent.

The environment of a reinforced learning problem could be either a Markov Decision Process (MDP) or a Partially Observable Markov Decision Process (POMDP). According to Richard Sutton et al [SUTT98a], a state signal that succeeds in retaining all relevant information[5] is said to be Markov, or to have the Markov property. The learning methods based on Markov Decision Process require complete observation [KAEL96]. In a Markov situation the optimal policy for the agent is considered

[4] The term agent represents the active adaptive entity that is connected to its environment via perceptions and actions.

[5] The state signal should retain or imply information that has lead to the current situation or state.

deterministic solely on the current state as past knowledge or experience is not required to take suitable action [TUNG01]. If the state and action spaces are finite, then it is called a finite MDP [ARAI00].

The reinforced agent-environment interaction could be described using a series of notations. The identified variables are: t: Discrete time steps, S: Set of possible states, A: Set of available actions, R: Reward value. An agent at a given time t and been in state $s_t \in S$, which takes an action $a_t \in A$, would be given a reward $r_{t+1} \in R$. There are four main sub-elements to a reinforced learning system such as a policy, a reward function, a value function, and optionally a model of the environment [SUTT98a] [HARM96] [KAEL96]. The following is a discussion of these elements' contribution to the overall reinforced learning process.

A policy is used to define the behaviour of a learning agent. In a broader sense, a policy is a mapping from perceived states of the environment, to actions to be taken when in those states, and corresponds to a set of stimulus-response rules or associations [HARM96] [CASS95]. A policy could be implemented by means of a simple lookup table function that maps actions to stimulations or by a complex search process. Further, according to Bob Price [PRIC03], the calculation of the optimal policy could be done by either model-based or model-free approaches.

A primary objective of a learning agent is to find a policy, mapping states to actions, that maximizes the long-run measures of reinforcement [KAEL96]. According to Richard Sutton et al

[SUTT98a], policy is the core of a reinforced learning agent in the sense that, it alone is sufficient to determine behaviour. The difference between a policy and a plan is that a plan specifies a sequence of actions to perform and does not necessarily specify the appropriate action for each possible situation [CASS95].

Further, a policy could be either deterministic or stochastic [SUTT98a] [CASS95]. A deterministic policy is one that specifies a single action to take in each state. A stochastic policy specifies the probabilities of a number of possible actions to execute in each state where the value of a state is defined as the sum of the reinforcements received when starting in that state and following some fixed policy to a terminal state [SUTT98a]. The optimal policy would therefore be the mapping from states to actions that maximizes the sum of the reinforcements when starting in an arbitrary state and performing actions until a terminal state is reached [HARM96]. In the AAANTS model, a stochastic policy is used to select the optimum actions during exploitation mode of activities.

The next element of the reinforcement learning methodology is the reward function and is usually provided by the environment to the learning agent. A reward function defines the goal in a reinforcement learning problem and it maps perceived states of the environment to a single value called a reward [SUTT98a] [TANG02]. Therefore, a reinforcement agent should be designed with an inherent urge to maximise the rewards it receives over a long period of time.

A particular state may provide a list possible actions and the agent may initially need to try actions at random in order to gather enough information on expected rewards. An agent with a limited list of information about actions and rewards may tend to act in a greedy fashion. Also the rewards shouldn't be evaluated in isolation since a single reward value cannot be compared with a reference value. Therefore, rewards could only be gauged in comparison to other reward values generated during trial-and-error activities. A central intuition underlying reinforcement learning is that actions followed by large rewards should be made more likely to recur, whereas actions followed by small rewards should be made less likely to recur [SUTT98a]. A reference level, called the reference reward [SUTT98a] could be used by a learner for evaluating rewards and the reference could be the average of previously received rewards. Learning methods based on this idea are called reinforcement comparison methods [SUTT98a].

The incorporation of models and planning into reinforcement learning systems is a relatively new development [SUTT98a]. A model incorporates the planning capabilities to a reinforcement learning system. This means that any way of deciding on a course of action by considering possible future situations before they are actually experienced [SUTT98a]. According to Richard Sutton et al [SUTT98a], early reinforced learning systems were explicitly trial-and-error learners and it gradually became clear that these methods are closely related to dynamic programming methods, which do use models, and that they in turn are closely related to state-space planning methods. However, building and maintaining a model is an optional feature in a reinforcement learning system.

2.4 The Notion of Intelligence and Knowledge

According to Dimitris Chorafas [DIMI98], intelligence[6] is a reflection of the environment, the situation and the subject that we consider. Further, Carlos Garcia [GARC01], claims that something could be called intelligent if that entity exhibit behaviour that is considered intelligent by an outside observer. Therefore, what is defined as intelligent is relative and may differ depending on the type of behaviour and the view point of the observer. There is also an argument that intelligence is a propensity or ability to adapt [KENN01]. It is identified that the proper understanding of simple behaviour are prerequisites to understanding higher levels of intelligence, as they are the precursors in evolutionary history [PFEI02].

In order to demonstrate intelligence there are several key elements identified such as perception[7], cognition[8], conceptual modelling, logical representation and memory [DIMI98] [MICH99]. Since intelligence is of relative nature, the above key ingredients could be mixed in various combinations to achieve different levels of intelligence. The components that characterise increasingly intelligent behaviour are memory, calculation, learning, inference, speculation, abstract thinking, concretization of thoughts and

[6] "The ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria" [WEBS88]. "Quickness of understanding, sagacious" [OXFO88].

[7] Perception is the awareness of the elements of the environment through physical sensations.

[8] The act or process of knowing including both awareness and judgement is called as cognition.

integration [DIMI98]. Memory, calculation and learning could be described as lower levels of intelligence that is usually demonstrated by current state-of–the-art intelligent agents. When evaluating recent contributions to artificial intelligent artefacts by the research community it is evident that the considerations are more towards the lower levels of intelligent behaviour and new innovative paradigms, theories should be forwarded by the research community to achieve relatively anthropomorphic levels of intelligence.

A very important aspect required to demonstrate intelligence is knowledge. Knowledge could be of real use only when it is properly represented. Therefore, knowledge representation is an important criterion for systems that demonstrate intelligence. According to Randall Davis et al [DAVI93], knowledge representation plays five distinctive roles that; it is fundamentally a surrogate to the real thing, there should be a degree of fidelity between the representation and the real thing, it is a fragmentary theory of intelligent reasoning, it is a medium for pragmatically efficient computation, and it is a medium of human expression. It is understood that knowledge should be used to refer to the representations that support behaviour [ELMA99].

Knowledge could also be expressed as relations among facts or episodes, as well as between these facts and their values [DIMI98]. Defining a relation among facts, establishing causality (the relation between a cause and its effect), mapping the relations through signs and rules, and manipulating them to obtain a result could be considered as signs of knowledge [DIMI98]. Further, the concept of knowledge, as it had been developed in artificial intelligence, could be defined as all the information (learning, know-how, experience, memories, concepts and facts) needed by a human being (or machine), organised in such a way that an individual could carry out a task considered as being complex [FERB99]. Though we discuss knowledge in unison, there are different variations of knowledge such as skills, virtue, world knowledge, crystallised intelligence and fluid intelligence [DIMI98].

It would be valuable to differentiate the confusion between the terms, data and knowledge. According to Riichiro Mizouchi [MIZO95], data stored in databases are less context-sensitive than knowledge; in other words data could be interpreted independently of any context than knowledge. When a user applies knowledge in a knowledge base to a different problem, special attention should be given to the context of the problem solving and check applicability of the knowledge.

2.4.1 Techniques of Knowledge Representation

From the point of view of classical artificial intelligence, knowledge could be represented in the form of a set of symbols expressed in units of knowledge called as "symbolico-cognitivist" paradigm [FERB99]. This model considers that the representations accessible to an individual are expressed in the form of symbols, which directly refer to the entities of the world in which an individual is immersed. These symbols are articulated with the help of an internal language, the syntax of which is generally derived from the logic of first-order predicates [FERB99]. The reasoning consists of manipulating these sets of symbols to form other sets through a process of inference.

The next form of knowledge representation is that of a connectionist point of view, which assumes that knowledge is integrally distributed within a network of cellular automata in the form of numerical values attached to connections [FERB99]. The connectionist models are mathematical or computer models based roughly on the local structure of nervous systems of animals [STAN95]. Many features of the nervous system are suppressed and that several simplifying assumptions are made [STAN95]. Connectionist reasoning consists of propagating numerical values within this network, that is, modifying the connections established between the different elements in the network. The interesting thing about this approach is that reactive systems could be created which are capable of learning and of having adaptive behaviour by linking perceptions directly to actions, without any explicit intervention by cognitions [FERB99].

The next approach called as kinetic or interactionist hypothesis, which postulates that an individual's knowledge could be considered as a multi-agent system in its own right, the concepts, ideas and representations then being agents of a specific nature which live inside the agents [MINS86] [FERB99].

Another approach for knowledge representation is based on "frames" which is conceptualised with the use of K-lines [MINS86]. In this theory, there exists a conjecture that our brains are composed of a host of agents called K-lines, which could be used to make records of what some of the brain agents are doing at a certain moment. Later, when a particular K-line is activated, this restores respective agents to their respective state. Hence, our knowledge that attaches meaning to the ideas generated is organised in a networked manner with the use of frames and K-lines according to Marvin Minsky [MINS86].

The rules, semantic networks and frames are related to each other since there is close a resemblance to the approach taken by humans. The semantic network is like a graph that consists of nodes and links [CAWS97]. The links are unidirectional connections between nodes where the nodes correspond to objects or classes of objects in the world, whereas links correspond to relationships between these objects [CAWS97]. However, semantic nets seem to suffer from representational adequacies related to representing complex logic which are possible through other techniques such as rules [CAWS97]. The semantic and neural networks have similarities as well as differences. A difference is that the links of a semantic network represents logical relations between concepts and the links of neural networks represents weighted paths along with activation energy [POLL90].

2.4.2 Knowledge Representation based on Frames

A special emphasis was given to the concept of frame based knowledge representation because it was evaluated as suitable to build shared memory models to be used across multiple agents and further frames across multiple agents could be amalgamated for coordination activities. It was also identified based on the Society of Mind theory by Marvin Minsky [MINS86], that frames are more suitable for reaching common-sense reasoning models. Hence, frame-based knowledge representation was selected as the most suitable representation method for the research due to the AAANTS model's inclination to realise some level of common-sense reasoning. A detailed discussion of the frame-based representations is found in the continuation.

In relation to frame-based representations, Marvin Minsky [MINS74] argues that in order to explain the apparent power and speed of mental activities, there ought to be more structure to the chunks of knowledge than there is in logic, and the declarative and procedural aspects of a given chunk must be more tightly connected. Frame Knowledge Representation Systems are also known by a variety of names including semantic networks, frame systems, description logics, structural inheritance networks, conceptual graphs and terminologic reasoners [KARP93]. A detailed description of the structural arrangement of frames together with the control, manipulation and transformation are discussed in Appendix C.

The frames concept based on the original definitions is understood as an ontological commitment and a theory of intelligent reasoning based on insights about human cognition and the organization of knowledge in memory [DAVI93]. The concept of frame is defined as a data structure for representing stereotyped information about a situation or structure that represents knowledge about a limited aspect of the world [MINS74] [REIC91] [DAVI93]. They are formed on the foundation of previous experiences in similar situations and could best be seen as a structure representing expectations of the system about situations of this kind [REIC91].

It was mentioned that the concept of frames as described by Marvin Minsky [MINS86] is used to implement the state representations. Each agent will capture its share of state instances, as experience of the environment grows. These frames are attached to each other in a manner that represents the experience with the environment. The links among the frames are strengthened or weakened with the help of episodic reinforcements given to the agents. Over a period of time, an agent may accumulate a vast collection of state instances, where in a complex environment may become impractical to maintain. In order to overcome such situations humans settle down on standard averages to define similar states and also use heuristics to overcome complex situations. This could be called as common-sense reasoning. In a learning methodology, function approximation techniques and non-monotonic logic could be used to implement such heuristics.

Object-oriented and frame systems have many similar characteristics, but are usually designed for different purposes such as programming and knowledge representation respectively [LASS90] [KIFE95]. Central to the object-oriented paradigm is the definition of types and the creation of instances of these types, and that types share descriptions of structure and behaviour via inheritance [LASS90] [CAWS97]. The high-level definitions of frame systems have close semantic resemblance to object-oriented systems. For example, frames and classes and also attributes and slots demonstrate very close resemblance [LASS90]. These similarities qualify the object-oriented techniques as a suitable mode of implementation of frame-based systems. However, according to Michael Kifer et al [KIFE95], object-oriented systems require considerable improvements to be used for both knowledge representation and implementation.

According to Ora Lassila [LASS90], there are also considerable differences between frame and object concepts. In frame systems inheritance is typically dynamic in nature when compared to object-oriented systems. This allows the default values to be changed during program execution; hence frame systems may allow the use of multiple inheritance paths other than the usual is-a hierarchy. Another useful feature of some frame systems is the ability to create and maintain multiple parallel and nested worlds which is not possible in object-oriented systems [LASS90].

2.5 Background on Collective Behaviour

The concepts of cooperation and coordination represent some type of agreement among a defined set of entities in achieving a common goal. Both coordination and cooperation requires the participation of a collection of autonomous entities. It would be ideal to discuss these two concepts based on MASs where a collection of agents autonomously execute actions for a common cause. The subtle differences between cooperation and coordination are discussed in continuation.

2.5.1 Variants of Collective Behaviour based on coordination and cooperation

Coordination is the process by which a participant of a team reasons about its local actions and the anticipated actions of others to ensure the community acts in a coherent manner [NWAN96b] [JENN95b] [JENN94]. The prime reason for the need for coordination according to N. Jennings et al and Jacques Ferber [JENN95b] [FERB99], is that a single entity does not posses the competence, resources and information to solve a complicated problem. A complicated global behaviour or a solution should be reached with the contribution of a community of agents, each executing actions both sequentially and or concurrently with others, or in other words in a coordinated manner.

In contrast, cooperation is a form of interaction, usually facilitated through communication which requires acceptance and commitment of others for a cause but excludes the need for synchronised actions [MATA94a]. Hence, the synchronisation of actions could be described as the primary differentiating factor between coordination and cooperation. In summary, coordination may require cooperation; but it is important to emphasize that cooperation among a set of agents would not necessarily result in coordination; indeed, it may sometimes result in incoherent behaviour [NWAN96b].

The collective behaviour is described as a subclass of cooperative behaviour and generically denotes any behaviour of a community of agents in a system [CAO95]. The differentiation of collective and cooperative behaviour lies in the fact that the latter should result in the increase of the total utility of the system [CAO95]. Therefore, being collective does not suggest the results of cooperation such as emergence and coordination.

2.5.2 Software Agents Communicating Explicitly or Implicitly

A popular method of interaction among software agents is the use of messages exchanged either implicitly or explicitly [PARU94]. There is considerable amount of research in the area of explicit communication especially in the domain of cognitive agents. Explicit communication requires capabilities such as organisational structuring, contracting, planning, negotiations, arbitration and collaboration [NWAN96b] [FERB99] [CAO95]. Explicit communication is driven by commitments undertaken towards each other for future anticipated interactions. A convention is a means of monitoring a commitment and the circumstances under which a commitment could be abandoned, and how an agent should behave both locally and towards others when one of these conditions arises [WOOL00]. Usually, agents communicate explicitly using patterns of messages called conversations [LACE00a] [LACE00b]. The conversations could be done through direct messages or through mediators (also called as brokers) with the use of performatives[9] [DECK99].

The main substance of explicit agent communication is defined in an Agent Communication Language (ACL) [LACE00a]

[9] A performative specifies the format of any given message and dictates how an agent should respond to messages.

[GENE94]. Two popular communication languages are the KQML and FIPA ACL [LACE00a] [LACE00b] [DIMI98]. According to Michael Geneserath and Narinder Singh [GENE94] [SING95], ACLs could best be thought of as consisting of three parts - its vocabulary, an inner language such as KIF (Knowledge Interchange Format), and an outer language such as KQML or FIPA ACL. For example, an ACL message could be a KQML expression in which the arguments are terms or sentences in KIF formed from words in the ACL vocabulary.

The Speech Act Theory is a high-level theoretical framework developed by philosophers and linguists to account for human communication based on which the ACLs are modelled [LABR94]. It treats communication as an action [WOOL00] and have been extensively used, formalized and extended within the fields of computational linguistics and artificial intelligence as a general model of communication between arbitrary agents [LABR94]. There are three different aspects of speech acts: the locutionary act, or act of making an utterance, the illocutionary act, or action performed in saying something, and prelocution, or the effect of the act [WOOL00] [LABR94]. The speech act theory has been used to build general grammar out of which arbitrary protocols could be constructed, and agents that understand this grammar could evolve new protocols for their conversations as they operate [PARU98].

A communications protocol determines how conversation among agents is structured with respect to an ACL. The protocols determine a fixed set of options that a conversation could follow, and this structure is fixed prior to actual communication [PARU98]. There are different communication protocols such as directive, voting, negotiation and speech acts [PARU98]. Especially in voting and negotiations, a series of messages are exchanged among the participating agents before a decision is reached.

The explicit communication within cognitive agents requires top-down view of the execution needs at each level of the organisation in terms of sequence, concurrency and interdependencies of actions. This means that some level of understanding of the capabilities of participants and task distribution among agents should be known prior to commitment. This technique is opposed to the emergent behaviour expectations of this research which needs a bottom-up strategy. Emergence demands dynamic coupling among the participants consequently a static top-down view of participation cannot be maintained. Hence, implicit communication that facilitates such dynamism is deemed suitable for this research.

2.5.3 Coordination Models and Structures

When discussing emergent models, top-down hierarchical and bottom-up emergent models were given special consideration within the research community. It is identified that most mathematical models of natural systems fall into these two categories [SUMP00]. In top-down models, the relationships of interest are between variables that capture the global properties of a natural system and mathematically as they are expressed in an ordinary or partial differential equations and bottom-up models start from a description of local interactions which usually involve algorithmic descriptions of individuals [SUMP00].

The structure of an agent community is also an important factor in realising the coordination model. The two popular control structures that complement the above discussed coordination models are hierarchical and egalitarian control structures [FERB99]. The hierarchical structures (defined as monolithic centralised/distributed [DORI93]) are associated with fixed coupling where commands are passed down the hierarchy with each subsequent agent obeying the command given by a superior.

The egalitarian structures are more characteristic of organisations in which all the agents participate in a uniform way in deriving the final decision. However, according to Marco Dorigo [DORI93], distributed architecture, activities and intelligence are preferred in the multi-agent domain when compared to monolithic systems. The arguments for supporting distributed nature are that most problems are physically distributed and heterogeneous, distribution of processing power and rapid adaptation to the environment.

2.6 Lessons from the Nature

The coordinated behaviour of distributed elements is still a conundrum to the artificial intelligence community. However, the natural world is abundant of examples with the blessing of millions of years of evolution where genetic mutations have resulted in individualistic and group level distribution and emergence of complex behaviour. Consequently, researchers from various disciplines such as ethology, myrmecology, synecology and ecology have indirectly contributed immensely to the behavioural aspects of MASs.

2.6.1 Lessons from Natural Insect Colonies

Insects have secured special recognition within the animal world in relation to collective behaviour and could be defined as the pinnacle of altruistic social behaviour adopted by any animal form on the planet. Out of all social insects, ants have secured the highest level of respect in relation to producing social altruistic behaviour. This argument is justified from the evidence disclosed by B. Holldobler et al, where only 13,500 species of highly social insects are known out of a grand total of 750,000 that have been recognised to-date by biologists and 9,500 of which are represented by species related to ants [HOLL94]. A fact that is oblivious to most humans is that these disproportions confirm that we live in a planet that is in fact dominated by ants in number and mass.

The ants are aware of no more than a few centimetres around their bodies and no more than minutes of time into the past and have no mental construct of the future [HOLL94] [GORD99]. Yet, they have survived as colonies for nearly ten million generations when compared to hundred thousand human generations [HOLL94]. It is evident from this fact that ant group tactics, though being simple, has enabled them to adapt and evolve to different conditions. This element of adaptive behaviour in ants with the use of simple coordination techniques have been the prime inspiration to this research.

It is a well-known fact that the amazing success of the ants is due to the synergy arising from the members of the colony [GORD99]. The synergy of actions at this level of efficiency is primarily made possible by the advanced development of chemical communication capabilities [GORD99]. The release of a medley of substances from different parts of the body called pheromones, stimulates the other members of the community within close proximity to perform actions such as alarm, attraction, nursing, food offering, and a diversity of other activities [HOLL94] [GORD99]. The other counter-part of this mechanism is the sensation of the chemicals, which usually is done with the use of sensory receptors in the antennas of the insects [GORD99]. Hence, the communication of ants happen when individuals emit pheromones to the environment and others within the local vicinity senses the signal and activate accordingly by executing a sequence of actions that are either innate in genomes or learnt from experience. It is confirmed that some complicated responses such as self-grooming and regurgitation, appear to be wholly programmed so that the insect performs them more or less expertly with no prior experience [HOLL90], which in another sense justify innate behaviour.

The key attraction of ant behaviour to this research is their ability to cooperate and coordinate to accomplish complex behaviour while being simple in structure and capabilities. Hence, it is a key objective to identify the techniques employed by some species of ants for co-ordinated behaviour. The experiments of B. Holldobler et al [HOLL94] have discovered that weaver-ant workers deploy several messaging techniques to guide nest mates in different types of activities and situations discussed below.

1. A chemical substance could be laid down as a trail and combined with a particular body movement, either a little dance or a touch of the antennae to indicate a newly found food source.

2. When a scout worker finds a suitable location for a new nest, she then lays a rectal-gland trail of pheromones combined with touch signals that convey a message of direction to the other ants in the colony.

3. When an enemy is encountered, the workers broadcast an alarm by laying short looping trails around the intruder, drawn with substances smeared on the ground from the Sternal[10] gland.

4. Some weaver ants have an alarm system for intruders based on multiple pheromones with multitude of semantics such as arousal, search for source of trouble, attack and aggressiveness.

Further, ants employ techniques such as tapping, stroking, squeaking and body contact dancing for cooperation [HOLL94] [GORD99]. However, the most preferred mode of communication is through pheromones. It is identified that an ant species generally employ between 10 to 20 such chemical "words" and "phrases" each conveying a distinct but very general intuition such as attraction, recruitment, alarm, identification of other casts,

[10] A gland pertaining to the sternum or lower portion of the body that secrete alarm signals in ants and other insect species.

recognition of larvae and discrimination between nest mates and strangers [HOLL94].

The theory of pheromone design is based on the concept of active space [HOLL90], which is the zone within which the concentration of a pheromone is at or above threshold concentration and the effect of the pheromone could be defined temporally and spatially [HOLL90]. However, the complexity of behaviour of an insect overwhelms the repertoire of pheromones possessed by an individual of a colony.

It is identified that pheromones and the corresponding actuations may have one to many relationships due to the fact that a chemical could relay different messages based on the context, role, time and space [GORD99] [HOLL90]. This was demonstrated from the experiments carried out by Deborah Gordon [GORD99] where ants covered in oleic acid were treated as dead by nest maintenance workers and treated as food by foraging ants. An important fact that stems from these observations is that depending on the currently occupied activity and the role of an ant, a given pheromone may trigger different actions within the individuals of a colony.

This fact could be further explained using the concept of Stigmergy [BECK94]. The consequences of behaviour affecting the subsequent behaviour could be called as Stigmergy [BECK94]. In this backdrop, it should be understood that communication among ants is indirect, via pheromone deposits that change the state of the environment, rather than via message passing with a handshake [KEIL03]. Stigmergy could be explained by indirect communication prevalent in insects which use volatile pheromones in the environment to deposit short-term memory [BECK94]. The releasing and sensing of pheromones to be affected locally within a large colony of insects could be converted to a myriad of behavioural patterns with the use of Stigmergy.

The super organism theory[11] of ants is another important way of looking at group work [HOLL94]. It describes an ant colony as a super organism consisting of the queen as the reproductive organ and the workers as entities that support brain, heart, gut, and other tissues. The exchange of food among the colony members is equivalent of the circulation of blood and lymph. This theory has been rejected in concept during the early 20th century but as of recently, there seems to be a resurgence of interest in the research community [HOLL94].

2.6.2 Coordination Lessons from the Nature and Facts for Emergence

In natural systems, the local interactions that result in the evolution of complex and stable behaviour are difficult to analyse using traditional, top-down approaches [MATA94a] [PAOL97]. It is believed that in order to reach the level of complexity found in nature, the behaviour must be generated through an interaction-driven, incrementally refined process such as emergence [MATA94a].

[11] This is described in the work of William Morton Wheeler's statements on ant colony super organism theory and mentioned in B. Holldobler *et al* [HOLL94].

The term emergence could be explained as a property of a system as a whole not contained in any of its parts and the produced behaviour would be more complex than the behaviour of the individual components [PFEI01] [PARU97b]. Therefore, the movement from low-level rules to higher-level sophistication could be called as emergence [JOHN02]. A collection of local interactions would not truly be considered emergent until those local interactions resulted in some kind of discernible macro behaviour not present in its constituents [JOHN02]. Such behaviour could be called as "emergent behaviour," because it emerges from the interactions within the overall system, often in ways not intended by the original designers [PARU97b].

There are three main reasons for emergent cooperation in biological societies: pair bonding, kin selection and altruism (direct and reciprocal) [RUI02]. Direct altruism that results in relation to pair bonding and kin selection is found especially in insects where an individual has no life without the colony and genetically imprinted caste[12] and role/task[13] mechanisms determines the individual behaviour [GIFF00]. Hence, the decision to become altruistic is inherent to the insects and does not require a cognitive process to handle this nature of behaviour.

However, reciprocal altruism is found in more higher-level animals where increase in the cognitive power of the individuals helps to

[12] Broadly defined, as in ergonomic theory, any set of individuals of a particular morphological type or age group, or both, that performs specialized labor in the colony [HOLL90].

[13] A set of closely linked Behavioural Acts (BAs) though different in nature could be defined as a **role** and a particular sequence of acts that accomplishes a specific purpose, such as foraging or nest repair could be called as a **task** [HOLL90].

maintain mental accounts or balance sheets by which it keeps track of its debts to others and vice versa, as well as who cooperated and or cheated in the past [GIFF00]. Therefore, reciprocal altruism needs long term memory and a formula within each individual in the colony to be maintained and evaluated when assistance is requested by another participant in the accepted group. However, some rare types of altruism practiced by some humans that sacrifice their lives to the good of the others should be treated as special.

There seems to be a relation between morphogenesis and sociogenesis with respect to emergence. The set of procedures at the level of an organism by which individual cells or cell populations undergo changes in shape or position incident to organismic development is called as morphogenesis [HOLL90]. The definitive process at the level of the colony is sociogenesis, the procedures by which individuals undergo changes in caste, behaviour, and physical location incident to colonial development [HOLL90]. Both sociogenesis and morphogenesis processes are perfect examples of emergent systems where local elementary actions result in complex macro behaviour and structure respectively. Sociogenesis and morphogenesis though heterogeneous in origin, gives us a hint that there is a fundamental theory behind emergence which is still to be resolved.

The participants acting locally by paying attention to neighbours as opposed to following direct orders from a superior seems to result in macro emergent behaviour in ant colonies, embryo development and other type of swarm systems [JOHN02] [GORD99]. Further, there are many interpretations about the relationship among ants and brain cells. It is stated that self-organisation of neurons into a brain-like structure, and the self-organisation of ants into a swarm are similar in many respects [CHIA95]. Memories are believed to be written as a stronger coupling among individual, or groups of neurons and these couplings are strengthened by neural co-activity much in the same way that the pheromonal field is preserved or strengthened by coherent frequent ant traffic.

Pareto efficiency is another solution evaluation criterion that takes a global perspective [WEIS00] [PIRJ99] [PETR95]. Again, alternative mechanisms could be evaluated according to Pareto efficiency by comparing the solutions that the mechanisms lead to. A solution x is Pareto efficient or Pareto optimal, if there is no other solution x' such that at least one agent is better off in x' than in x and no agent is worse off in x' than in x [WEIS00] [PIRJ99] [PETR95]. So, Pareto efficiency measures global good, and it does not require questionable inter-agent utility comparisons.

Particle swarm is another system that argues self-organising through dynamics of local rules [KENN01]. It argues that when individuals in a group adjust towards the success of their neighbours, the population converges into an optimal arrangement [KENN01]. However, it is investigated that the population landscape settles on multitudes of local optima with the result of applying a particle swarm algorithm which to some extent is similar to evolutionary algorithms [KENN01]. Similarly, social interaction results in cultural convergence on patterns of beliefs, and culture results in relatively good cognitive performance [KENN01].

Finally and importantly, Marvin Minsky [MINS86], describes the mind as composed of separate "proto-specialists", where each would be concerned with some important requirement, goal or instinct and equipped with special sensors and effectors designed to suit its specific needs. Genetically, the swarms of social ants and bees are really multi-bodied individuals whose different organs move around freely [MINS86]. However, most animals economise by having all their proto-specialists share common sets of organs for their interactions with the outer world [MINS86].

In summary, the above discussed natural phenomena such as morphogenesis, sociogenesis, Pareto efficiency, embryo development and society of mind seems to have a special relationship to emergence. Hence, it could be argued that a basic set of coordination rules that govern emergence is prevalent in both organism and society level.

2.7 Chapter Summary

The concepts in this chapter act as the background for the research discussed in rest of the chapters. It had been a daunting task to select the suitable concepts, technologies and framework that suit the research objectives. The overall discussion revolves around the necessities of creating artificial intelligence, defined based on the AI Mix, which consists of Coordination, Adaptation and Representation.

There are several methodologies of implementing artificial intelligence ranging from traditional cognitive approaches to distributed approaches. The concept of software agents falls under distributed approaches and was selected as more suitable for this research because the inherent characteristics such as autonomy, adaptability, communication, rationality and mobility clearly complements the research objectives related to emergence.

The next clear direction taken from this chapter is the use of reinforcement learning as the learning methodology of this research. The thesis is related to building an adaptive layer of functionality on an innate base of capabilities. The objective of the upper most layer is to facilitate the agents with capabilities to survive in the environment. Hence, it is clear that the environment does not act in a supervisory nature but in a reinforcement manner. A major portion of this chapter describes the nature of reinforcement learning which further act as a base for the discussions in chapter 4. A special emphasis was given to the concept of frame based knowledge representation because it was evaluated as feasible to conduct shared memory models to be used across multiple agents and further frames across multiple agents could be amalgamated for coordination activities.

The latter part of the chapter describes the coordination models that support emergent behaviour based on the research conducted on natural systems such as insects, swarms, cell development and traffic movements. The methodologies of these natural systems would be taken as inspiration as well as the foundation for the AAANTS model realised in the rest of the chapters.

Chapter 3

Behavioural Congruence through Implicit Communication

3.1 Introduction

The behaviour of an animal could be disintegrated into constituent composite actions. These composite actions could be iteratively sub-divided until reaching the elementary actions that are non-divisible into meaningful sub-actions. These elementary actions are collectively referred to as Atomic Actions (AAs) within the context of this dissertation. It would be argued that the miracle of intelligent behaviour lies in the dynamic and proper coordinated execution of AAs in the temporal dimension.

The survival of an animal could be described as executing appropriate behaviour to the ever changing environmental sensations; some pre-programmed and others learnt. An animal is born with a repertoire of inherent pre-programmed actions which could be called as innate which is usually in the basic form of AAs. After birth, the innate AAs are amalgamated and iteratively restructured to create more complex and useful behaviour based on the supervisions and reinforcements from the environment. This is called as achieving behavioural congruence[14]. This process had created myriad of heterogeneous intelligent behaviour throughout the planet.

A primary objective of this research as mentioned in Chapter 1, Section 1.3, is to model behavioural congruence on artificial substance, more suitably using software agent technology. Further,

[14] As an analogy to behavioural congruence, the resulting relation between the structures of the coupled systems is known as **structural congruence** [PAOL99] and it is to be found particularly between organisms that engage in interactions repeatedly and recursively.

it was discussed in chapter 2 that software agents could take an individualistic or group approach to demonstrate intelligent behaviour. However, implementing behavioural congruence using a cognitive agent model could be considered relatively trivial when compared with the use of a group of agents, where each agent is responsible for a subset of actions that should co-operate with each other to demonstrate emergent behaviour. The complexity in behavioural congruence lies in the fact that, none of the agents have any understanding about the end product – the emergent behaviour, though being driven to cooperate.

This research concentrates on group approaches for cooperative behaviour among agents where several agents contribute towards the global interest. This chapter presents a methodology based on an inter-disciplinary coordination model that harnesses on emergence to derive complex behaviour. Henceforth, the use of "AAANTS coordination model" or merely "coordination model" refers to the coordination methodology conceptualised within this dissertation.

The AAANTS coordination model describes a novel methodology based on "implicit cooperation" which is distinct from traditional intentional coordination strategies. The coordination methodology based on implicitness was inspired by pheromone based chemical information exchange prevalent in insects as discussed in Section 2.6. These chemical messages could be described as distributed (disseminated spatially), dynamic (degrading in concentration conveys different interpretations) and implicit (message placed in the environment is not intended for a particular individual but to any party interested and capable of participation). The ants do not intend to communicate with "specific" individuals within the proximity when passing a message, hence the reason to be called "implicit". The decision taken to stimulate a neighbour is a result of adaptive and stochastic nature.

3.2 AAANTS Coordination Model based on Patterns and Emergence

The "AAANTS Coordination Model" was conceptualised based on the inspiration from the natural emergent systems. The model encompasses aspects such as identifying sensory patterns, relationship among actions and sensations and team formation among agents for coordination. The interactions among agents act as perturbations and the system achieves congruence with the use of reinforcements. The resulting model consists of heuristics and algorithms that could be used to implement an agent system that demonstrates emergent behaviour.

3.2.1 The Agent Life-Cycle

A key requirement for the survival of a community, based on the insight gained from insects is the necessity of a life-cycle. A life-cycle encompasses the creation and destruction of an entity together with different states in between that correlates to the changing functional objectives demanded from that entity. Based on these fundamentals, the concept of an AAANTS agent life-cycle was fused to the AAANTS agent model which could be described as a contribution of this research The Figure 3.1 depicts the AAANTS agent life-cycle with the breakdown of states and transitions.

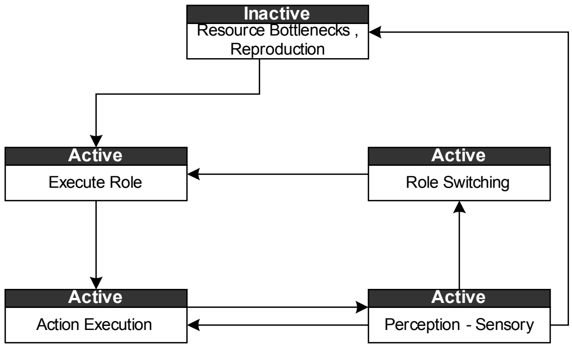


Figure 3.1: State transition related to the agent life-cycle of the AAANTS model.

The AAANTS system is programmed to release a limited number of agent instances during the birth of a colony. These agents are segregated into casts by differentiating the innate characteristics (attributes and action templates) of each individual. When an agent colony matures, initial participants of each respective caste would regenerate new agent instances based on the demands from the environment. Therefore, all agent instances would adhere to a life-cycle as depicted in Figure 3.1.

An agent's life within a caste could be primarily organised into two states: inactive and active. All agents when initially instantiated, start in the inactive state which could be called as the starting state of all agents (Figure 3.1). The agents in the inactive state are incapable of sensing the environment changes and performing any actuations. The inactive agents transit to the active state with an activation command from the other active agents. This is analogous to the recruitment alarms applied in insect colonies. The agents in the active state would execute a series of coordinated actions that belong to the repertoire of roles within the respective castes. The agents may periodically switch back to the inactive state based on resource limitations, reproductive needs of the colony and ultimate termination/elimination from the colony. Hence, the inactive state could be described as both the starting and ending states of all agent instances within an AAANTS based system.

Each agent while being in the active state would be trapped in an iterative cycle of sensations and actuations as represented in Figure 3.1. The sensations are the primary triggers for actuations and the agents adapt as a community based on the reinforcements from the environment (explained in detail in sections 3.2.2.1 and 4.5.1). During this cycle of activity, an agent may consciously migrate to a role switching state that would change the current role of activity. The role switching would facilitate the migration of agents to roles that are of demand in relation to the changing environmental conditions. However, the probability of migration due to role switching is very minimal when compared to mundane action execution.

3.2.2 Creating Behavioural Concentres with Atomic Actions and Action Templates

The term Atomic Action (AA) could be defined as an action that cannot be further subdivided into elementary actions. For example, in humans, the contraction of a homogeneous muscle could be thought of as an AA. A given AA could produce different effects based on the intensity and the degree of temporal progress. If the minimal duration of an atomic action a is defined as t, a.t represents the minimal temporal result of executing action a. However, changes in the temporal dimension of executing the same atomic action a, would produce different end results – e.g. [a.2t], [a.3t], etc. Within the boundary of this research, AAs are considered innate and could only be enhanced within the dimensions of time and intensity.

3.2.2.1 The Action Templates

The concept of the Action Template (AT) is introduced herewith as the primary method of grouping AAs to define behaviour. The concept of AT could be considered as an original contribution of this research in relation to the use with multi-agent systems and frame-based knowledge representation. Though there is common use of templates in relation to data, information and knowledge [RUSS95], there is minimal research done in relation to use of templates for behaviour. A template could be defined as a generalisation of related instances that determines or serves as a pattern [GAMM95]. Further, the concept of templates is used analogously to represent the concept of a class in object-oriented programming and design methodologies. A template could be also considered as a description of an aspect of a task. In-line with these definitions, a definite collection of AAs executed in concurrency and or sequence in relation to environmental sensations could be called as an AT.

An AT is analogous to a class template in an object-oriented system. The object-oriented class template that defines an AT consists of necessary attributes and methods to implement the capabilities of its constituent AAs. A class of an AT would not be of any use without being instantiated. An AT could be instantiated by several agents, where each agent would represent one or many AAs defined in the AT. For example, the AT depicted in Figure 3.2 consists of four AAs. One possible arrangement of instantiation is to have four agent instances creating four instances of this template where each agent executes their respective AA in coordination with other agents. Another arrangement would be to instantiate two agents and two AT instances where each agent takes responsibility to manage two AAs. However, it should be stated that within this research an agent instance is only attached to a specified AT, which means an agent is assigned to a definite AT type.

Concurrency is a basic fact of nature for achieving complex behaviour. The survival in the environment demands concurrent threads of attention to both sensations and actuations. It should be noted that due to the need for concurrency, the AAs within a single AT could be contributed by several agents.

The methodology used by agents to collectively execute synchronised tasks without the knowledge of the overall outcome was given special emphasis during the conceptualisation stage of this research. According to Keith Decker *et al* [DECK95], the coordination problem of choosing and temporally ordering actions is more complex because the agent may only have an incomplete view of the entire task structure of which its actions are a part, the task structure may change dynamically and the agent may be uncertain about the outcomes of its actions.

The type and sequence of AAs and their synchronisation with sensations for initiation and termination uniquely differentiates ATs from each other. Hence, in summary, three aspects are important to an AT: types of AAs, maximum temporal exposure of each AA and the influence of sensations (environmental sensations and the temporal progress of other AAs within the same AT could also be served as a sensation) for the purpose of initiation and termination effects of each AA.

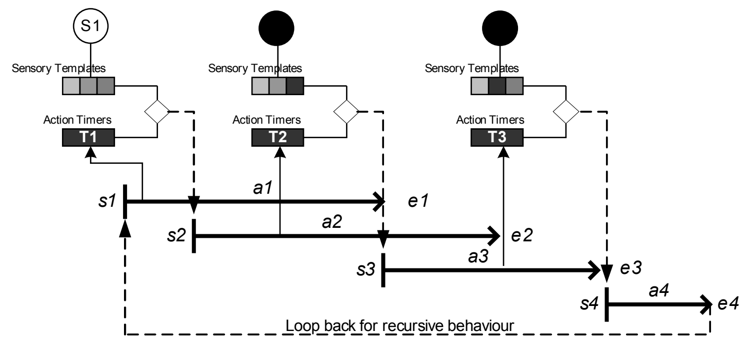


Figure 3.2: Action template with a defined sequence of atomic actions

Figure 3.2 depicts an AT defined using four AAs (a1, a2, a3 and a4). Here each AA is constrained with a start and a finish. The symbols s1, s2, s3 and s4 represent the states that trigger the AAs into activity and symbols e1, e2, e3 and e4 represent states that inhibit the execution of AAs (Figure 3.2). These states are internal representations within the participating agents of an AT. These internal states relate to the external sensory triggers from the environment. These external sensory elements are represented in Figure 3.2 using symbols S1, S2 and S3 as attachments to the Sensory Templates. Hence, the Sensory Templates that are modelled based on the frame-based technique is the knowledge representation scheme used in an AT.

Each started AA instantiates a timer that measures the temporal progress of that atomic activity. These timers are represented by symbols T1, T2 and T3 in Figure 3.2. A started action could finish due to lapse of allocated maximum execution time or due to a trigger from an external sensation. The maximum allocated time of each AA would be defined during the creation of the AT. Further, the initiation of actions would be triggered from the temporal progress of other dependent actions within the same template and or sensory stimulation from the environment.

An important aspect of the AT concept is in the methodology used for action synchronisation. An AT should be first instantiated to facilitate the defined behaviour. Subsequent to the initial instantiation, the first action in the sequence would be activated. However, there could be situations where several AAs that belong to an AT are activated simultaneously at the initiation based on the stochastic nature of the action selection mechanism. An ongoing action would publish the temporal progress within the respective domain, and other participants could use this information for coordinated participation. Therefore, both the temporal progress of the other actions and the sensory information from the environment is used for action coordination. The coordination sequence is improved over a period of time due to the reinforcements received after executing an instance of a template.

The AAs could be described as innate to an intelligent entity. However, the ATs could be formed both in terms of innateness and adaptations. The innate ATs would be ready to use though with further fine-tuning through environmental supervisions and or reinforcements. The adaptive ATs would be created through a stochastic process where innate AAs are randomly selected to form novel behavioural structures. The exploitation, exploration and credit assignment methodologies of the AAANTS learning model is described in section 4.3 and 4.5. Further, ATs would be able to form hierarchical or lateral bonds with each other, again through a stochastic process to create complex behavioural outcomes. The AAANTS model conceptualises both flavours of ATs but the experiments are focussed on the innate ATs that are refined through reinforcements.

A similar approach is taken in leaning systems like ALECSYS [COLO93], where the learning "brain" of an agent could be designed as the composition of many learning behavioural modules. The modules are called as basic behavioural modules which are connected to sensory and motor routines that learn from external stimuli. The behavioural modules of ALECSYS could be made analogous in concept to ATs discussed above. Simply, AAs are like the bricks and templates are like different wall types of a building, where different combinations of walls could be used to create buildings of diverse architectural complexities.

The concept of the AT would also be similar in some extent to behavioural assemblages [BALC97]. According to Tucker Balch [BALC97], groups of behaviours are referred to as behavioural assemblages. One way that behavioural assemblages may be used in solving complex tasks is to develop an assemblage for each sub-task and to execute the assemblages in an appropriate sequence. The resulting task-solving strategy could be represented as a Finite State Automaton (FSA) and the technique is referred to as temporal sequencing.

The use of ATs consisting of multiple AAs in modelling coordinated behaviour could be further explained with the use of a robotic arm movement example. Figure 3.3 depicts a model arm with 3 joints – shoulder, elbow and wrist which is analogous to an upper limb of a human. Each of the joints J1, J2 and J3 is moved by AAs a1, a2 and a3 respectively as depicted in Figure 3.3. The execution of each atomic action in different temporal values would result in the respective component of the arm changing the angle of movement (Q1, Q2 and Q3).

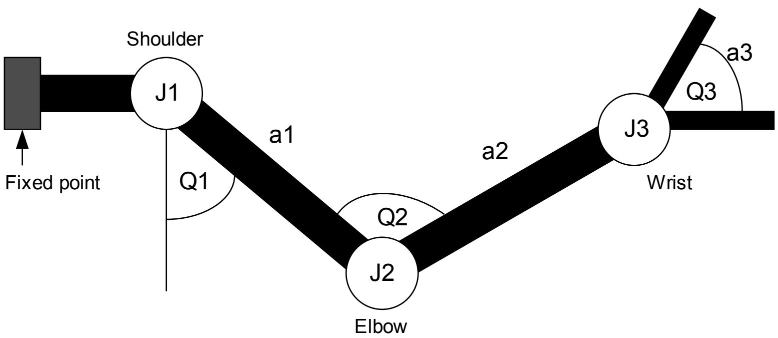


Figure 3.3: Robotic Arm Model with 3 degrees of Freedom – movement of a single plane

The AT responsible for the movement could be defined by actions a1, a2 and a3 that initially execute in sequence in order to change the respective angle to perform myriad of tasks. The temporal execution of these AAs could be easily understood when comparing with the AT represented in Figure 3.2.

3.2.2.2 Behavioural Concentres

The groups of actions in an AT that consist of AAs are the basis for building complex behaviour. A group of AAs within an AT (depicted in figure 3.2) that are executed in a coordinated manner may contribute in behaviour, fully or partially to a Behavioural Act (BA). Hence, one or many ATs may represent a BA. The concept of a BA is similar to the definition found in myrmecology for a collection of elementary actuations [HOLL90]. For example, in Figure 3.2, the depicted AT with actions a1, a2, a3 and a4 could represent a BA, or several ATs that are coordinated with each other could also represent a BA. Further, a collection of closely linked BAs could be defined as a Role where a Task could be differentiated as a similar sequence of BAs that are coordinated.

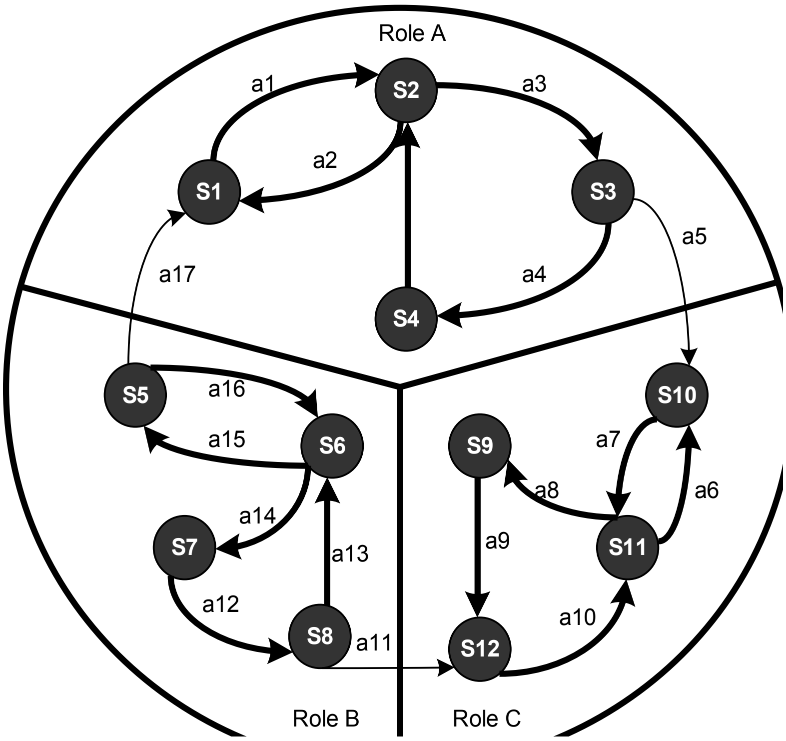


Figure 3.4: Ethogram for the transition of behavioural actions across different roles

A popular method of depicting a behavioural repertory is by the use of an ethogram, which incorporates repertory of a caste, transition probabilities of acts and the time distributions spent on each act [HOLL90]. The Figure 3.4 represents an ethogram that depicts the roles within a group of entities and the states and actions that facilitate the transition among roles. It should be noted that some actions (actions a5, a11 & a17 in the ethogram – Figure 3.4) enable a role to be navigated to states of another role.

Roles could also be described in terms of cohesion and coupling of ATs when multiple ATs contribute to a role. There exists high cohesion among the AAs that belong to an AT. The ATs that belong to a specific role should have higher coupling within them than with others external to the role.

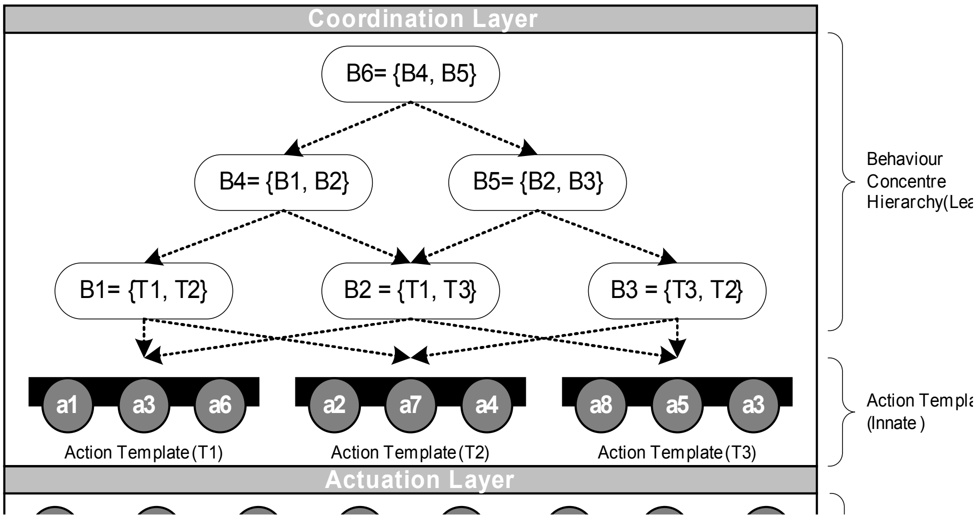


Figure 3.5: Conceptual action breakdown structure of the AAANTS coordination model.

The Action Breakdown Structure (ABS) depicted in Figure 3.5 would be a good approach to explain the rest of the behavioural complexity of the AAANTS coordination model. The ABS conceptualised within the AAANTS model is an original contribution of this research which clearly aligns in realising the objective of demonstrating congruent behaviour as a result of a static innate layer refined through a reinforced adaptive layer of behaviour. Hence, the behavioural structure the ABS is segmented into two primary layers of functionality based on the innateness and adaptability. The actuation layer represents the raw AAs that are innate in nature and less complex. As examples, the basic contraction of muscles, release of enzymes and hormones, change of chemical composition in animals are analogies to these types of actions. Hence, AAs are the building blocks of any complex behaviour.

The ATs represented within the coordination layer in Figure 3.5 are responsible for grouping AAs into elementary chunks of coordinated behaviour. However, these templates would be useless without being coordinated with other ATs to perform more complicated roles. The AAANTS model introduces the concept of Behavioural Concentres (BC) [RANA05] as the enabler for coordination among the ATs. The BCs could be described as high-level ATs that link up other constituent ATs to form more complex behavioural assemblages. These BCs are created, adjusted and destroyed based on the reinforcements from the environment.

It is assumed that the innate repertoire of AAs should suffice the expected behaviour of an individual. However, absence of a particular behaviour in an individual does not imply that relevant AAs are missing. Many of us possess the atomic actuations in the upper limbs to become an artist, though few of us are capable of such coordinated behaviour. Further, many of us have the innate AAs to play a violin, though few of us could. Therefore, the BCs and ATs are important in harnessing the capabilities of AAs. In most in-born talents such as art, music and athletics are mostly due to the inherited ATs. Hence the assumption is that some types of special innate ATs are required to full-fill some higher-level complex behaviour. However, even with inherited ATs, without proper environmental adaptations to build up BCs could be called as a "waste of talent" by most of us.

3.3 Heuristics for Emergence

The subsequent sections would use the pheromone usage of insects to formulate six heuristics for the AAANTS coordination model. These heuristics would introduce the emergent capabilities to the already described AAANTS coordination model. These heuristics are inspired from the capabilities of natural insect world discussed in Section 2.6.

3.3.1 Heuristic 1: Locality of actuations and sensations – The view-point of the overall active space is limited and restricted.

In an ant community, pheromones are used for local communication within a restricted spatial and temporal dimension. This aspect is instrumental in facilitating the distributed control nature within insect colonies.

Locality[15] is based on spatial dimensions for most animals, with examples ranging from insects and embryonic cells to swarms. However, limited locality could also be gained by establishing channels of communications even across spatially distant entities. Hence, in respect to software agents, locality is defined by the limited communication interactions among a group of agents and restricted information in the sensory signals from the environment. This heuristic explains that agents tend to synchronise and associate with others that are local to their current existence.

[15] Locality could be explained by Euclidean distance among the sensory generators and the receptors. The intensity of the sensation would be primarily based on the Euclidean distance between the point in space pheromones are released and the point in space of the sensory receptors.

Therefore, global situation is not perceived by any agent and there is no central authority that knows about it either.

3.3.2 Heuristic 2: The effect of sensations may differ based on the context of behaviour

An ant would execute a series of actions in performing a task or role. The transition from one action to another would be based on a trigger from a sensation. The activity switch is a product of the current action and the sensation. Hence, a homogeneous sensation may trigger several subsequent actions depending on the present action.

The effect of sensations could be gauged by the immediate actuations. However, a pattern of sensation may have different effects based on the progress within a role. This could be called as the context or a situation that is related to a certain aspect of an on-going task. This aspect was described earlier using the concept of stigmergy in section 2.6. Within that section, an ideal example was described (the effects of oleic acid on different castes) where a single pheromone may convey different messages based on the current activity of ants within a given proximity. Therefore, sensations and situations should be matched before the actuations.

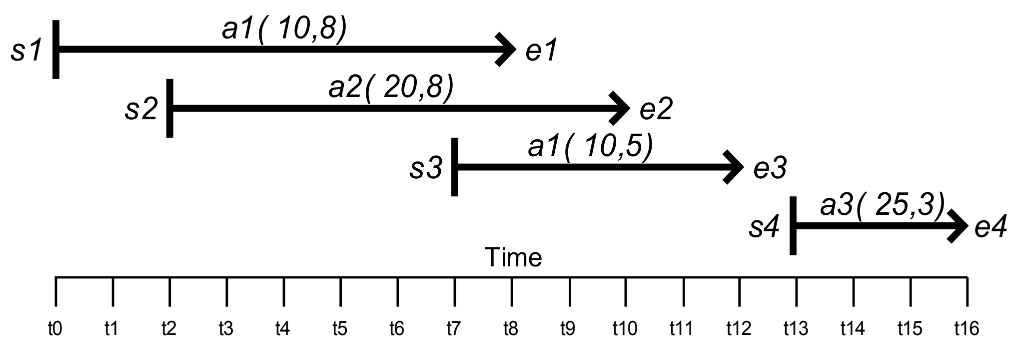


Figure 3.6: Situation identification based on action type, strength and duration.

A situation could be described using a series of states. The same sequence or pattern of states that occur iteratively represents the same situation. Hence, identifying a situation correctly would be very useful in activating the appropriate behaviour for that situation. Figure 3.6 depicts a situation related to the execution of actions a1, a2, a3 and a4 within a temporal frame of 16 units. When a situation is identified, the agents should instantiate and trigger the most appropriate AT. Therefore, the most primitive level of control to handle a situation should be embedded in an AT. The formula of the Situation Index for quantifying a situation as represented in Figure 3.6 is discussed in section 4.6.3.

The AAs in an AT would get activated in sequence or in concurrency as described in section 3.2.2. The participating agents of an AT would refine the behaviour in line with the overall outcome for a specific situation based on the reinforcements from the environment. The agents may explore different sequence patterns of execution in a stochastic manner, which would lead to some AAs conflicting with each other in achieving the final goal. These conflicts are neutralised by the credit assignment process (section 4.5) that eliminates unfit behavioural patterns of participating agent instances.

3.3.3 Heuristic 3: Sensations are organised into patterns

The next objective is to identify the methodologies used to convey myriad of sensory patterns with the use of a handful of pheromones. One approach to achieve this objective is to create a message using a pattern of basic chemical constituents. This is analogous to creating meaningful bytes of information using a collection of bits where a byte is identified as a unique pattern of bits. According to Bert Holldobler *et al* and Steven Johnson [HOLL90] [JOHN02], it is possible to create a compound message by using a graded sequence of concentrations of the same substance. Hence, information could be exchanged using a collection of elementary sensations organised into a pattern. A pattern of information could be created using a single Pheromone with graded concentrations, using multiple Pheromones or with the use of a combination of these two strategies.

The sensations are temporally discretised and could be organised into patterns. Further, the patterns could be based on single and multimode sensory modalities mixed through the temporal dimension. Another dimension to sensory patterns could be added with the use of temporal relatedness of temporally adjacent sensory frames. The agents could use the identified patterns with respect to the reinforcements received from the environment to adjust execution variables of AA within and across AT instances. The pattern identification and quantification of sensations is further described in section 3.4. The pattern identification is done through a technique called as fuzzy clustering where the closest previously recognised pattern is taken as a match. Hence, the fuzzy clustering techniques assist to resolve uncertainty in sensory identification.

Figure 3.7 shows two patterns of sensations created from a single chemical type using different concentrations. Each circle represents a patch of pheromone emission which is spatially apart from each other for an ant to differentiate the sensation. This fact is used by Alfred Wurr [WURR03] to explain the use of stigmergic markers. There are two types of markers: homogeneous markers convey simple patterns to repel from local maxima and heterogeneous markers are dropped in sequence to encode a pattern [WURR03].

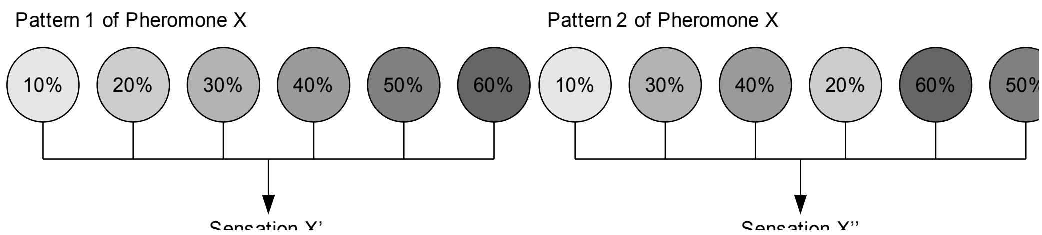


Figure 3.7: Creating sensory patterns using a single pheromone by discrete distribution of concentrations[16].

A single substance has to be either temporally or spatially discretised to convey a pattern of information. Temporal discretisation is very common even in human communication where a message is broken-up into sentences, words and phonemes. Phonemes are temporally sequenced by the initiator and perceived by the receiver in the same sequence. The perceptive organs together with the cognitive aspects are able to assemble them into meaningful messages.

16 The concentration listed is the % above threshold concentration at which the receiving animal responds.

Similarly, insects smear pheromones along the passages they travel and create patterns of messages in the spatial dimension. The messages also have a temporal dimension based on the period of retention of message attributes within a given medium of transport. With respect to pheromones, the concentration reduces at a rate due to evaporation based on the environmental conditions. The varying strength of pheromones due to evaporation with time may offer special intuition to foraging ants such that a pheromone path with a positive or negative gradient may lead to a food source [HOLL90].

The strategy of using multiple pheromones for communication is much more intuitive than using a single pheromone. Figure 3.8 shows that a pattern consists of a number of chemicals (in ants usually secreted from several glands) arranged either spatially or temporally. The distinct bits of chemical sensations are collected to form a sensory pattern with the help of perceptive apparatus. The advantage of using multiple chemicals is that if the receiver has the capability to sense different chemicals concurrently, the patterns need not spread across spatial and temporal dimensions. For example, an ant may smear several types of chemicals in the same position to convey a specific signal and the receiver arriving at the location may sense all these chemicals concurrently using several sensory apparatus to decipher the signal.

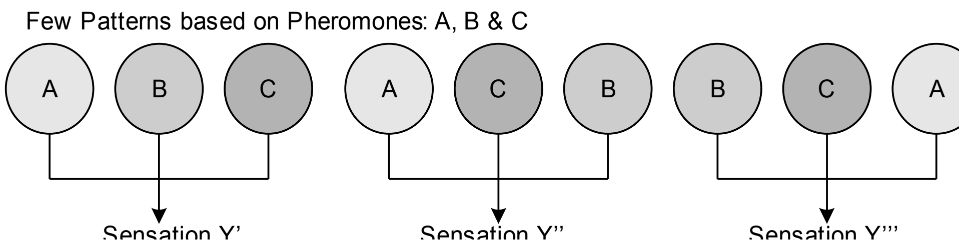


Figure 3.8: Sensory patterns created from several pheromones. Three chemical types grouped into clusters of three.

According to Bert Holldobler *et al* [HOLL90], most alarm signals are multi-component, typically consisting of two or more pheromones, which often serve simultaneously to alert, attract, and evoke aggression. The situation explained in figure 3.8 is called as multimodal systems [HOLL90], which transmit signals through more than one sensory modality. Multimodal systems in some insect groups may even use acoustical signals to complement the effect of pheromones [HOLL90].

3.3.4 Heuristic 4: High rate of interactions would lead to more refined and precise patterns of behaviour

It is identified that ants use "rules of thumb" [HOLL90] or in other terms heuristics to select the appropriate action based on the local stimuli. An action selected by a heuristic has a probability of correctness attached to it or is limited in precision. How does the coordinated and emergent outcome of probabilistic actions with limited precision give way to such complex and precise behaviour of building nests, taking care of the brood, foraging and invasion of territory? According to Bert Holldobler *et al* and Deborah Gordan [HOLL90] [GORD99], it seems to be the result of combining heuristics with the high rate of interaction among individuals due to egalitarian nature of insect colonies. When aligning this theory to the coordination model of AAANTS, it could be assumed that low precision AAs when collectively coordinated and refined through high rate of rejections and acceptance results in complex and precise patterns of emergent behaviour.

The BCs are constructed based on the reinforcements from the environment. The concept of BCs is the key to achieve behavioural congruence. When the environmental reinforcements change, so should the arrangement of the concentres. This may be analogous to the fact that more refined you get, the more you train yourself. Though, AAs are less precise, a bundle of actions reinforced frequently would lead to refined behaviour. Further, the high rate of interactions has an effect on the congruence of behaviour. The BCs would get reshuffled to a new arrangement as the environmental situations change.

3.3.5 Heuristic 5: The recent interactions of successful agents could be used to evaluate the most needed role to the community.

A deterministic model was discussed by Deborah Gordan [GORD99] to investigate the task allocation of ants. At a given time, ants in a colony are totally dedicated to executing their most valued task as per the local perceptions. As time progresses, ants either continue the current task or switch over to another task based on the relative success of other perceived tasks with the current preference. The intensity of task switch-over improves as the colony size increases due to accelerated interaction rate among individuals [GORD99]. Consequently, larger colony sizes are more receptive to environment changes, since high interaction rates spread the emphasis of successful tasks across the colony and rapidly adapt to new situations.

It shouldn't be misinterpreted that the interaction of a successful ant would instantly influence an unsuccessful ant. According to Deborah Gordan [GORD99], the signal is neither in the contact nor in the chemical information exchanged, but the signal is in the pattern of contact and especially in the recent history of encounters. Similarly, in the AAANTS coordination model an agent would continue to execute a particular task until the recent history of interaction patterns advise to switch over to another successful task.

The overall behaviour of a colony is spread across castes and roles. A group of ants would belong to a particular caste and would be responsible for a repertoire of roles within the caste. However, within a given period of time an ant would concurrently execute several roles though a few may be more important to the survival of the colony than others. Hence, a group of agents within a given caste would switch roles based on the changing importance to the colony.

3.3.6 Heuristic 6: An agent could become altruistic based on kin relationships

The last aspect to investigate is related to the task distribution within a group based on the kinship among the participants. Kin recognition is an important aspect of any insect colony for collective behaviour. It is identified that each individual in an ant colony possess phenotype[17] matching details of kin such as a recognition label and a sensory template specifying a learned set of data likely to be borne by kin [GORD99]. Here a label is like an identifier assigned to a group of individuals preferably generated from offspring of a single queen. Ants bathe in special chemicals either generated from the queen or from specific individuals of the colony, in order to setup this recognition label across the colony [HOLL90]. This concept could be further enhanced to a model where individuals in a colony are categorised to groups based on roles and identification assigned to each role. Individuals then could maintain multiple identification labels and sensory templates related to each of these groups. This would enable individuals to perform heterogeneous tasks as requirements arise in a colony.

The trigger for any Behavioural Act (BA) is a sensation. The sensation could be generated from colony members as well as from sensory sources in the environment. If an ant could differentiate colony members based on kinship, then another dimension could be added to the behavioural congruence mechanism. As depicted in figure 3.9, an individual may execute different BAs for the same sensory template based on kin differentiation. The kin is uniquely identified by a group identifier and members affect each other by communicating through pheromones. The group identifier is piggybacked on the original message of the pheromone, which the

[17] The collection of genes in an organism is called its genotype. Every organism begins as a single cell containing such a genotype [STAN95]. The organism's phenotype is the collection of properties or attributes it exhibits during development [STAN95].

receiving party intercept and aligns with the most suitable behavioural acts.

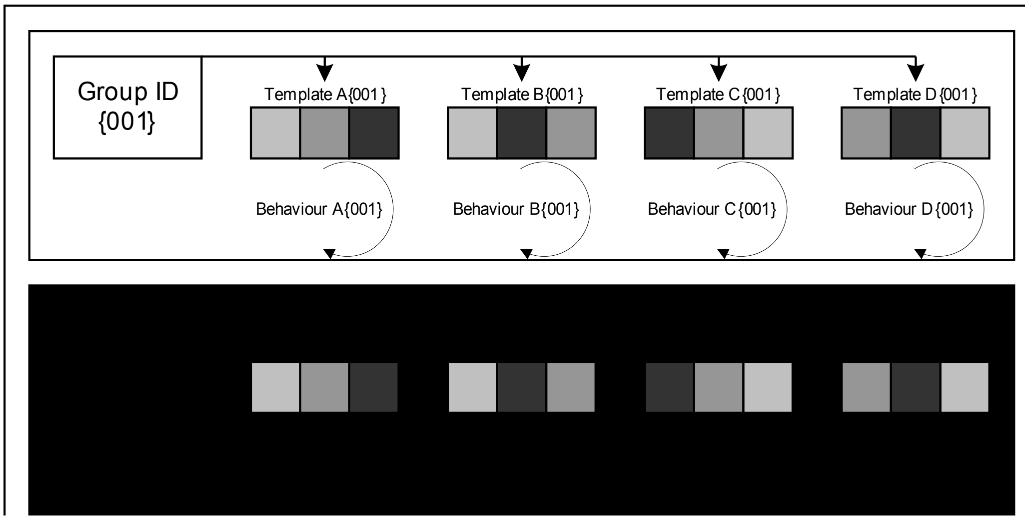


Figure 3.9: Maintaining sensory templates for coordinating activities with Kin.

Emergence would require altruism from participants of a colony of agents. Each agent should be committed to the roles rather than concentrating on a selfish agenda. No sooner the agents deviate from the colony life to concentrate on individual well-being, the whole concept of emergence falls apart. Therefore, the agents are bonded together with the use of kinship towards each other in their respective castes.

The six heuristics discussed above is responsible for emergent behaviour in a typical ant colony. These heuristics were selected as the foundation for emergent behaviour within the AAANTS coordination model.

3.4 Organising Sensations

The sensations in the environment should reach the responsible sensory receptors to enable the perception of that sensation. For example, visible light reflected from an object should reach the human eye to initiate the visual recognition process. The receptor process consists of perception and cognition functions. The perception activity translates the physical sensation to a neural signal and it is the cognitive process that gives meaning to the signal. In another sense, any pattern of cerebral activation in response to an external stimulus could be identified as a representation of the stimulus, which may encompass both activities of perception and cognition [MARA00]. The word "representation" is usually reserved for pattern of activation characteristic of the particular external stimulus or some property of the stimulus [MARA00].

3.4.1 Fuzzy Clustering for Identifying Sensory Patterns

There are several techniques mostly based on supervised learning being used by the research community extensively for the purpose of pattern recognition [RUSS95]. For example, neural networks and Bayesian classifiers are two popular techniques that fall into supervised pattern recognition nature which mostly requires a set of patterns that have already being classified. Due to the unsupervised and reinforced nature of the AAANTS research, methods based on clustering analysis were selected as the most appropriate. Further, the AAANTS model suggests a relationship between frame-based representations and clustering algorithms.

Clustering analysis is used for many aspects of research and applications such as pattern recognition, data mining, image analysis, artificial learning and bioinformatics [RUSS95]. Clustering analysis is a process of organising objects into groups, where the members of a group have something similar based on one or more attributes [KANA03a] [NASC00]. Hence, there should be some level of dissimilarity among objects that belong to different clusters. Therefore, the goal of clustering is to determine the intrinsic grouping in a set of raw data elements.

There are many algorithms for clustering analysis such as K-means, Fuzzy C-Means (FCM), Hierarchical and Gaussian [KANA03a] [NASC00]. The FCM algorithm was selected as the ideal candidate for the clustering capabilities of the AAANTS model because it allows one piece of data to belong to two or more clusters [NASC00] [WANG03]. The segmentation of sensory information into overlapping clusters facilitates the creation of heterogeneous patterns based on dimension of analysis, which is considered important to this research. When using FCM, each piece of data is assigned a value to describe the relatedness to each group where a group is represented by a clustering centre. Further, the research argues a relationship among clustering centres and hubs. A detailed description of the FCM algorithm is found in [WANG03] [NASC00] [ABON02], of which a summary is given below using the formulas 3.1 to 3.6. The objective of the FCM clustering technique is to optimise the objective function (3.1).

$$J(U,V) = \sum_{j=1}^{C}\sum_{i=1}^{N}(U_{ij})^{m} \parallel x_i - v_j \parallel^2 \; .-(3.1)$$

$$U_{ij} \in [0,1], \forall i = 1,..N, \forall j = 1..C. -(3.2)$$

$$\sum_{j=1}^{C} u_{ij} = 1, \forall_i = 1,..N. -(3.3)$$

Let x={x_1, x_2, .., x_n} represent the collection of data elements to be clustered. x would be classified to "c" clusters by minimising the objective function (3.1). U_{ij} is the membership degree of data X_i to a fuzzy cluster set v_j, where v={v_1, v_2, …,v_c} are the cluster centres. U = $(U_{ij})_{N*C}$ is a fuzzy partition matrix, in which each U_{ij} indicates the degree of membership of each data element in the dataset to the cluster j. The $|x_i - v_j|$ is the Euclidean distance between x_i and v_j. The fuzziness index is represented by parameter m, which could be used to control the fuzziness of membership data elements.

Step 1: Initialise the membership matrix with random values while satisfying conditions (3.2) (3.3). The number of clustering centres needs to be decided at this point since there would be similar number of columns in the matrix.

Step 2: Calculate the clustering centre using the following equation. (where v: clustering centres; u: degree of membership; x: data elements; m: any real number greater than 1.)

$$v_j = \frac{\sum_{i=1}^{N} (u_{ij})^m x_i}{\sum_{i=1}^{N} (u_{ij})^m}, \forall j = 1,..., c. -(3.4)$$

Step 3: Calculate the new Euclidean distance (d) between data elements and clustering centres.

(Where d: Euclidean distance; N: number of elements in the dataset; C; number of clustering centres)

$$d_{ij} = \| x_i - v_j \|, \forall i = 1,...., N, \forall j = 1,...., c. -(3.5)$$

Step 4: Update the Fuzzy partition matrix U: If $d_{ij} \neq 0 (\therefore x_i \neq v_j)$.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{d_{ij}}{d_{ik}})^{\frac{2}{m-1}}}. -(3.6)$$

Step 5: Stop when the termination criterion is reached else return back to step 2. The maximum number of iterations, checking against a tolerance value or difference among previous iteration could be used as suitable termination criterion.

The selection of good clustering centres and the correct number of clusters that represent the overall dataset is an important factor for the success of the clustering process. It is described that data could be clustered using an iterative version of the FCM algorithm, but the drawback of the FCM algorithm is that it is very sensitive to cluster centre initialisation because the search is based on a hill climbing heuristic [KANA03b].

Ant Colony Optimisation algorithms [KANA03a] [KANA03b] are one of the most successful methods that could overcome FCM algorithm's sensitivity to the initial values of clustering centres. The clustering centres filtered through Ant Optimisation Algorithms could be refined using the FCM algorithms. However, the AAANTS model did not pursue this path in order to keep the FCM algorithm less complicated to suit the generic pattern recognition needs of this research. However, this could be taken as a method of refining the overall research outcome in the future.

3.4.2 Segmenting Sensations Temporally

A sensory signal would eventually become a continuous potential fluctuation in the nervous system where the raw sensation is based on activation and inhibition of nerve endings. A given mode of sensation is not performed by a single sensory cell, but by myriad of sensory receptors. For example, in mammals the eye has cells that are receptive to light, tongue and nose have cells receptive to chemicals (similar receptors in antennae of insects) and the skin has cells to sense touch.

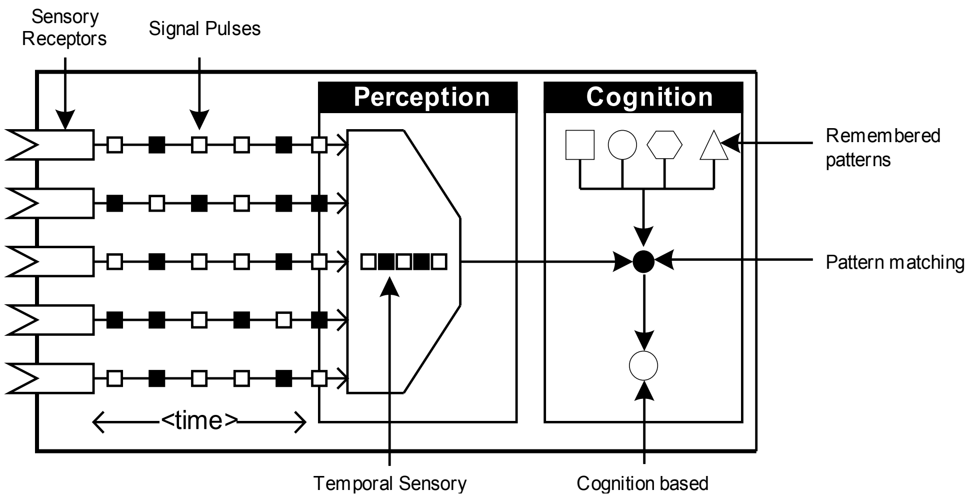


Figure 3.10: Generating sensory patterns using elementary signal pulses

Signals from a collection of receptive cells in a given region of a sensory apparatus generate a rhythmic array of inhibitions targeted towards the cognitive elements. Figure 3.10 shows the steps taken in the AAANTS model to transform sensory signals from the environment to meaningful sensory patterns using perception and cognition processes. The stimulation of the sensory receptors generates a repetitive array of signal pulses that results in formulating a Temporal Sensory Frame (TSF) in the perception layer. The TSF is processed using Clustering Algorithms to generate a cognitive pattern. Different types of cognitive patterns could be generated based on the input values of the TSF. The responsibility of the cognitive process is to perform a pattern matching activity to identify the input pattern with respect to the already remembered patterns. Hence, the cognitive elements should be able to amalgamate the concurrent rhythmic pulses to a known pattern of sensation. If a processed cognitive pattern is non-existent in the repository, it is added to the list of remembered patterns for future references.

The perceptive process requires inputs from a collection of receptors to perceive a meaningful model of the external stimulus. For an example inputs from multiple receptors in the human eye is required to capture a complete picture of an external image. The AAANTS model uses formula 3.7 to calculate the overall perceptive value (called as the Receptor Sensory Grade (RSG)) of a single sensory receptor in a given modality. With respect to the formula 3.7, SI represents the quantification of the sensation felt by the sensory receptor, ST is the minimum strength of a sensation that could trigger a receptor and SG is the average difference between two consecutive granular values.

$$RSG = \frac{(SI) - (ST)}{(SG)} . -(3.7)$$

(Where SI: Sensory Intensity; ST: Sensory Threshold; SG: Sensory Granularity)

The input from a single receptor alone would not be useful to perceive a meaningful outcome from a sensation. Hence, a collection of values from all receptors of a modality would provide sufficient inputs to the perception and cognition processes. This summation of RSGs collected from all receptors of a sensory mode at a point in time is called as a Temporal Sensory Frame (TSF). For example stimulations from all the light sensory cells in a mammalian eye at a point in time would represent a TSF for the visual sensory mode. There would be an abundance of TSF instances created within a given period of time from each mode of sensation. The success of an artificial learning model would depend on building meaningful relationships among RSGs within and across TSFs of inter and intra-modalities of sensations.

A TSF is defined as a 2-Dimensional (2D) array consisting of RSGs of a single modality of sensation. The elementary sensory values are called as "sensolets" for easy reference within this discussion. The sensolets are spatially arranged in a 2D array to produce a TSF. When applying the FCM algorithm, clustering centres are placed in a random manner across the 2D array of a TSF. The number of clustering centres for a frame may be fixed based on the mode of sensation. The degree of membership of each sensolet with all clustering centres is calculated initially (U_{ij}). However, spatial arrangement and the average value of the clustering centres have to be adjusted to overcome initial drawbacks in selecting the clustering centres.

The relatedness of clustering centres would result in a pattern that is unique to a certain instance of a sensation. The formula 3.8 is used to quantify the relationship of the members in one cluster with the rest of the clusters in a given TSF. The RSG formula (3.7) is used to calculate the value of each data element (x_i). The value of RC_j (Relationship Coefficient (RC) of each clustering centre) would give the summation of values based on the relatedness of each element in a given cluster (x_i) with the rest of the clustering centres (C_j) other than the one it belongs to. Hence, RC_j would represent a quantification of relatedness of a given cluster with the rest in a given TSF.

$$RC_j = \sum_{i=1}^{N} U_{ij}.x_i. - (3.8)$$

(Where $x_i \in C_j$; U: degree of membership; x: data elements; N: number of elements in the dataset; C; number of clustering centres)

This formula summarise the attraction of data elements of a certain clustering centre, to the rest of the clustering centres of a single TSF.

3.4.3 The Effect of Hubs

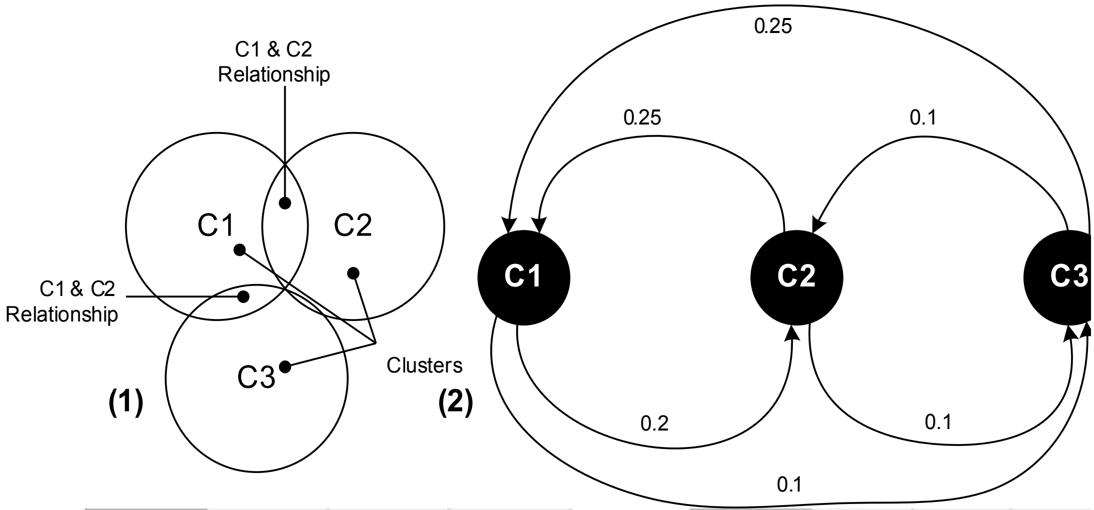
A complex system is usually constituted of many elements that interact with each other and the global behaviour arises from interactions of the constituent elements. The complexity of the system is proportional to the number of elements, the number of element interactions and the complexities of the elements [GARC01] [BABA01]. A complex system is also called emergent because the macro system possess properties that are absent in micro elements. These macro properties emerge from the interactions of the constituents of the system [GARC01]

[BABA01]. However, it could be argued that complexity is relative and it is difficult to define an absolute reference point.

A complex network is composed of nodes, of which, some could be called as Hubs. The concentrated and highly connected nodes of a complex network are called as Hubs [BARA02]. The Hubs seem to prevail in many types of complex networks ranging from the nervous systems of animals to dynamic topology of the Internet [BARA02]. It is identified that random node removal from a network with hubs, initially seems to be unaffected until reaching a critical threshold, after which the network disintegrate into isolated segments [BARA02] [SOLE01].

The neural groups in the nervous system perform a similar function to that of Hubs. It is found that neural groups are collections of 50 to 10000 neurons, relatively close to each other which are formed during development and prior to experience, whose intra-connections allow them to respond to particular patterns of synaptic activity [STAN95]. The neural group arrangement occurs outside the influence of the genetic control processes; hence, no two individual animals are likely to have identical neural connectivity, not even identical twins with identical genetic material [STAN95]. A secondary repertoire forms as a result of this postnatal interaction with the environment via synaptic modifications both within and between neural groups [STAN95].

The objective of using the concept of hubs is to derive a formula that describes the uniqueness of a TSF. This could be achieved in two steps. Firstly, the values of the RCs (formula 3.8) of clustering centres could be used as the measurement for selecting hubs. The high coefficient value indicates the high degree of attraction of member data elements of rest of the clusters. The Figure 3.11 depicts a scenario of a TSF with three clustering centres related to six data elements. Among them, C1 and C2 show high RC values and could be considered as hubs for the discussed data set.



	C_1	C_2	C_3
X_1	0.8	0.1	0.1
X_2	0.6	0.3	0.1
X_3	0.2	0.7	0.1
X_4	0.3	0.6	0.1
X_5	0.3	0.1	0.6
X_6	0.2	0.1	0.7

	C_1	C_2	C_3
C_1		0.2	0.
C_2	0.25		0.
C_3	0.25	0.1	

Figure 3.11: Creating a unique pattern using relationships among hubs

The second step is to calculate the Hub Cohesiveness with respect to the relatedness of selected hubs with each other. A unique pattern called the Hub Cohesiveness Frame (HCF) (figure 3.11 – (4)) which is a matrix that could be derived using this technique. Figure 3.11 (1) shows three clusters with overlapping data elements. It is the overlapping data elements that contribute towards calculating the HCF. The cohesiveness among two clusters is bi-directional (figure 3.11 – (2)). For example, from all data elements (figure 3.11 – (3)), X_1 and X_2 belongs to Centranoid C_1 based on the fuzzy membership values.

The C_x to C_y Cohesiveness Index (CI) is calculated as follows:

$$CI_{xy} = \frac{\sum_{i=1}^{N} U_{iy} X_i}{N}. - (3.9)$$

(Where N is the number of elements that belongs to C_x , u: degree of membership; x: data elements)

However, CI is a bi-directional index. Therefore, the values of CI ($C_1 - C_2$) and CI ($C_2 - C_1$) may be different to each other, which is obvious from data set depicted in figure 3.10 – (3). Similar sensations should represent TSFs with approximately similar pattern values. The HCF data matrix is recorded and compared with the HCF of new sensations to identify relationships. The HCF matrix comparison is done based on value comparisons of related data elements within defined thresholds. The agents should attach these frames for action synchronisation as discussed in section 3.2.2.

The above discussed formulas (3.7, 3.8 and 3.9) was instrumental in identifying less complicated sensations (e.g. direction and angle) used in the grid-world foraging and robotic arm movement experiments discussed in chapter 6. The experiment results confirm that the same sensation processed iteratively produces values for CI (formula 3.9) within a tolerable range. However, the experimented sensory types were unimodal and less complicated in nature. The application of these formulas to more complex (e.g. visual sensations) and multimodal sensory pattern recognition may require further enhancements.

3.4.4 Relationship across several Temporal Uni-Modal Frames

The word "situation" is descried in Webster's dictionary as "the way in which something is placed in relation to its surroundings" [WEBS88]. A situation could be represented by a series of TSFs in a selected modality. Two vehicles colliding with each other could be called as a situation of a vehicle accident. A visual sensory frame of this situation that shows two vehicles very close to each other would not give the intuition of the situation in its entirety. Only the sequence of frames that lead to the accident would give such an intuition to the situation.

Time/Indexes	C1-C2	C2-C1	C1-C3	C3-C1	C2-C3	C3-C2
T1	0.2	0.25	0.1	0.25	0.1	0.1
T2	0.1	0.3	0	0.35	0.125	0.15
T3	0.15	0.35	0.1	0.2	0.2	0.2
T4	0.2	0.3	0.125	0.25	0.25	0.25
T5	0.25	0.25	0.2	0.35	0.2	0.35
T6	0.3	0.2	0.15	0.4	0.12	0.3
T7	0.275	0.15	0.125	0.35	0.1	0.32
T8	0.4	0.2	0.2	0.3	0.23	0.33
T9	0.35	0.3	0.25	0.3	0.3	0.4

Table 3.1: Cohesive Indexes spread temporally within a situation in a single modality

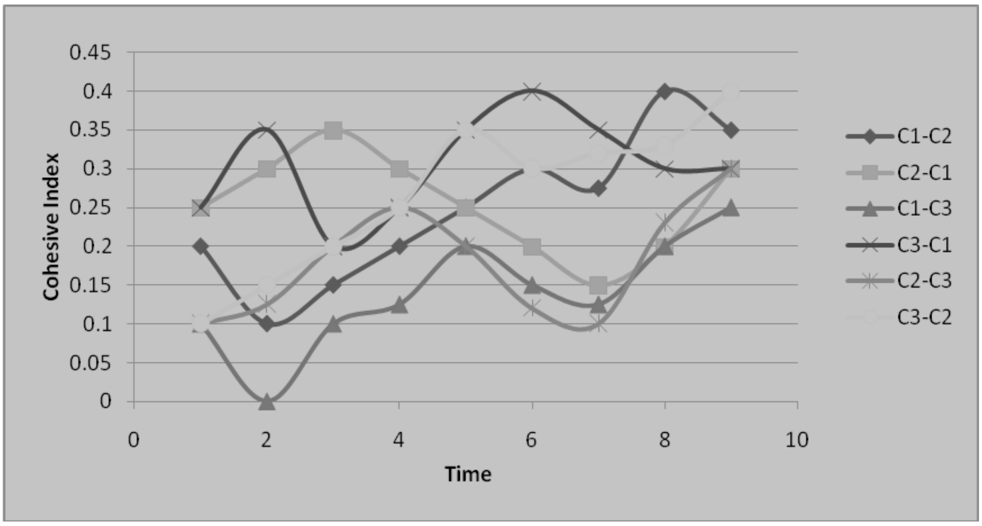


Figure 3.12: Line graph of the spread of Cohesive Indexes temporally

The dataset depicted in Table 3.1 and Figure 3.12, shows the changing values of Cohesive Indexes of a situation related to the neighbouring state values of a moving agent in the Grid World experiment. This dataset would be more meaningful when represented as the difference of indexes across subsequent time intervals as depicted in Table 3.2 and Figure 3.13. These differences fall into a pattern which would be unique to a given situation. The pattern could be identified by using clustering algorithms described earlier.

Time/Clusters	C1-C2	C2-C1	C1-C3	C3-C1	C2-C3	C3-C2
T1-T2	0.1	-0.05	0.1	-0.1	-0.025	-0.05
T2-T3	-0.05	-0.05	-0.1	0.15	-0.075	-0.05
T3-T4	-0.05	0.05	-0.025	-0.05	-0.05	-0.05
T4-T5	-0.05	0.05	-0.075	-0.1	0.05	-0.1
T5-T6	-0.05	0.05	0.05	-0.05	0.08	0.05
T6-T7	0.025	0.05	0.025	0.05	0.02	-0.02
T7-T8	-0.125	-0.05	-0.075	0.05	-0.13	-0.01
T8-T9	0.05	-0.1	-0.05	0	-0.07	-0.07

Table 3.2: Cohesive Index differences across subsequence time slots within a situation

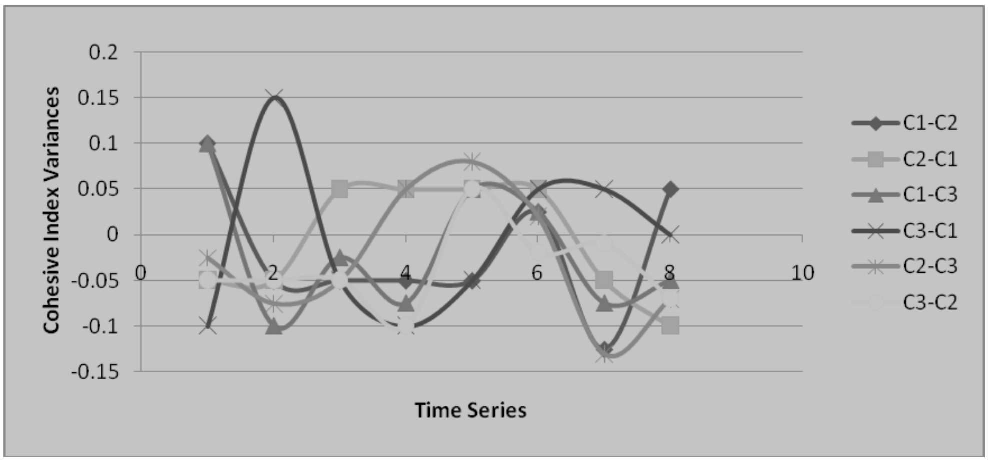


Figure 3.13: Line graph of the spread of Cohesive Index variances temporally

In relation to the AAANTS model, an agent needs to practically keep the history of sensory patterns to identify a situation. In the real-time environment an agent maintains a queue of TSFs where the new TSF is added to the top of the queue and obsolete frames are taken from the tail of the queue. At each temporal gap, the agent needs to perform a Situational Analysis of the cached frames against the learnt patterns of known situations. If a situation is identified based on the reinforcements received during the past, the agent would execute the necessary action for that situation.

3.4.5 Relationship among Multi-Modal Frames

The Cohesiveness Index takes into consideration the sensations in relation to a single mode of sensation such as smell, touch, light etc. However, a given situation would produce sensations in multiple modalities that would ideally produce several Cohesiveness Indices. Figure 3.14 shows that sensory frames are collected from several modes by the sensory receptors and forwarded as an amalgamated frame to the cognitive elements. The pattern matching engine in the cognitive tier should implement an appropriate algorithm to match against remembered patterns.

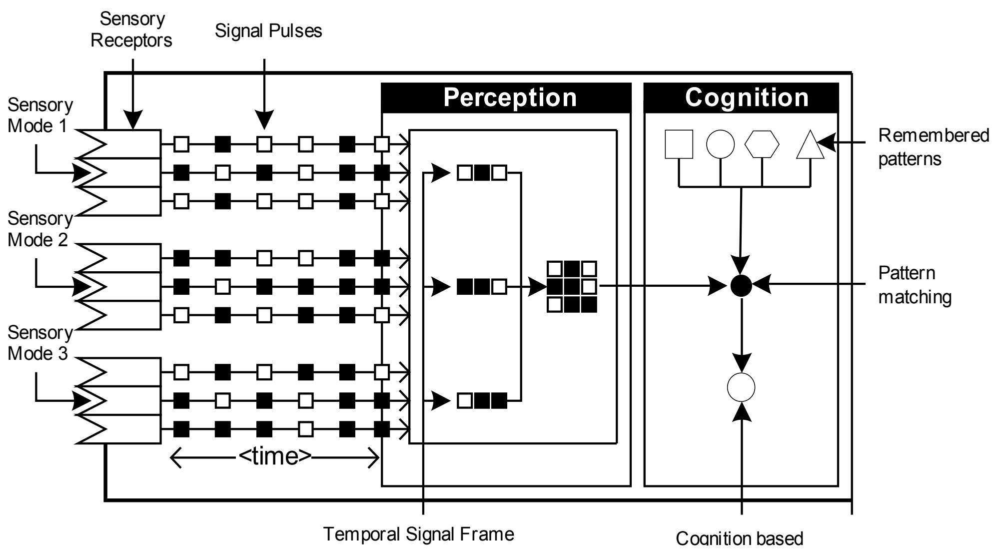


Figure 3.14: Sensory patterns using elementary signal pulses in a multi-modal scenario

This aspect of building relationships among multi-modal sensory frames is not pursued in this dissertation. This is an aspect that could be researched further to gain insight to integration of multi-modal sensations. This is similar to the concept of Perceptual Integration [COEN00], where perception layers from multiple modalities are integrated into a holistic abstraction. The McGurk effect [COEN00] is perhaps the most convincing demonstration of the inter-sensory integration where one modality radically changes perceptions in another through perceptual integration. Post-perceptual and multi-modal integration are two popular techniques for perceptual integration [COEN00]. Post-perceptual integration occurs in systems where the modalities are treated as separately processed, increasingly abstracted pipelines and the outputs of these pipelines are integrated in a final integrative step where as in multi-modal integration perceptual events are separated from the specific sensory mechanisms that generate them and integrated into a higher-level representation.

3.5 Chapter Summary

The objective of this chapter is to introduce the coordination model that achieves behavioural congruence as a result of altruistic interactions amount a group of agents. The AAANTS coordination model consists of aspects related to structure/composition, procedures and heuristics. All these aspects were described in terms of innate and adaptive characteristics. Structurally, agent life-cycle and AAs were identified as innate. However, ATs could be categorised as hybrid in nature where an initial collection is represented inherently and the rest formulated using a stochastic method. Behavioural Concentres are special hubs that are structurally placed based on the reinforcements from the environment and considered adaptive in nature.

Six heuristics were introduced with the inspiration from the insect world with the intension of formulating behaviour as a result of emergence. The implementation of these heuristics would be partly done by individual agents and the rest by the AAANTS framework.

Procedures relating to behaviour were mostly related to recognising sensory pattern from sensory modalities. The latter sections clearly described the methods of identifying sensory patterns using clustering algorithms and further identifying sensory frames uniquely using hubs. Also the relationships among sensory frames were established to identify situations.

Chapter 4

Reinforced Group Adaptation

4.1 Introduction

The adaptations to the environmental changes could happen either as improvements in knowledge or as genetical adaptations within a species. The less complex animals such as insects tend to react to the environment mainly based on genetically implanted information. The genetics tend to implant innate capabilities to map appropriate behaviours to environmental perceptions. Even higher level animals such as humans tend to posses these innate capabilities primarily as reflex behaviours. In fact, the animals with shorter life-span such as some species of insects tend to thrive more on genetical information for their behaviour due to the lack of reasonable time to learn either by supervision or reinforcement. Therefore, genetical mutation is a very useful mechanism for the survival of many species where time-consuming learning becomes a limitation. However, there is one simple lesson to be learnt throughout history; species that failed to adapt (either genetically or by active learning) to the environment have failed to survive and were eliminated from the landscape.

According to Marvin Minsky [MINS86], the development of the human mind from infancy to adulthood is achieved in terms of stages where each stage acts as a teacher to the next stage by providing guidance and assistance. MAXQ method [DIET00] provides a similar strategy to that of AAANTS where it decomposes a reinforcement learning problem into a set of sub-problems. However, MAXQ differs from AAANTS due to the use of both hierarchical policy and reinforcement learning in a top-down arrangement as opposed to a bottom-up emergent arrangement. The stage wise development is further confirmed by the arguments of Vijaykumar Gullapalli [GULL92], where a high level complex task could be decomposed into a sequence of lower-level tasks and thereafter, the same activity performed on the sub-tasks until atomic functions are reached. HAM (Hierarchical Abstract Machines) is another analogy where non-deterministic finite state machines are organised in a hierarchy and higher level abstractions invoke lower level machines [PARR97].

The preparatory experiments of this research confirmed that instructive feedback used in isolation produced less optimal results in comparison to the AAANTS model. The experimentation of a rule-based instructive feedback method in comparison with the AAANTS model is depicted in Figure 6.19 in chapter 6, section 6.5. In real-world problems, reinforcement learning algorithms using delayed reinforcements converge too slowly to the optimal solution [PIER94]. Further, it is accepted that most supervisory methods are time consuming and domain specific [PIER94]. Consequently, there was initial inspiration to search for a technique that could blend these two approaches.

When analysing the learnt behaviour in the natural world, it became clear that some aspects are learnt, and the rest innate. The innate behaviour could be pre-coded knowledge such as a good exploration strategy or an initialisation of action-values [PIER94]. It was initially identified that all animals are born with a repertoire of pre-coded innate capabilities and the rest is learnt during the lifetime of existence, either through supervision and or rewards from the environment. This understanding has set the foundation for the adaptation model of AAANTS.

The rest of the chapter focuses on the adaptation methodology of the AAANTS model. The adaptations are discussed in two broad strategies. The initial discussion relates to the use of reinforcement learning as perturbations to the internal knowledge representation. This is the main mechanism that makes sure that, the participants of an agent colony adjusts to each other, for achieving behavioural congruence. The latter part of this chapter focuses on the continuation of existing knowledge to the future generations of the colony. The AAANTS model is inspired by the periodic regeneration of new agent instances by combining genetical material of successful individuals of the colony. The learnt knowledge of successful individuals is mixed to generate new offspring and unfit individuals are eliminated from the colony.

4.2 The Learning Architecture

The architecture of a system should encompass the building components and their interconnectivity [GAMM95]. The objective of architecture is to facilitate the functionality expected out of an entity. The purpose of learning architecture is to facilitate the survival within a particular type of environment. Hence, the demands of the environment and the purpose of the species dictate the overall learning structure of an entity.

4.2.1 The AAANTS Learning Architecture

The overall architecture of the AAANTS model consists of myriad of agents that harmonise to achieve behavioural congruence. All agents are homogeneous in structure but may vary in behaviour and capabilities based on the state model and experience. The generic learning architecture of an agent is depicted in Figure 4.1 [RANA05].

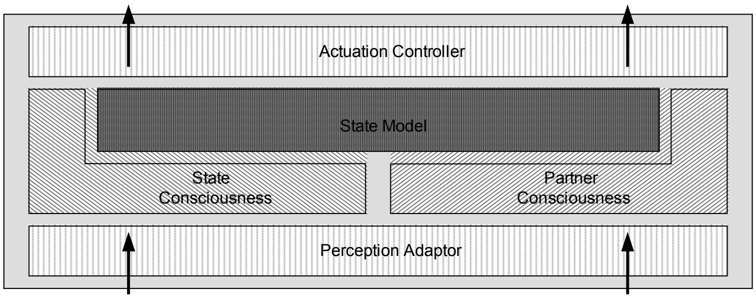


Figure 4.1: Learning Architecture of an agent within the AAANTS framework

Figure 4.1 shows a generic agent interface to the outer world with the use of the architectural components called as Perception Adaptor and Actuation Controller. The Actuation Controller translates the internal need of an action to an external command acceptable by the actuators. The actuators are detached from the agents due to the fact that a community could share the services of a single actuator. Similarly, the sensory receptors are also detached from agent instances which publish a continuous array of signals to be shared within a community. These signals that relates to the respective modality of interest by the agents are intercepted and filtered by the Perception Adaptation layer.

The State Model maintains a summary of states that is of interest to the agent which would always represent a subset of global states. It maintains a repository of information about state-action values, rules, hub states and optimal / sub-optimal paths. The building elements of the State Model are based on frame based knowledge representation. The frames contain information related to the sensations [RANA03c]. A frame or group of frames may represent a state in the environment where a state is uniquely identified by the amalgamated sensations from all sensory modalities [RANA03c]. Therefore, at a given instance the inputs from all sensory modalities would, map to one or many frame collections with a calculated probability of relatedness. The frame instance with the highest probability of relatedness would be taken as the match.

The consciousness modules (State and Partner) keep references to the on-going sensations as well as to the peers in the colony within the locality of a homogeneous group. It is with the use of the consciousness elements that an agent could take actions that are coordinated with others. The state consciousness module is responsible for keeping track of the current state of the agent. It keeps real-time pointers to the frame instances found within the state model. The state model keeps the frame instances organised into a connected mesh. The consciousness indicates the progress made within a given period of time which obviously relates to a definite past and an uncertain future. The past is a simple aggregation of the sequence of states visited executing respective actions while receiving rewards from the environment. The past is very important especially when considering reinforcement learning methodology where the past rewards are used to predict the most desirable future. The most desirable future relates to the path with the optimum reward based on the past reinforcements.

4.2.2 Relating Knowledge and Learning

Learning and intelligence are intimately related to each other. It is usually agreed that a system capable of learning deserves to be called intelligent; and conversely, a system being considered as intelligent is, among other things, usually expected to be able to learn [WEIS00].

A goal could be achieved with a properly coordinated sequence of actions by a community of agents through inter-agent communication. A collection of actions that are highly reinforced by the environment that generates the most appropriate solution to a specific problem situation could be called as properly coordinated. During the initial agent interactions with the environment, each agent tends to maintain a flat structure of state sequences. However, as the agents iterate through the environmental states with the objective of finding the global optimum may result in a collection of states amalgamated through Hubs arranged into a layered hierarchy. For example in Figure 4.2, the states represented by L1S1, L1S2, L1S3 ending with sub-state L1S4 in layer 1, could be represented by a single hub state (L2S1) in layer 2 [RANA05]. The selection of the hub state would result in automatic promotion of the sub-states as the next level of reachable states.

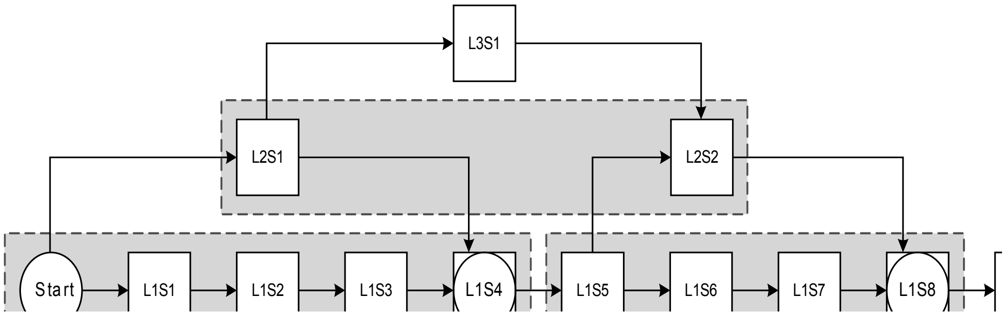


Figure 4.2: Hierarchical layers based on hubs and states

A higher level state represents a link between two very important lower level states – the Hub states. These hub states are highly connected states from which the system could reach many other critical states. The states in between hubs are more decided in terms of the sequence of execution whereas the hub states open up a list of options. This is to some extent similar to Nearest Sequence Memory (NSM) [GARD98] where raw experiences are recorded as a linear chain and the choice of the next action is evaluated based on the nearest neighbours in the experience chain.

It was found that organising past experiences hierarchically scales better to problems with long decision sequences than, organising past experiences as a linear chain of primitive observations and actions [GARD98]. The hierarchy of states within the state model of AAANTS agents are not pre-defined, but dynamically expanded and collapsed with the iterative adaptations to the changing environment.

Marvin Minsky [MINS86] describes a theory in relation to the mind about agents in layers of societies which is analogous to the hierarchical layers of hub states described above. In this theory, knowledge is represented in memory with a type of agents called the "Knowledge-line" or shortly as "K-line" agents (discussed in Section 2.4). The K-lines are organised into layers. Each new layer begins as a set of K-lines, which starts by learning to exploit whatever skills that have been acquired by the previous layer. Whenever a layer acquires some useful and substantial skills it tends to stop learning and changing, and then yet another new layer could begin to learn to exploit the capabilities of the previous layer.

4.3 Exploration and Exploitation Strategies

An action transfers an agent from one state to another. In a given state there could be many actions that transfer an agent to many other different states. Normally, an agent may tend to take the action with the highest expected reward as per the previous experience. If an agent adopts this type of strategy more frequently, it could be called as greedy and non-exploratory. It is identified that greedy actions usually contribute towards local optima [SUTT98a]. Therefore, an agent should exploit what it already knows in order to obtain rewards, but it also has to explore in order to make better action selections in the future.

The dilemma is that neither exploitation nor exploration could be pursued exclusively without failing at the task [SUTT98a]. According to Leslie Kaelbling [KAEL96], formally justified approaches to the problem of finding the optimum blend of using exploration and exploitation are absent. For example, when a group of agents are released to a grid world for foraging, a heuristic could be developed to initially encourage exploration and thereafter converge towards an exploitation policy. However, this strategy would not succeed in an environment where the environmental dynamics change. The new environmental dynamics would require periodic exploration to optimise the policy in relation to the changes.

Exploration could be done using methods such as Boltzmann distribution (actions selected randomly), pseudo-stochastic choice (best action or random action chosen) and pseudo-exhaustive choice (best action or least recently chosen) [PIER94]. The experiments of Pierguido Caironi *et al* [PIER94] conclude that Boltzmann distribution produces worst results in terms of steps to converge to the optimal solution. Further, it is confirmed that on a stochastic task, each action must be tried many times to reliably estimate its expected reward [SUTT98a].

A robotic experiment by Poj Tangamchit *et al* [TANG02], confirms that exploration could be made more efficient by dividing the problem area into sub-areas and having the robots disperse to explore these sub-areas, which would consequently induce cooperation. Therefore, exploration among spatially distributed collection of agents when converged would provide improvements to the global optimum with a reduced effort. Further, a scenario of an agent community partitioned into neighbourhoods is discussed in [SCHA95]. The form of communication considered is based on the idea that the efficiency estimators of agents within a neighbourhood would be shared among them when a decision is made. It was also identified that Nash Equilibrium [NASH50] could be applied to restrict unilateral deviation of agents and would cause each agent's choice to be in synergy with all others [HU98].

The additive and subtractive mechanisms during brain development could be used to explain exploratory and exploitary mechanisms [ELMA99]. This mechanism includes growth of new connections among brain cells (synaptic sprouting / synaptogenesis) and elimination of normal connections. The former is an additive process and the latter is a subtractive process. The additive mechanism is similar to exploration where new avenues are added to the repertoire and subtractive mechanism happens when proper reinforcements are not given to respective state transitions. The additive and subtractive mechanisms are identified to support early learning of infants and further the presence of noise was identified as an advantage due to protecting from falling into local optima [ELMA99].

The action selection strategy for exploitation and exploration are quite different. When in an exploration mode, the next action is selected purely based on a stochastic decision. However, the exploitation methodology is more complex. The selection of the highest rewarded action to transit from one state to another would be a trivial task, but this would result in sub-optimal solutions that converge towards local optima. The following exploitation strategies were adopted by the AAANTS model to reach global optima [RANA05].

4.3.1 Exploitation Strategy 1 – Preference to Highest Rewarded Actions

This strategy relates to selecting actions within a defined action template based on the rewards accumulated during past experiences. Where there are several options for selecting actions within an action template, the agents would simply try to select the actions that would result in the highest probable rewards. Further, it is important to state that the selection is applied mainly to one-step look-ahead situations but could be modified to accommodate multi-step situations too.

It should be mentioned that the AAANTS problem solving is defined as a co-operative activity among a group of agents as opposed to direct competition (discussed in chapter 3). Further, agents are assigned to distinctive AAs within an AT where there could only be one-to-many relationship among agents and AAs respectively. Therefore, one AA is not assigned to many agents which eliminate agents competing with each other within the innate layer of the AAANTS model. The selection of exploration and exploitation activities happens through a stochastic process, where the actions selection algorithm reasonably promotes both activities. Hence, based on these reasons, it could be confirmed that the AAANTS model does not become greedy and suffocate a segment of the agent colony. This aspect is further confirmed by the experiments discussed in chapter 6.

Each action $(a_i \in A)$ in state space (S) and Context (C).

$E_c(a_i)$: Accumulated reward for a type of action in a given context,

$V_c(a_{ix})$: Reward for a specific instance of action instance of a given action type i;

$$E_c(a_i) = \sum_{x=0}^{n} V_c a_{ix}. -(4.1)$$

The $[\max E_c(a_i)]$ is the reward for the most globally preferred action. Therefore, $[\max E_c(a_i)]$ could be regarded as the most preferred action within context C.

The above situation becomes complex when a community of agents act concurrently and compete with each other for a defined state transition. For example, at a given moment there could be several agents that may decide to transfer to the same state based on past experience. The coordination model adheres to the following well-defined methodology to overcome this issue.

The process starts after all the agents broadcast the forecasted reward of the next transition state within the community. It should be first understood that the whole community of agents work under the control of a single timer. The agents perform the transition at the edge of a time slot, which ensures the temporal coordination across the whole community and further the inter-agent communication only affects the individuals within a given locality (discussed in section 3.3.1). The agents in the community are totally autonomous from each other in performing their activities. The discretization of their life-cycle into time slots is merely for the purpose of temporal synchronisation of actions of multiple agents and to resolve conflicts in solving critical sections of the state space. The preliminary experiments done in relation to the grid-world foraging presented situations where multiple agents overlapped on a single state. The agents get a chance to compare the most suitable candidate to proceed to a conflicting state using the time-slot method.

As mentioned above a broadcast signal is effective within a specific time slot. The agents that compete for a given state would compare their own reward values against others and allow the agent with the highest value to proceed. Hence, the agent with the highest reward value would effectively perform the transition. However, this could happen only when agents within a community is of benevolent disposition and further the starvation of under-performing agents is controlled through stochastic use of exploration and exploitation by dominating agents.

After getting clearance for the state transition, the winner performs a lock on the required state, transit to the new state and releases the lock of the old state. The information related to locks of resources is distributed within the community of agents. The agents should only retain state information within a restricted locality. When the most preferred state is occupied by another for a reasonable amount of time, the agent should not waste time with the expectation of reaching that state. A good heuristic is for the agent to select the next best state. This is another approach to escape from starvation. Further, if two or more agents share the highest position, the first agent to lock the resource would be appointed as the winner.

4.3.2 Exploitation Strategy 2 – Preference for Hubs

Unlike Strategy 1, this strategy would focus on multi-step lookup situations. The action taken to reach the immediate Hub state is a good heuristic for exploitation. If a hub state is sensed within the limited locality of sensations, the agents might consider planning to take multiple steps to reach this much desired state. Since Hub states are major decision points for global success, the respective actions to reach them could be assumed as highly probable actions to reach the ultimate goal.

4.3.3 Exploitation Strategy 3 – Similarity based on Behaviour and State

Although two states are very unlikely to be exactly the same, some states could demonstrate a degree of similarity based on a selected set of attributes and behaviour. The AAANTS learning model employs several methods to evaluate the similarity of states. The outcomes of these methods are used to decide on a state transition based on the preference to another similar state, as described in continuation.

Method 1: Similarity based on action patterns

When a definite sequence of actions are executed iteratively starting from a known state, the terminating states could be considered at least closely similar to each other. For example, generically a simple list of actions executed sequentially could be used to open a door. Hence, it could be concluded that execution of door open behaviour would always end up in a general state – inside/outside the premises.

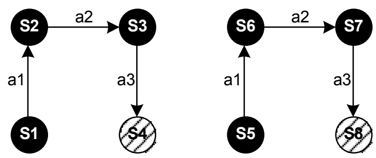


Figure 4.3: State similarity based on action patterns

The "door open" behaviour could be discussed in relation to the actions depicted in figure 4.3 as follows.

1. State S1/S5 is related to the situation before opening the door where the person may walk and stand in front of a closed door.

2. Action a1 is related to unlocking the door which moves the current state from S1/S5 to S2/S6 which is a situation where the person stands in front of a closed but unlocked door.

3. Action a2 is related to pushing the door open which moves the current state from S2/S6 to S3/S7 which is a situation where the person stands in front of an opened door but on the out side.

4. Action a3 is related to the person moving inside the premises which was earlier restricted by the closed door, which would move the current state from S3/S7 to S4/S8.

With reference Figure 4.3, states S4 and S8 could be called as similar due to the execution of an identical action pattern $\{a_1, a_2,$

a_3} in reaching the final state, provided the initial states S1 and S5 also have higher degree of similarity. This type of pattern recognition is a trivial process when considering search space of a single agent. However, it becomes complicated when considering coordinated concurrent actions of a group of agents. The AAANTS model solves this problem with the use of ATs where the AAs are coordinated by multiple agents in a prescribed pattern (discussed in detail in section 3.2.2).

Method 2: Similarity based on feature patterns

States could be similar based on the attributes inherent to a state (e.g. temperature, pressure, radiation levels, light intensity, etc) or based on the spatial arrangement of objects with respect to the neighbouring states and entities. Hence, the neighbouring states could be used to identify spatial and attribute based patterns within an environment.

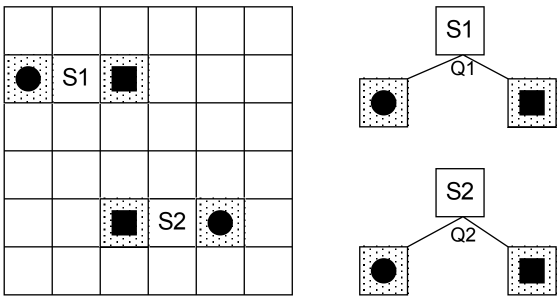


Figure 4.4: State similarity due to spatial and attribute patterns of neighbouring states

With reference to Figure 4.4, the spatial arrangement of neighbouring states of S1 and S2 based on the angle of separation could be used to assess the similarity of states ($Q1 \approx Q2$). The angle of separation of this grid world experiment was taken as the angle created on the two dimensional plane among the cells in concern and the other two neighbouring states. For example, with reference to Figure 4.4, the angle of separation of the states S1 (Q1) and S2 (Q2) with reference to the two neighbouring obstacles (which are identical as a pair) is 180 degrees. Hence, states S1 and S2 could be considered to demonstrate some level of similarity.

Method 3: State values based on rewards

Another way to evaluate the similarity of states is with the use of reward values received during the past experiences with the environment. The state values are numerical representations of reinforcements received during past experiences within a given domain. The states could be grouped into homogeneous categories with the use of value bands. The assumption is that the past experiences of all the agents are captured in a shared context; therefore experiences of individual agents could be shared amongst a community.

The exploitation strategies are applied and selected by an agent in the above listed sequence of importance starting from strategy 1 to 3 where strategy 1 being the most preferred. The agents when deciding on the next most preferred state of transition would evaluate the forecasted future rewards from each of the above exploitation strategies. The calculated values for each available state for all 3 strategies are discounted based on the level of preference. For example values representing strategy 1 to 3 should be multiplied by 1.0, 0.75 and 0.5 respectively. Thereafter, the strategy with the highest outcome is selected as the most suitable path for the next state transition.

4.4 Coordinated Rewards and Learning Patterns through Hubs

As discussed in Section 3.4.3, Hub is a special state that is recognised as important when compared to its neighbours. A Hub could be created due to reasons such as high state values resulted from episodic reinforcements, accumulation of special interim rewards before reaching the final goal, local optima, and lastly and most importantly, states that are used by agents to coordinate dependant actions [RANA05].

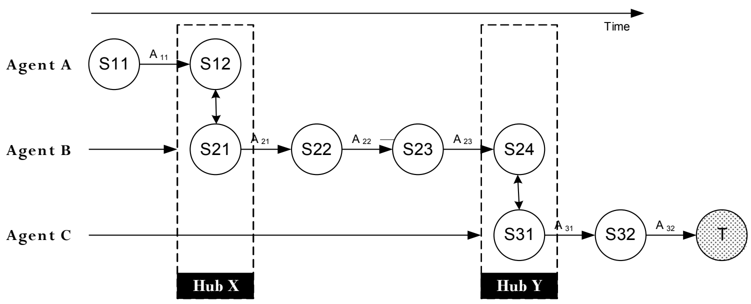


Figure 4.5: Use of Hub states for agent coordination

Figure 4.5 depicts the use of hub states to coordinate agent behaviour. It depicts a situation where three agents A, B, and C perform a series of actions individually but in a coordinated manner. These actions may belong to one or more action templates coordinated by the listed agents. Continuous experience in the environment has enabled these agents to learn that state pairs S12 and S21 together with S24 and S31 synchronise the elementary actions of the three agents to perform a coordinated task.

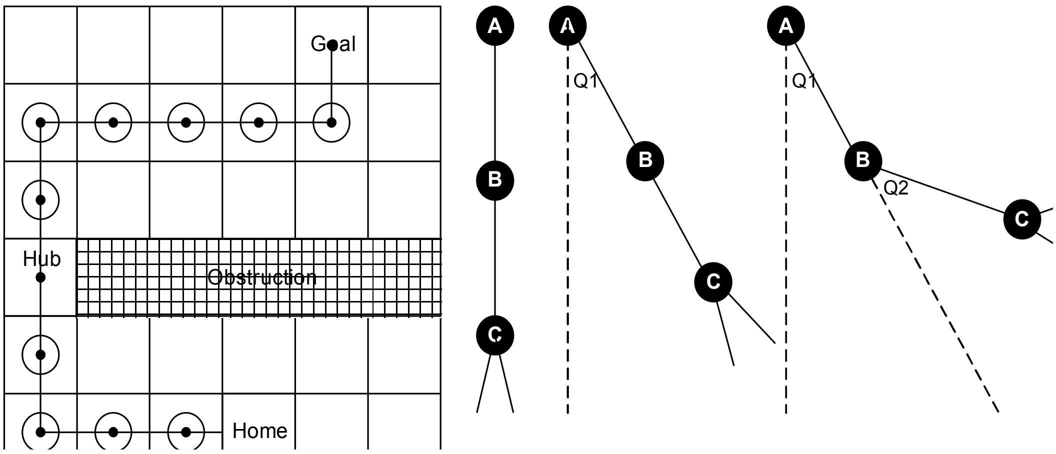


Figure 4.6: Use of hub states for coordination in heterogeneous environments

Figure 4.6 depicts two examples of the use of Hubs by agents in heterogeneous domains. In the grid world, the agent moves from home to goal state through a natural obstruction while receiving a reward from the environment. The successive iterations through the maze would ideally accumulate a high reinforcement value for the state that connects the two obstructed areas. This relatively higher state value would classify this state as a Hub in respect to the rest of the states in the maze [RANA05].

When considering the robotic arm scenario in Figure 4.6, reference angles among the joints A, B and C, are used as the criteria for coordination. In this example the vertical axis is taken as the reference point for measuring the angular differences. The angular difference is the key attribute of a state. The robotic arm achieves different tasks by learning from past experiences and investing in the hub states for each type of activity. For example, if the angles Q1 and Q2 are learnt as required to hold an object, agent A and B would coordinate at the edge of reaching the hub state angular values of Q1 and Q2 respectively.

4.5 The Credit Assignment and Distribution

One of the most important functions of reinforcement learning is credit assignment. The function of credit assignment should be done by an entity in the environment. This entity which could be either an automated or manual process decides the success of the behaviour and assigns a quantified reward value to the participants of that behaviour.

4.5.1 Introduction to Credit Assignment

A composite action is made up of elementary actions that must be performed together by a group of agents while concurrent actions are those that could be performed in parallel by different agents, without the need for synchronisation [GRIF99]. Assigning credit to composite and concurrent actions is one of the most important issues in reinforcement learning [MATA94b].

A complication arises when formulating strategies in dividing a single reward value among all contributing actions [GULL92]. The problem to solve is whether to divide them equally or based on the importance of contribution. Obviously, equal distribution is trivial as opposed to contribution based distribution. In the case of group-level learning the complexity of the problem is further increased, because it requires determining which set of actions performed by the different agents in the sequence deserves credit or blame for the final outcome [DROG98].

It should be understood that the credit assignment function only generates the overall reward value and it is the responsibility of the agent system to decide on the distribution strategy. The distribution logic itself could be centralised or distributed. In most agent systems a centralised entity decides the distribution whereas in distributed systems each agent is responsible for taking a slice of the total, based on its own evaluations. The Appendix D elucidates the credit assignment and distribution techniques adapted by other agent systems. The subsequent section discusses the technique used by the AAANTS learning model.

4.5.2 Credit Assignment and Distribution Methodology of AAANTS

It was stated in section 3.2, that a group of actions executed in a unique sequence and context would belong to an Action Template. An action pattern of a given AT instance should get refined over a period of time based on the reinforcements from the environment. Thus a methodology to properly evaluate and distribute credit among participating agents of an AT is an important aspect discussed hitherto. A full description of the constituents and operations of an AT is found in section 3.2.2.1. It would be useful to reiterate that each started AA instantiates a timer that measures the temporal progress of that atomic activity. These timers are represented by symbols T1, T2 and T3 in Figure 4.7.

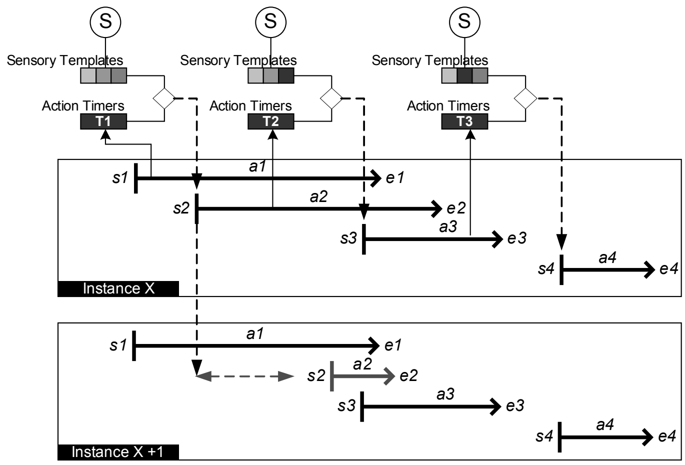


Figure 4.7: Isolating and assigning rewards to the key contributors within an AT

A new action pattern may differ from a prior action pattern due to changes in the action execution sequence as depicted in Figure 4.7. The instances X and X+1 represent two evolutionary stages of a single AT instance. The stage X+1 differ from the original due to the change in the temporal execution start in action a2. This would have resulted due to preference for exploration which the respective agent may have decided to deviate from the original pattern and explore a different combination of execution sequence.

The reinforcement from the environment is related to a quantitative integer value produced after completion of the behaviour. The new action sequence shown in Figure 4.7 (Instance X+1) should receive relatively higher or lower accumulation of rewards over a period of time in comparison to the previously preferred action pattern represented as Instance X. If relatively higher value of reinforcement is given to instance X+1, then that AT instance would have higher probability of getting selected in the future. However, proportionate and disproportionate credit distribution options are available for selection. The experiments in chapter 6 would prove that disproportionate distribution of rewards as a result of assigning larger proportions to relatively differentiated actions yields better global optimum.

In relation to the rewards distribution strategy, the actions within an AT be represented as $a_1 ..... a_n$ where n is the number of total action instances and n~ represents the subset of actions that deviated from the original plan.

Let R_x be the quantitative reward received for this action pattern previously and let R_{x+1} be the reward received for the recently executed action pattern. The differentiation of R_x and R_{x+1} is due to the contribution of n~ actions. The Ratio of Improvement (RI) could be defined using formula (4.2).

$$RI = \frac{R_{x+1} - R_x}{R_x}. -(4.2)$$

The strategy is to reserve a special portion of reward to be assigned to deviating actions and the rest to be equally assigned among all participants. The rewards to be given to each participating AA produced by an agent are calculated as a simple linear apportionment based on formula (4.3.). This value is equal to the previous reinforcement recorded by the AT. The credit or blame to be given to deviating agents is calculated with formula (4.4).

$$[R_{x+1} - (R_x * RI)] / n. -(4.3)$$

$$R_x * RI / n \sim . -(4.4)$$

It should be noted that an increase in reward would be additive and the decrease in reward would be a subtractive to the originally inherited value from previous reinforcements. A reciprocal effect on the reward assignment was devised (based on formula 4.3 & 4.4) where an increment in rewards would increase the values of deviating agent actions and a decrement in rewards would increase the rewards of the non-deviating agent actions.

4.6 Reproduction of Successful Behaviour

A colony over a period of time collects agents with optimal behaviours as well as agents trapped in local optima which altogether hinder the overall performance of the community. Therefore, considerable effort was taken while designing the AAANTS model to periodically eliminate unfit agents and to introduce more fitting agents. This procedure is very similar to the evolutionary process understood in the biological world.

A considerable amount of research has been already done in the academia and industry on evolutionary model relationships with agent adaptations. Anthil [BABA01], uses genetic algorithms to design ant algorithms based on pheromone distribution for selecting the fittest set of parameters for a particular task. MAB-Net [OHTA00] is related to creating an artificial neural network model with a dynamic structure where agents could change in structure by additive and subtractive functions. The evolutionary aspects of some popular experiments are described in Appendix E.

According to Rolf Pfeifer [PFEI01], an evolutionary process could be described by using elements such as genotype and phenotype. A genome encodes all the features of an individual and a genotype refers to the set of genes contained in a genome, where there are fixed number of genes in a defined species. The final organisms that could be developed with the help of genomes, through the process of development are called as the phenotype. It is the phenotypes that compete with one another in an ecological niche, and the winners are selected to reproduce, leading to new genotypes. This is the basis of evolution acceptable in biology. This process has succeeded for millions of years of evolution and is an inspiring analogy of reference when designing artificial systems.

It is common in nature that whenever resources are scarce, competition is beneficial. If the AAANTS model is to implement a process that could adapt and evolve similar to that of the biological world, then special emphasis on encoding the hereditary features in genomes, translating genomes into phenotypes, and using reproduction capabilities to improve new generations of existing species should be considered.

4.6.1 Building the Genome

It is clearly understood that genes are the building blocks of a genome and a genome represents the complete genetical information of a biological entity [PFEI01]. An AT could be described as an analogy to a gene. Hence, the ATs that could be defined through the genes are identified as innate. Within the context of this research it should be stated that there are ATs defined through antecedents, innateness and adaptations though the experiments in chapter 6 only concentrates on innate ATs.

The overall collection of genes that belong to a community is distributed across agent instances. The agent instances could be compared to phenotypes as discussed previously. A gene representing an AT would be represented by several agents that is responsible for the role or task that require this AT within its repertoire of actions (the relationship among agents and ATs are explained in section 3.2.2.1). For example a segment of ants that perform foraging as the primary role would have inherent ATs that are relevant to the task of foraging which are transferred through a limited list of related genes. It should be clarified that there exists one to many relationships among genes and agent instances, which means a single gene assists in implementing a defined phenotypic behaviour among several instances of agents within a caste.

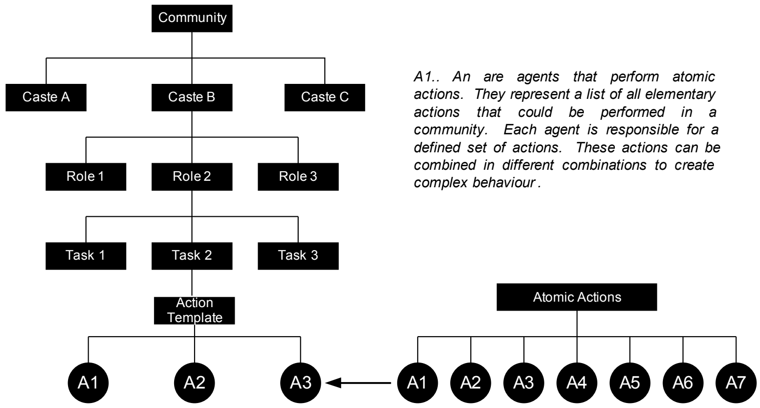


Figure 4.8: Task breakdown within a community and affiliation with elementary actions

The structure of how the AAs and ATs collectively define the overall behaviour within an AAANTS agent colony is represented in Figure 4.8. As stated earlier, the behavioural parallel to the AT is a gene. A collection of genes defines a task that progresses in a depicted hierarchy of assemblages such as roles and castes. A stochastic or declarative approach is adhered when instantiating agents from genomes. For example with reference to Figure 4.8 an agent that belongs to Caste B-Role 2 could be instantiated with Task 2, which would consequently acquire AAs 1, 2 and 3 within the respective AT. A scenario could be that the same task (Task 2) could instantiate 3 agents where each agent represents each of the 3 AAs in the AT. The experiments performed within this research of the AAANTS model only focused on declarative method of instantiating agents for ATs. This means that the number of agents instantiated for a given AT is configured in the innate layer as opposed to using a stochastic process.

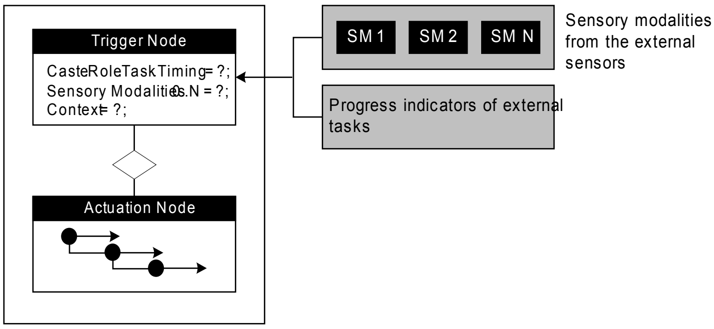


Figure 4.9: Representation of a behavioural gene at a very basic level

The basic structure of a behavioural gene embedded in an instantiated agent is summarised in Figure 4.9. A gene is comprised of a Trigger Node and an Actuation Node. The Trigger Node describes the conditions under which the agent would be active and the Actuation Node is capable of activating AAs of an AT that is under the control of a particular agent. The Trigger Node within an agent is activated from the environment sensations that get processed through the sensory templates (depicted in figure 3.2). The Actuation Node represents the AAs for which the agent is responsible and assists the Trigger Node to make the relationship among sensations and actuations based on rewards from the environment. The AAs within an AT initiate behaviour by sending appropriate signals to the actuators. The AAs within the AT is triggered based on the sensations from the sensory modalities intercepted form the Trigger Nodes of the Gene instances. These sensory inputs also provide the information related to the actuations from agents within the locality.

Even though agents gain the insight to overall structure of an AT consisting of AAs, their contribution may be limited to executing specific isolated aspects of an overall AT (discussed in section 3.2.2.1). Hence, the execution of an AT is realised by the contribution of a collection of agents within the same locality and role.

4.6.2 Transforming the Genotypes to Phenotypes

When the behavioural genes are properly organised into a functional breakdown structure, the overall genome is fully defined. Each gene would spawn agents during the phenotype definition stage. The first step of creating the phenotype is instantiating the genes into distributed pools of agents (figure 4.10 – first step). The relationship among a homogenous pool of agents is closer, and mostly agents representing a given role would be co-located. The closeness is defined not in terms of spatial constraints but in terms of communication. The communicated messages of a group of agents, travels within a shared channel and disseminated within a limited number of agents.

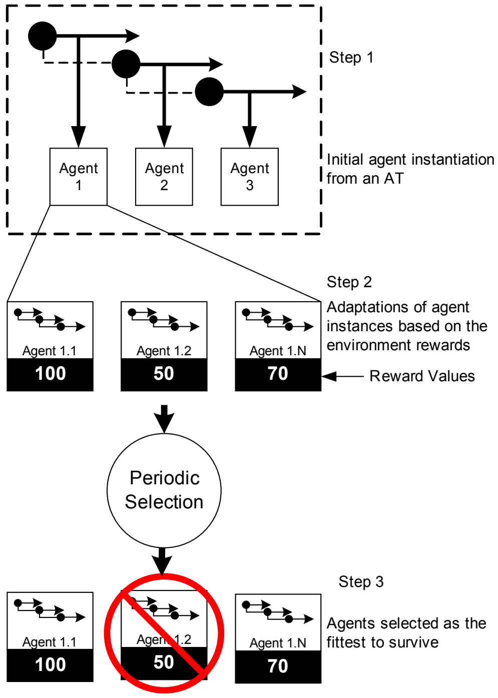


Figure 4.10: Evolutionary changes of agents in an AAANTS system

When all the genes are instantiated, the genotype would take the form of a phenotype. The phenotype characteristics would be based on the genotype; however, the reinforcement from the environment would change the basic gene representation over a period of time (figure 4.10 – Step 2). This behavioural change may be based on the temporal changes in the execution of AAs or change in sensitivity to sensory modalities. However, the new agent might be taken as more suitable to the community based on the rewards, though the older versions would also be coexisting until expelled. The phenotype gets matured as it gains experience and reaches a stage where its instances are fully adapted to the environment. One problem of maturity is the accumulation of redundant AT instances, the competition among which may hinder the overall performance of the community.

An experienced and mature colony should periodically shed redundant burden of under-performing AT/agent instances (figure 4.10 – step 3). This objective is achieved by translating the phenotypes back to the genotypes and filtering the fittest using a technique similar to natural evolution. A technique based on the natural theories of evolution related to ants is described in the continuation.

4.6.3 Survival of the Fittest Behaviours

The Hamilton's formula (Appendix F) [HOLL90] states that, when part of a group sacrifices their reproductive rights for the others, then the reproductive group should be able to perform that function and benefit the overall community considerably better than the former getting involved in reproduction [HOLL90]. This type of altruistic behaviour is very abundant in the insect world. The worker ants sacrifices their reproductive rights for that of their queen who performs that function for the benefit of the whole colony. As far as self-representative genes are contributed by kin, the insects would be willing to give up their reproductive rights. The fitness to reproduce in an altruistic environment could be measured using Hamilton's Formula [HOLL90] for Inclusive Fitness (IF). This formula is explained in detail in Appendix F.

$$(IF) = \frac{E(RS) + \sum b_j E(RS)}{A(IF)}. -(4.5)$$

$E(RS)$ is the average direct reproductive success of individuals possessing the genotype of interest which measures the number of offspring the individual injects into the population, in comparison with the remainder of the population [HOLL90]. The portion of the formula $\sum b_j E(RS)$ is the effect on the reproduction of all of the collateral relatives. The variable b_j is the coefficient of relatedness, which is the probability that the relative j of the focal individual also possesses the allele of interest. The above elements are discounted by $A(IF)$ which is related to the average Inclusive Fitness for the overall population. The Hamilton's rule says that, the benefit to relatives is discounted by their degree of relationship, so that the lesser the relatedness, the greater the benefit must be to counter balance the cost [HOLL90].

The AAANTS model for evolution uses the reinforced credit as the direct reproductive success $E(RS)$ represented in the above formula. The selection is done among the agent instances within an AT. As described earlier, agents representing genes are assigned AAs within an AT. Where there are several instances of the same AT, the related agents across these instances need to compete to be the fittest to be selected for reproduction. However, after performing the above calculation to each AT, based on the relatedness and altruistic background, the best agents and their respective ATs would be selected for reproduction and the rest could be filtered out. The Relatedness Coefficient (b_j) would make sure the selected agents would contain maximum capacity of characteristics of others.

The Relatedness Coefficient (b_j) is measured based on the similarities in the Trigger and Actuation Nodes of the genes. The AT instances could be compared with each other with the use of Situation Index (SI) as represented in formula 4.6. SI provides a unique quantification to each situation represented by an instance of an AT. If SI values of two ATs are closer, it means that the situation of application is similar. The reproduction process would eliminate the less rewarded AT and keep the most suitable instance for future use. The SI values could be used to calculate the coefficient b_j using formula 4.7.

$$SI = [\sum_{j=1}^{m} a_j.s + a_j.t] + [\sum_{k=j+1}^{m} (a_k.p_s - a_j.p_s) + (a_k.p_e - a_j.p_e)]. -(4.6)$$

(Where SI - Situation Index; j - number of AAs in an AT; $a_j.s$ -strength of an AA; $a_j.t$ - execution time of an AA; $a_k.p_s$ - start time of kth AA; $a_k.p_e$ - end time of kth AA; $a_j.p_s$ - start time of jth AA; $a_j.p_e$ - end time of jth AA)

$$b_j = \left[1 - \frac{|SI_x - SI_y|}{(SI_x + SI_y)/2}\right]. -(4.7)$$

(Where SI_x - Situation Index of xth AT; SI_y - Situation Index of yth AT; b_j - Relatedness Coefficient)

The agent instances of an AAANTS colony periodically migrate to an inactive state (Section 3.2.1 – Agent Life Cycle) thereafter a process in the AAANTS platform performs the AT phenotype selection activity based on the formulas stated above. Subsequent to the selection process, the unfit agents are eliminated from the colony and the rest of the agents are transferred back to the active state to continue with their mundane activities.

4.7 Chapter Summary

This chapter discussed another important characteristic of an intelligent artefact – adaptability. Learning is an important part of knowledge and intelligence. It is through learning that an entity can adapt and improve itself to suit the changing environment conditions. AAANTS model is designed as an autonomous and adaptive system with the contribution of a myriad of simple agents. The adaptation of each of these agents is important to the overall adaptability of the community.

This chapter focused on the methodologies of learning adapted by the AAANTS model. The adaptations are discussed in two broad strategies. Initial discussion relates to the use of reinforcement learning as perturbations to the internal knowledge representation of executing agent instances. This is the main mechanism that makes sure the participants of an agent colony coordinate with each other to achieve behavioural congruence. The latter part of this chapter focused on the continuation of existing knowledge to future generations of the colony. The AAANTS model is implemented to periodically regenerate new agent instances by combining behavioural blueprints of successful individuals of a colony. The learnt knowledge of successful individuals is mixed to generate new offspring and unfit individuals are excluded from the colony.

Chapter 5

The AAANTS Framework

5.1 Introduction

The realisation of an agent system is usually done by using the most appropriate open agent platform available in the industry or academia. The purpose of a platform is to provide an application skeleton composed of reusable micro-architectural elements called design patterns that could be customised by developers [BOGN99]. An agent platform provides a generic platform with services, APIs, protocols and standards to realise various agent models. The main advantage of this approach is the reusability of software components[18] that are readily available to create agent applications dynamically. However, in adhering to this approach, the agent system designers would be constrained by the limitations and functional boundaries of the selected platform.

The author evaluated several existing agent platforms (Appendix B) in relation to the objectives and implementation demands of the AAANTS model. It was understood that the requirements of the AAANTS model differentiated from existing platforms due to the following four reasons.

1. AAANTS model is conceptualised on the reinforcement learning methodology and the selected platform should provide software objects that facilitate these functionalities. Most of the evaluated platforms did not possess the supporting APIs to implement reinforcement learning algorithms in an inherent manner. There

[18] "A software component could be identified as an independently deliverable package of software operations that could be used to build applications or larger components and that which assumes an architectural context defined by its interfaces." [BOOC98].

were external libraries that specialised on these aspects, but integration with the evaluated agent platforms were not tested and published.

2. The AAANTS model represents agents defined as a colony with relationship among each other based on kinship. The evaluated platforms demonstrated object-oriented hierarchical relationships, but lacked flexibility expected in the AAANTS model to define agent relationships based on kinship. The existing platforms support the grouping of agents, however, it was difficult to realise various degrees of overlapping relationships among agents.

3. A core attraction of AAANTS model is the ability to define AT with the use of AAs. Further, the AAs within a given AT could be distributed across several agent instances and requires mechanisms for coordination. Most platforms use standard cognitive communication languages that were not suitable for real-time coordination of actions and lacked the capabilities of defining ATs to be coordinated across multiple agent instances.

4. The agent life-cycle concept was present in many agent platforms; however, the states and in-built behaviour related to evolution and elimination of unfit agents were absent. There were possibilities of modifying some systems (e.g. Jade and Zeus) to accommodate this feature, but there were doubts on positive realisation without major modifications to the generic platform.

Based on the above four reasons, a rational decision was taken to build a generic agent platform that could achieve objectives demanded by the AAANTS model.

5.2 Design considerations of the AAANTS Platform

The design of a system should encompass dimensions to achieve the defined objectives. The following dimensions were taken into consideration during the design of the AAANTS platform. These dimensions were derived based on the existing agent system designs [FARH97], general characteristics of software agents (Section 2.2) and the objectives of the AAANTS model [RANA03b].

1. Functional Distribution model

   The method of distributing application functionality among different types of agents that constitutes the agent system is focused in this consideration. The functional approach is particularly well suited to centralised systems, but unprecedented in naturally occurring systems, which divide agents on the basis of distinct entities in the physical world rather than functional abstractions in the mind of the designer [PARU98].

   The distribution model of the AAANTS system possesses hybrid characteristics of existing multi-agent systems, where individual agents in the system are not fully functional monolithic components and contributes collectively towards the overall functionality. Therefore, the distribution model is such that there exists, no single agent that is responsible for a defined function, but the responsibility of a function is shared among a group or more precisely a colony of agents.

2. Internal structure of agents

The internal constituents and their interoperability to accomplish the desired functionality of an agent is a key design dimension. The structure of an agent should facilitate the demands of its functionality in relation to the AI Mix and the agent life-cycle. An agent is constructed as a compound entity composed of elements that support each of these aspects. Further, special data structures were considered to facilitate the retention of information related to sensations and actuations.

3. Coordination of behaviour

Special attention is required to facilitate the locality based communication within agents of a given caste. The messages should be disseminated but effective within a restricted community of participants. Subject-based information dissemination approach was adapted to realise the locality of communication. Further, the design facilitates the definition of ATs to be shared across multiple agent instances.

4. Knowledge sharing among agents

Communication is the primary method of knowledge sharing. However, knowledge sharing among the agents is unique due to implicit and non-pervasive characteristics.

The AAANTS design accommodated a method for knowledge sharing, where information is published with a limited retention lifetime within which the interested agents should fetch the information of interest.

5. Agent environment

   The agent environment is primarily represented by a collection of embedded distributed services responsible for sensations and actuations [RANA99]. These services could be used as neural extensions of the agents for the purpose of sensations and actuations. These heterogeneous services could announce their capabilities within the network and interested agents could collectively use them in their goal-driven activities.

6. Adaptation and continuous improvement

   The platform should provide facilities in terms of functions and data structures to accommodate the reinforcement based learning of agents. The continuous improvement of the agent capabilities is achieved through adaptation and evolutionary methods.

5.3 The AAANTS Architecture

A very important aspect of realising an agent platform is related to selecting and or building an agent architecture. The role of agent architecture is to define the separation of concerns that identify the main functions, which ultimately give rise to the agent behaviour and define interdependencies between them [LUCK97]. The agent and software architectures are very much related to each other where at an abstract level, software architecture involves the description of components from which systems are comprised, the interaction among these components and the patterns according to which the components are combined to form the entire system [BOGN99]. Similarly, the AAANTS architecture is comprised of components that interact with each other into a layered pattern [RANA02a] [RANA02b]. The components of some level of homogeneity are accumulated into a single layer and the interactions of these layers are done through well-defined interfaces. A detailed description of the AAANTS architecture is discussed in continuation.

5.3.1 The Layered Architecture of AAANTS

The AAANTS agent architecture is conceptualised based on the best practices of software architectural and design patterns. The layered approach [COEN97] is one of the most popular architectural patterns found in both natural and artificial system designs. The AAANTS architecture is built on a three layered architecture. Figure 5.1 shows the organisation of the three layers, namely, distributed service layer, service adaptation layer and the layer representing colony of agents [RANA03b].

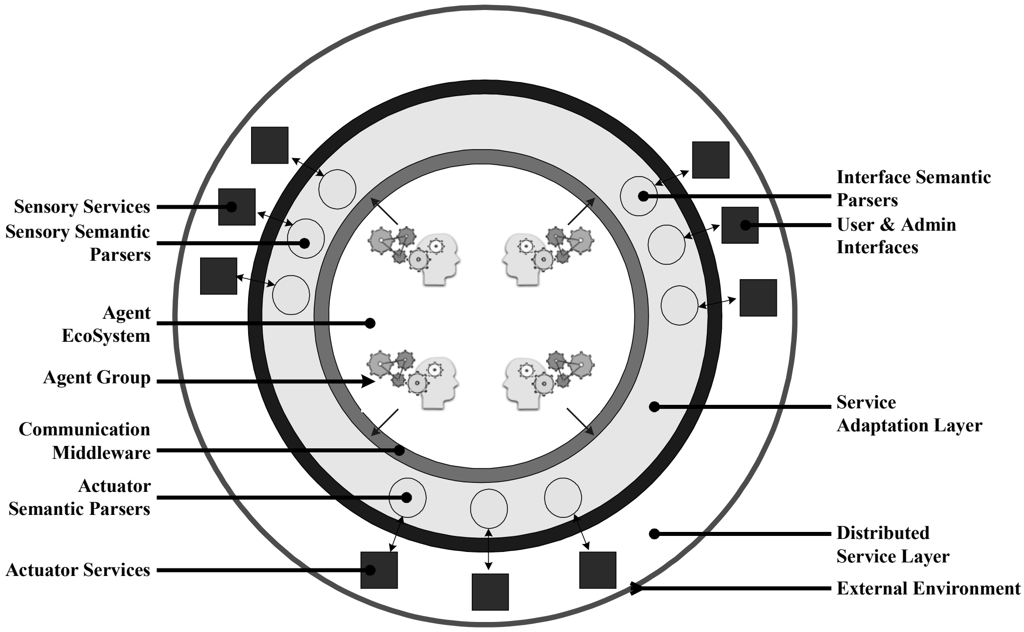


Figure 5.1: Overall layered architecture of AAANTS platform

The external environment refers to the natural environment where sensations are generated and further the actuations change the state of the environment. The external environment is the facilitator for the outer layer that hosts the distributed services responsible for various kinds of sensations and actuations [RANA03b]. The services could be of heterogeneous nature based on the modalities of sensations and actuations. New services could be introduced to the Distributed Service Layer with minimal disruption to the existing services [RANA03b]. A new service, which may either perform an actuator or a sensory function, would become part of the agent platform by simply adapting the common communication protocol among agents and services.

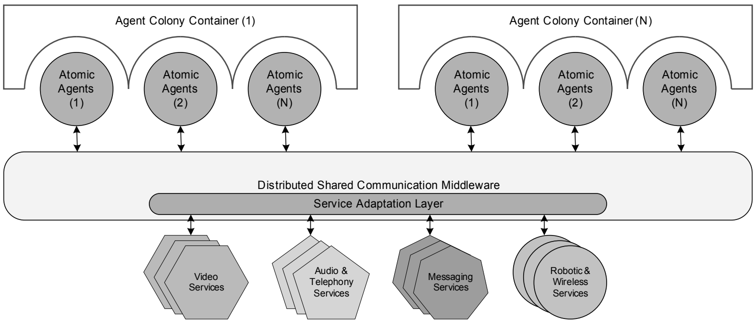


Figure 5.2: Integration of agent colony with distributed services

Figure 5.2 depicts the Service Adaptation Layer [RANA03b] [RANA02b] that consolidates the information exchange process among services and agents. Different types of parsers and translators are used for this purpose. The heterogeneity of the services are neutralised using this layer of functionality. This layer contains communication wrappers that enable the services to translate information to common messaging descriptors.

The inner most layer represents the software agents that may be distributed among several colony containers. The agents interface with the services through the communication middleware which is part of the Service Adaptation Layer [RANA02b]. The agents are organised into groups or more accurately, colonies which represents self-sustaining synergistic entity. A group is sustained by a colony container which provides an execution environment and other facilities common to all agents. The services such as execution control, persistence, life-cycle management, reproduction and fault tolerance is handled by the agent colony container.

A single implementation of AAANTS is considered as a colony since it represents a distinguished localised population. A typical colony consists of at least single instances of the above discussed processes. Among them the Agent Colony Container (henceforth referred as the container) is a dominant process that facilitates fundamental and shared services to agent instances. The container is a run-time environment that contains and executes agent related components and provides a standard set of services to them. Some implementations may consist of many instances of distributed containers for the purpose of load balancing. The containers control the life cycle of the agents that consist of initialisation, start, execution, reproduction, stop and inactive states (life-cycle discussed in section 3.2.1).

The container is of prime importance to the AAANTS concept since there is a requirement to manage an immense collection of agent instances concurrently. Each agent is represented with a self-contained unit of automation that is responsible for a partial outcome of a behaviour demonstrated by the colony. Other related implementations such as “Anthill” [BABA01] conceptualises a nest which is similar in concept to that of a container. However, the AAANTS container does not distinguish a single instance of a container as a nest, since a single nest could be distributed on multiple containers which are linked by a message oriented communication bus.

An agent thrives on the services provided by the colony container and uses its internal structural components to interface with the service adaptation layer. During the creation of an agent, the colony container assigns a thread of control for each agent to be autonomous. Each agent maintains structural elements related to knowledge representation, communication and adaptation.

5.3.2 Structural Elements of an AAANTS Agent

An ant colony is composed of a numerous collection of individual ants. As we discussed earlier, a colony could again be segmented into groups of ants called as casts that are similar in composition and behaviour to implement a specific set of functionality. Though there are differences in different groups of ants in structure and functionality, they are all designed with reusable components. Therefore, introducing a new ant to a group in the colony by the queen is so natural since it is another collection of reusable components that when produced and given life would be a complete autonomous entity that work in harmony with the existing ant community.

Based on the above understanding, the AAANTS model is conceptualised using objected-oriented paradigm due to its inherent facilities for modularisation. Therefore, an agent in the AAANTS platform is composed of a collection of objects, each specialised in a particular type of function in relation to the AI Mix. Each agent has a thread of execution for autonomy and depends on the container for the resources.

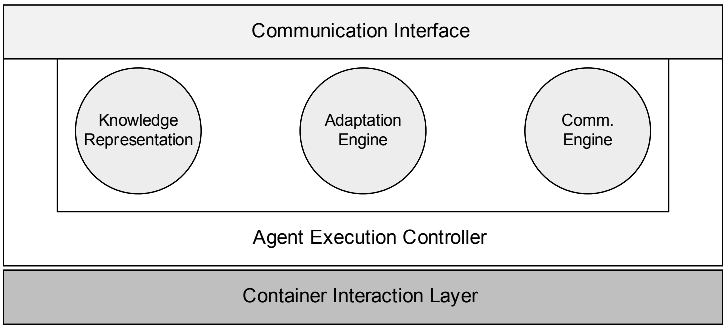


Figure 5.3: Conceptual structure of an agent in the AAANTS colony

The structure of an agent consists of elements represented in figure 5.3. The Container Interaction Layer supports the agent to interact with the agent colony container for the purpose of agent life-cycle management. The agent execution controller interacts with all the components within an agent to produce the final behavioural outcome. Further, there are several components dedicated to knowledge representation, adaptation and communication which are harnessed by the execution controller during execution. These modules contain both data structures and methods required to implement their respective functionalities.

Each agent component is derived from a generic parent class defined in the Java programming language. This "GenericAgent" class is responsible for implementing life cycle related functionality. Further, all the objects within the agent architecture implements the standard Java Serializable interface to support persistence. Consequently, all the agents in a colony could be persisted while preserving the state and thereafter activated to life. Persistence is a useful feature of a progressive colony since it could preserve the state across administrative chores.

The agent execution controller interacts with the communication interface to send and intercept messages. As discussed earlier, the communication middleware publishes messages in a subject-based manner and communication interface facilitates in capturing filtered messages as instructed by the execution controller. The execution controller together with the communication interface filters messages specific to kin and locality.

5.4 The AAANTS Agent Communication

The agent communication with the use of speech acts, protocols, ACLs and ontologies are mainly related to the cognitive agent models. The reactive agent models do not implement such diverse and complex communication methodologies. In contrast to these two popular approaches, the AAANTS model defines an agent communication methodology that is far too simple in nature when compared to cognitive models but relatively comprehensive than reactive models. This approach is appreciated by Martin Beer et al [BEER99], who states that cognitive agent communication languages may be too complicated for certain kinds of agent applications that do not need speech acts and logic to carry out their negotiations.

5.4.1 The AAANTS Communication Framework

The messages among the elements in the AAANTS framework are exchanged through the communication layer which is a sub-system of the overall platform. As represented in Figure 5.4, the communication layer is composed of two message busses, namely, the sensory bus and the actuator bus. The sensory bus carries information related to sensations from the sensory services towards the agents and the actuator bus carries information generated from the agents towards the actuator services. The separation of sensations and actuations into two message buses is done in a virtual manner rather than physical, i.e. two high-level messaging subjects are derived within a single physical message bus. The rationale for this separation is that the message filtration activity by the agent communication interface tends to be relatively simple with two logical message buses.

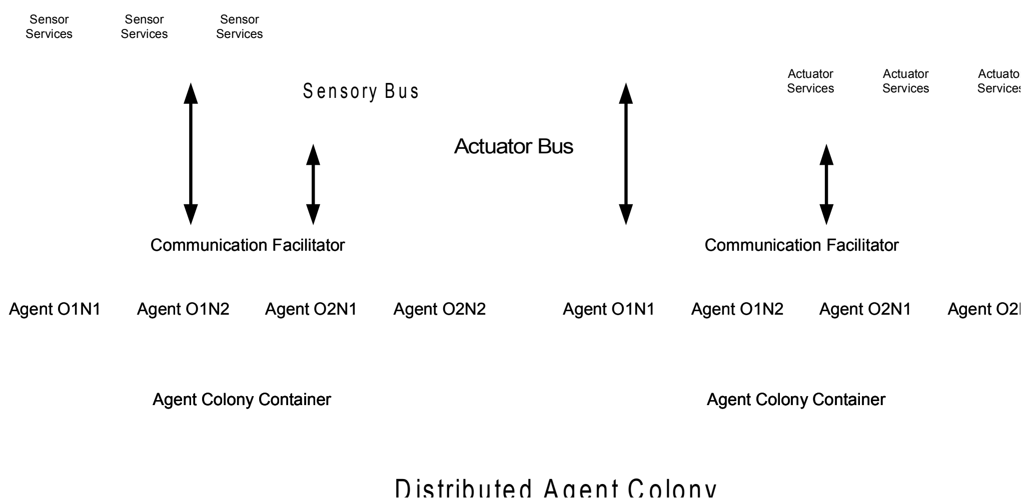


Figure 5.4: Distributed interaction of agents and services through communication middleware

A key feature of the communication framework is the loose coupling of agents and services [RANA99]. Consequently, the agents and the services could be rearranged with more dynamism and freedom. New services could be added and existing services removed with minimal configuration changes to the platform. The configuration changes would concentrate mainly on actuator and sensory semantic parsers.

5.4.2 The Messaging Process

The messages are exchanged within and across agents and services. A message consists of two encapsulated segments, namely, a header and content information. The sensory signals published by distributed sensory services and the actuator signals published by the agents are disseminated in the network on a predefined subject. The subject together with other meta-information is represented in the header portion of the message. The agents and services could publish, subscribe and intercept messages on a subject of interest. This concept adheres to the observer pattern with respect to software design patterns [GAMM95]. A subject simply represents a homogeneous collection of sensations and behaviours. The subjects are organised in a hierarchy, as a result which a message consumer listening to a parent subject would intercept all inherited messages classified under the parent.

The published messages are not retained in the network for later consultation, hence non-persistent. Therefore, the AAANTS framework has provided a service called the Message Queue Service (MQS) [RANA03a] to retain the history of published messages. This essentially acts as a repository of all sensory and actuator messages that have taken place within a specific period of time. Agents could communicate with the MQS to query recent patterns of data. The information stored in the MQS performs an analogous function to that of pheromones used in insect colonies; hence, the messages are only kept for a standard duration and deleted thereafter. MQS have similarities to the blackboard technique [FERB99] [WEIS00] used in several existing multi-agent platforms.

The agents in the AAANTS system work in a community that intercept environmental sensations and convert them to actions that benefit the society as a whole. However, it was clearly discussed that the behavioural contribution of the AAANTS model is due to the coordinated effort of a myriad of agents. Though the sensations and actuations are bridged with the agent colony through the communication middleware, the agents should exchange messages among the kin for the purpose of coordination as discussed in chapter 3. These coordination messages also passage through the sensory bus using differentiated subject headers.

5.5 Team Formation and Coordination

The team formation of a myriad of agents that try to coordinate with each other in order to achieve complex macro behaviour is a daunting task. However, the discussion hitherto describes the building blocks of such coordination in relation to the strategies observed in the natural colony life of insects. The continuation explains the implementation of macro level grouping and team formation among agents within the AAANTS platform.

The AAANTS model is composed of a colony of agents that achieves goals collectively. Therefore, the agents would periodically actuate individual actions to satisfy the needs of the community. When several urgent needs occur at once, there must be a way to select the best outcome for the overall community. One scheme for this might be the use of a central market place, in which the urgencies of different goals compete and the highest bidder takes control [MINS86]. Another way is to use an arrangement called "cross-exclusion", which appears in many portions of the brain [MINS86]. In such a system, each member of a group of agents is wired to send "inhibitory" signals to all other agents of that group which would make them competitors.

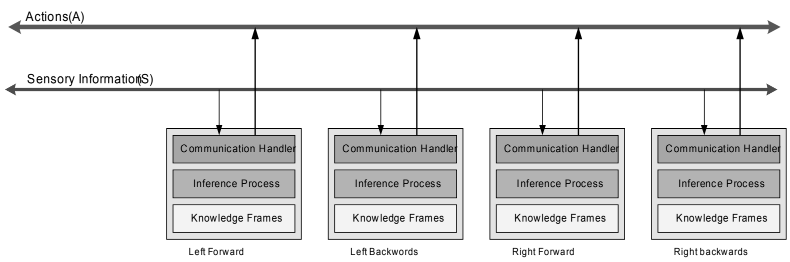


Figure 5.5: Assignment of atomic actions to a colony of 4 agents to implement various ATs

The AAANTS platform is composed of a collection of agents, each responsible for a defined type of activity. For example, with reference to the Figure 5.5, the movement of a robotic vehicle with four wheels and two motors on either side could be controlled by four basic actions such as left forward (LF), left backward (LB), right forward (RF) and right backward (RB) [RANA03a]. These four actions could be executed in various permutations in sequence and or concurrency, to result in a wide range of synchronised intelligent behaviours. For example, the following is a summary of some behaviour that could result from the above mentioned basic actions.

(LF) + (RF) = straight forward movement
(LF) + (RB) = Quick right turn
(LB) + (RF) = Quick left turn
(LB) + (RB) = straight backward movement

With reference to Figure 5.5, different agent compositions and ATs could be devised based on objective of the experiments. For example, four agents are taken into consideration, each responsible for the listed four basic behaviour. Let the objective be to define an AT to implement a 180 degree right turn around an obstacle. To realise this, the following basic actions should work in concurrency and sequence. E.g. Move Forward (LF + RF), Turn Right (LF), Move Forward (LF + RF), Turn Right (LF), Move forward (LF + RF). The AT with 5 steps could be implemented by 2 agents with clear temporal coordination with the use of the sensory and actuator busses.

5.6 Implementation of the AAANTS framework

The implementation of AAANTS platform consists of four principle sub-systems, namely, Colony Definition Tool (CDT), Information Repository, System Execution and Control Components (SECC) and System Monitoring and Visualisation Components (SMVC). Figure 5.6 depicts the implementation sub-systems with the respective interdependencies.

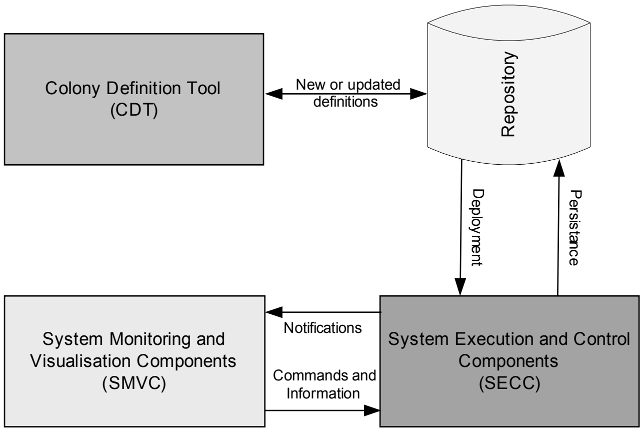


Figure 5.6: Sub-systems of the AAANTS platform

The CDT is a broad term representing a collection of application tools that is used to configure an agent colony. Using the CDT an administrator could initially create an agent colony for a specific purpose and later change the definition to introduce new features to the implementation. SECC represents the core run-time environment consisting of agents, communication channels and services. It is a highly dynamic and active environment that is analogous to an active ant colony. SMVC is a set of tools used by the administrators and the end users to interact with the active agent system. The agents found in SECC would use SMVC components for user notification and feed-back. The repository is a long-term storage mechanism of knowledge, configuration and audit information.

5.6.1 Colony Definition Tool (CDT)

CDT is a broad collection of tools used by the system administrator for initial definition, configuration and maintenance of an AAANTS implementation. The agents in the colony are segmented based on casts, roles and tasks where the nature of sensory and actuator needs of agent groups are different. The characteristics of an agent group are defined explicitly through a configuration interface (Figure 5.7). This interface defines the inner attributes and functions of an agent together with the relationships with the rest of the agent types in the colony.

CDT is also a very versatile administrator tool to modify the layout and internal data structures of an AAANTS implementation. It mainly uses the Information Repository for retrieval and storage of information. Definition Interface is used during initial stages (definition) of implementing an AAANTS colony and thereafter the implemented colony thrives on the definition stored in the repository. Administrators could still perform modifications during the execution of an implemented colony without interrupting run-time functions.

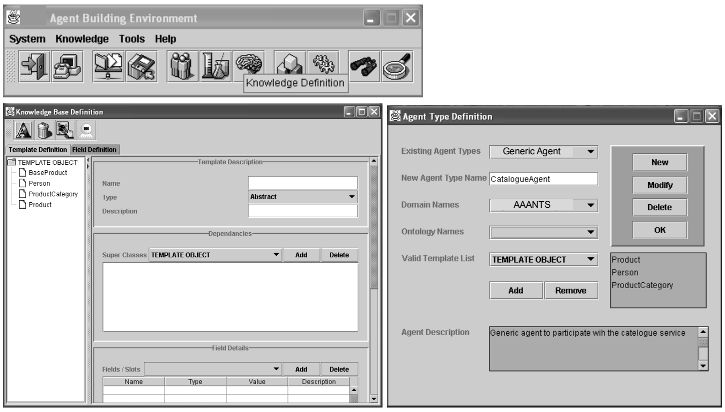


Figure 5.7: CDT agent type and related knowledge template definition interface

Figure 5.7 depicts few of the graphical interfaces used to describe the knowledge structures and agent typology in a colony. These knowledge structures (templates) usually match the attribute information of the sensory details published in the communication bus by the heterogeneous sensory services. Therefore, templates for new sensations could be easily configured when introducing new sensory modalities to the platform. Also the Agent Type Definition interface (Figure 5.7 – lower right) is used to create and describe agent types. Since there could be different types of agent communities in a colony that focuses on different functional aspects, the above mentioned form could be used to formulate new agent types to be introduced to the colony.

When configuring the grid-world and robotic arm experiments, the CDT was useful in implementing the following aspects.

1. Define the list of AAs required to implement overall behaviour. For example the up, down, left and right atomic movements within the grid-world experiment and the movement of the three joints of the robotic arm experiment was defined using the CDT.

2. The configuration of all the innate ATs was done using the CDT.

3. Assignment of the agent instances to the AAs of an AT is also done by the CDT. For example an AT with three AAs could be instantiated by assigning each AA to an individual agent or assigning all AAs to a single agent. The decision of this assignment is based on the instructions given by the system administrator to the CDT.

4. The grouping of the agents to tasks, roles and casts as described in section 4.6.1 is also done using the CDT.

5.6.2 Colony Repository

Colony Repository is a facilitator service responsible for the persistence of definitions, knowledge, and configuration details. It uses a file system structured through the XML format. Therefore, the content is independent of the database and implementation details. Two most common clients of the repository sub-system are CDT and SECC. The repository uses a file system for the storage of information. The data model of the colony is simple and the complex relationships of data structures are created using XML. In addition, the meta-level information related to the sensory templates is also stored in XML format.

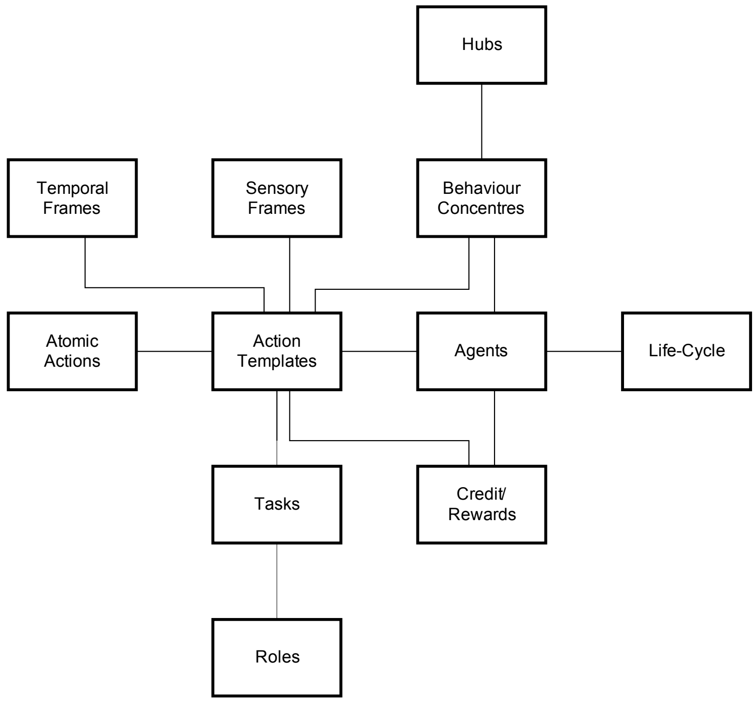


Figure 5.8: Entity relationships within the repository of the AAANTS framework

The high-level entity list and their respective relationships are summarised in Figure 5.8. The definition stage relates to the process of setting up the entities and relationships that is required for the configuration stage. The definitions are conceptual in description and it is during the configuration stage that instantiation of conceptual entities take place. For example, the entities related to AA, AT and Agents which are defined during the definition stage are given actual configuration details. The AA entities are assigned the type of the actuation, initial execution duration and strength. The AT entities are assigned with the constituent AAs and initial sequence of execution. The Agent entities are assigned to ATs based on the requirement of the experiment. For example, an AT with 3 AAs could be instantiated with a single agent or 3 agents.

The configuration of the definitions is performed using the CDT. Among the entities depicted in Figure 5.8, Temporal Frames, Sensory Frames and Behaviour Concentres is considered as meta-level entities. These entities are used during the execution phase of the agent colony. The configurations done using the CDT is instantiated to an evolving knowledge based by the SECC. For example, in the grid-world and robotic arm experiments, the iterative episodes of interactions with the environment is persisted in the knowledge base for future needs.

The configuration activity initiates with the definition of elementary structures such as the Atomic Actions, Temporal Frames and Sensory Frames. These become antecedents to the definition of Action Templates. When the ATs are defined, the dependent entities such as Tasks are Roles could be derived. The definition of the Agents and Agent Life-Cycle related structures are defined as the final phase of the configuration. The entities such as Behavioural Concentres, Rewards and Hubs are required for the execution phase of the implementation.

5.6.3 System Execution and Control Components (SECC)

After the system definition phase, AAANTS colony could be instantiated using the definitions found in the Repository. An instantiated platform is in the execution phase of a typical colony implementation. All the active components in the execution phase are called as System Execution and Control Components (SECC). SECC sub-system could be broadly segmented into Agent Colony Containers, Distributed Services and Messaging Buses.

The primary constituent of SECC is the Agent Colony Container. The instances and the architecture of the colony containers is depicted in figure 5.2 and 5.4. The inner most layer in figure 5.1 represents the container that facilitates the agent colony. This is the dominant process that mainly focuses on the well being of the agent instances. The container is a run-time environment that contains and executes agent related components and provides a standard set of services to them. Some implementations may consist of many instances of distributed containers for the purpose of load balancing. The containers control the life cycle of the agents which consists of several sub-states within active and inactive states (agent life-cycle section 3.2.1).

The container addresses the following issues in order to provide a comfortable environment for the agents.

- Performance: The concurrent use of external resources should be optimised and life-cycles of the agents should be managed properly.
- Scalability: Depending on changing demands, the instances should be deployable on other fault tolerant container instances.
- Security: Authorised access to the components from external entities is handled and managed properly.

- Availability: A running system should be easily recovered with minimal down-time during a failure.

The agents are autonomous in nature and the container only provides an execution environment and resources for facilitating the functioning of the life-cycle. The primary control of the container over the agents is the capability of changing the life cycle state from inactive to active and vice versa. With respect to figure 5.6, the container is responsible for interacting with the SMVC and Repository. The container reads all necessary definitions from the repository during the start-up and persist them back with enhancements in relation to reinforced knowledge. Further, the container links up with the SMVC to offer control and monitoring capabilities to the system administrators.

The next element of importance to the SECC is the Distributed Services. The broad term "Service" is used to represent all the sensory and actuator related processes in the external environment that are interfaced with the agents. The interface between these services and the agents are realised with the use of various types of parsers present in the Service Adaptation Layer (Figure 5.1). The environmental services could be used to produce sensory inputs (audio, visual, touch, smell, taste, etc) and actuator services (muscle movement, voice, etc) found in the natural environment. The sensory information such as video and audio could be quite complex to process. Therefore, the objectives of the services are to capture the sensory information, convert to a simple pattern and publish as a message to be intercepted by the agents. Actuator services are also controlled through properly constructed messages.

The agents are able to convert their intensions to a message that could be understood by the services.

The Messaging/Communication Bus is another important element of the SECC that supports information exchange classified under subjects among agents and services. For instance, a publisher of information could open a connection to the messaging bus and submit broadcast messages on a defined subject. In the meantime, there could be any number of subscribers already connected to communication bus listening on a subject. The subscribers would be asynchronously notified of any message available on the channel on the interested subject. There could be many publishers and subscribers on a given subject. Also, a particular subscriber could listen to more than one subject at a given time. This described aspect of subject-based addressing facilitates the "implicit" communication among the agents as described in chapter 3.

The messaging bus is distributed in nature and could be described as a Message Oriented Middleware (MOM). Java Messaging Service (JMS) is used for this purpose. The main purpose of the communication bus is to facilitate agent-to-agent and agent-to-service communication and messaging.

5.6.4 System Monitoring and Visualisation Components

SMVC is a complementary collection of tools that assist the administrators and end users to visualise the operational aspects of an AAANTS platform during run-time. It generates reports and statistics of agent group activities. Further, some interfaces of the

SMVC could be used to simulate sensory signals to conduct tests in controlled environments. This environment was very helpful during the experiments conducted to gather statistical information and to create simulations.

The grid-world and robotic arm experiments benefited immensely due to the assistance from the SMVC tool-set. All the experimental input and output data were manipulated with the help of this tool. Further, the SMVC tool was useful in acting as a simulator in most of the experimental scenarios. Especially, in the robotic arm experiment, the SMVC tool assisted a lot until the physical aspects of the experiment were established.

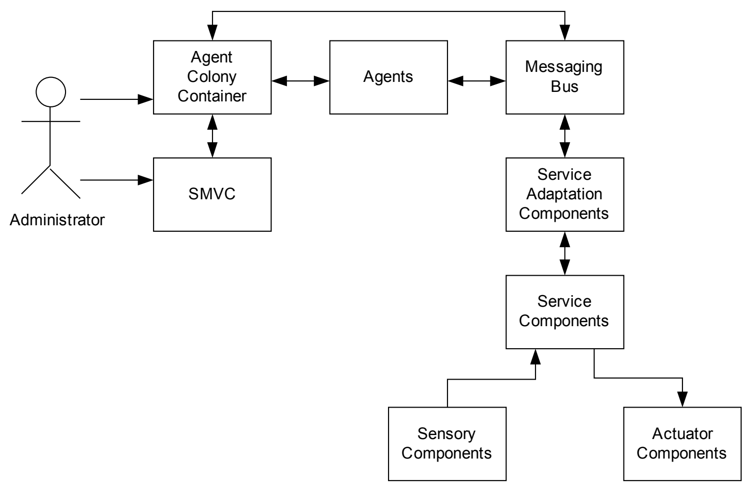


Figure 5.9: Entity and Component Collaboration Diagram of the AAANTS Framework

The overall collaboration among components/entities of the AAANTS framework is summarised in figure 5.9. It should be emphasized that the system administrators only interact with the Agent Colony Container and SMVC entities. The agents are exposed to the rest of the components through the Agent Colony Container and the Messaging Bus. The agents also interact with the services embedded in the external environment through the Messaging Bus. There is loose coupling among the container and the distributed service components due to the separation created by the messaging bus.

5.6.5 Implementing the Grid World

The conceptual architecture of the Grid World experiment implementation setup is depicted in Figure 5.10. The implementation architecture is composed of two key implementation elements, namely, the grid world simulator and the AAANTS framework. The grid world simulator is integrated with the AAANTS framework through a communication layer. Hence, the agents and the simulator exchange message based information through the communication layer. The messages are exchanged in XML format and segmented into two variants, namely, sensory messages and actuator messages. The actuator messages are sent from the agents to the simulator and sensory message sent vice versa. Both of these messages could be clearly identified based on the subject of the message defined in the message header. This aspect is clearly described in Section 5.4.

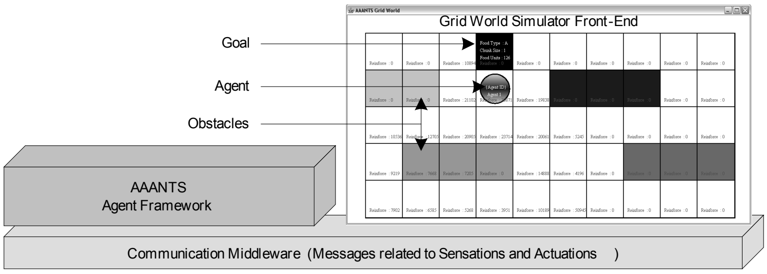


Figure 5.10: Grid-World Simulator integration with the AAANTS framework

Both the grid simulator and the agent framework were developed on a multi-threaded platform, hence facilitating concurrent execution. The simulator possess all the routines required to create agents, move agents from one cell to another, sense the neighbourhood characteristics and change the obstacle arrangement. These routines could be invoked from the outside entities using well-defined APIs. Consequently, these APIs interface the agents of the AAANTS framework with the simulator. The advantage of such modularisation is quite evident in the software industry when building generic design patterns.

5.6.6 Integrating a Robotic Arm to the AAANTS Framework

There are two aspects to this implementation: structural construction of the standalone robotic arm and building the interface to the AAANTS platform. The structural construction of the arm is conceptually depicted in Figure 5.11 and some aspects of the actual implementation in Figure 5.12. The arm has three joints (J1, J2 and J3) and each joint consist of a motor and a rotation sensor. The rotation sensor could simultaneously sense the angular change while moving a joint.

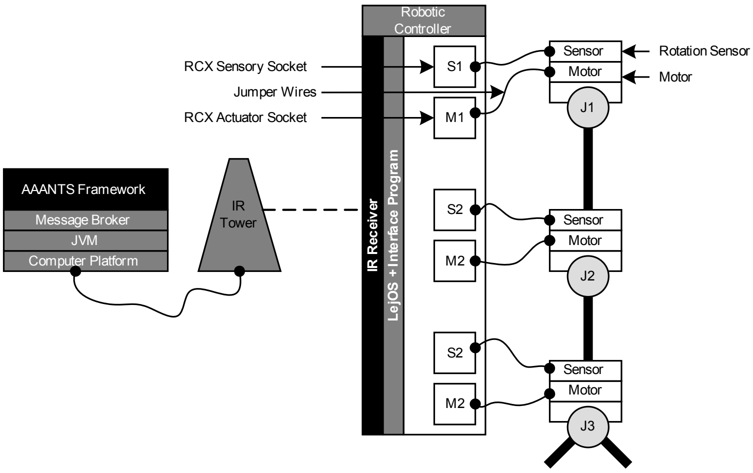


Figure 5.11: The conceptual design of the Robotic Arm

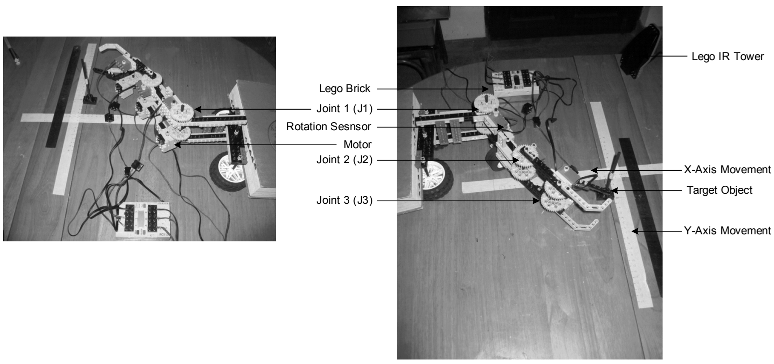


Figure 5.12: Robotic arm implementation using Lego Mindstorms Kit

The interface aspects to the AAANTS framework is controlled and monitored by programs that run on the robotic controller constructed from a Robotic Kit known as the Robotic Invention System[19]. The robotic controller could be described as the nervous system of the robotic arm and the real brain activity happens within the AAANTS platform. The robotic controller has 3 motor sockets and 3 sensory sockets which connect to the respective sensors and actuators of each joint using jumper wires.

The brick controller program (BrickController.java) executes on an open source kernel called as Lejos [ANDE01]. Lejos consists of Java libraries that help to control its peripherals while communicating with the parent program on the host using infrared signals. The author has also initially experimented with NQC (Not Quite C) language using Operational Codes of the Lego Brick. The Java (Lejos) based solution was considered superior in technology as well as in design when compared to the latter.

5.7 Chapter Summary

This chapter concentrated on the implementation aspects of the AAANTS agent platform. A new agent platform was developed to realise the objectives of the research since the existing platforms required considerable adaptations and enhancements. The AAANTS implementation was able to generate satisfactory test data to justify usefulness of the AAANTS theoretical model.

[19] Robotic Invention System 1.5 and 2.0, belong to the core set of the Lego Mindstorms product range introduced and marketed by The LEGO Group.

The architecture of AAANTS took into consideration the characteristics of the existing agent architectures and best practices of software design patterns. The architecture is based on a 3-layered design which facilitated modularisation to a great extent. Some of the major components of the AAANTS architecture are distributed services, messaging middleware and agent colony containers.

The AAANTS agent architecture was mapped to an implementation model based on the Java platform. The richness of the language, APIs and acceptance was very useful in realising the conceptualised AAANTS framework within a short period of time. The generic AAANTS implementation was extended to facilitate the two primary experiments of this research: the Grid World navigation for foraging and Robotic Arm movement. The extension of the AAANTS implementation for these two experiments justified the generalization of the framework.

Chapter 6

Simulations and Experiments

6.1 Introduction

During initial stages of the research it was attempted to implement the AAANTS model using popularly accepted agent platforms such as Zeus [COLL99] [NWAN98], Jade [BELL03] and Grasshopper [BAUM00]. These platforms were very useful during initial modelling phases, however as the research progressed, the restrictions (discussed in section 5.1) of these frameworks prompted the author to create a generic, though native framework to achieve the required experimental flexibility of the AAANTS model. The initial lessons learnt became stepping stones to arrive at the finally crystallised platform described in Chapter 5.

The rest of the chapter focuses on a range of experiments that were conducted to assess the capabilities of the proposed model in realising the research objectives.

6.2 Experimentation Methodology

Two experimental domains were explored to evaluate whether the AAANTS model delivers the objectives of this research. The experimentation domains were foraging in a grid world and optimising the movement of a robotic arm with three joints. An orthogonal objective of using two domains of experiments was to evaluate the reusability of the proposed model in relation to various learning situations.

The primary experiment was to develop an environment to simulate foraging activities of insects. The food collecting behaviour of insects called as foraging is a popular domain of experimentation among the researchers of collective intelligence [HOLL90]. Further, the experiments related to a grid world where agents are supposed to transit through states with the objective of finding the optimum path in reaching a defined goal have been popular among the artificial intelligence community for years. The original grid world problem was enhanced to include foraging related aspects to the simulation. Key control variables and their configurations for different experiments are listed in Table 6.1.

Variable Description	Ex 1 – Sc 1	Ex 1 – Sc 2	Ex 2 – Sc 1	Ex 2 – Sc 2	Ex 3 – Sc 1	Ex 3 – Sc 2
Grid Size	10 x 5	10 x 5	10 x 5	10 x 5	10 x 5	10 x 5
Obstacle arrangement	Constant	Constant	Constant	Constant	Constant	Constant
Characterises of agents	Constant	Constant	Constant	Constant	Constant	Constant
Learning algorithm	MC	MC	MC	MC	AAANTS	AAANTS
Number of agents	1	1	2	4	4	4
Number of search threads	1	1	2	4	1	1
Reward distribution	Equal	Dispro-portionat e	Dispro-portionat e	Dispro-portionat e	Dispro-portionat e	Dispro-portionate
Look-ahead	1 Step	1 Step	1 Step	1 Step	2 Step	2 Step
Shared memory context	No	No	Yes	Yes	Yes	Yes
Implicit communication	No	No	Yes	Yes	Yes	Yes
Use of action templates	No	No	No	No	Yes	Yes
Knowledge Representation	Individua l	Individual	Shared	Shared	Shared	Shared
Initial state initialisation	Random	Random	Random	Random	Random	Random
Exploration probability and rate of reduction	Constant	Constant	Constant	Constant	Constant	Constant

Table 6.1: Control variable summary across all grid-world experiments

There are many flavours of reinforcement learning methods such as Monte Carlo (MC), Dynamic Programming (DP) and Temporal Difference (TD) [SUTT98a]. Each of these methods have advantages and disadvantages based on the domain of application. It is considered that MC methods scale better with respect to state space size than standard, iterative techniques for solving systems of linear equations [BART94]. Further, an MC method does not require explicit knowledge of the transition matrix of the problem domain [BART94]. Hence, MC method was selected as the reinforcement learning algorithm for the experiments of this research due to the above stated uniqueness and also due to the similarity in concept to other similar reinforcement learning methods. Further, the fundamental learning algorithm of the AAANTS learning model was based on the MC method.

In all the experiments, the exploration probability was kept constant. The initial exploration probability was kept at 0.99, which thereafter was linearly reduced after each episode. The reduction rate of exploration probability hence was kept at a constant across all the experiments. Further, a uniform reward distribution strategy was adhered across all experiments except in the grid world experiment 1 scenario 1. The reward distribution was performed episodically while keeping state values to ascend from home to destination, hence encouraging the agents to follow a path of ascending state values similar to the effect of pheromones in ants.

Subsequent to the grid world experiment, a robotic arm related simulation was configured using the AAANTS platform. In this experiment, a group of three agents controls a robotic arm constructed with three joints to perform basic human upper limb behaviour. Each joint is equipped with a motor and an angle sensor to control the movement. Each joint is controlled by an agent who is capable of rotating the rod attached to the respective joint in a single plane within a 90 degree limit. The joints were designed to move in sequence from J1, J2 through J3, followed by bulk reinforcement for the overall behaviour. The initial position of the arm is fully stretched to have 180 degree angle among joints and the overall arm to be 90 degrees in relation to the target object.

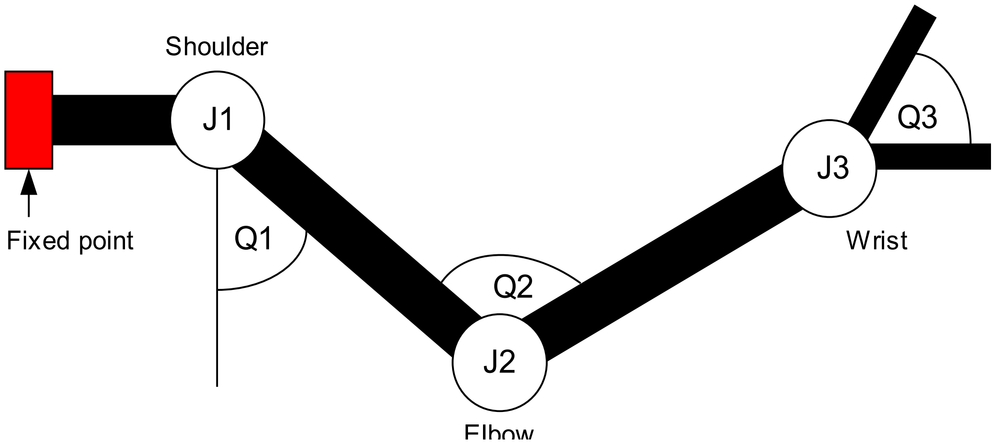


Figure 6.1: Conceptual Model of the Robotic Arm Model with 3 degrees of movement

The agents have the capability to instruct a specific joint to move at a specific angle (e.g. Q1, Q2 and Q3 – Figure 6.1) using the respective motor and angle sensors. Hence, by applying the angular movement to these 3 joints, the overall arm could be used to grab, push or myriad of other movements.

Variable Description	Experimental Values
No of Joints	3
Obstacle arrangement	Constant
Characterises of agents	Constant
Learning algorithm	AAANTS
Number of agents	4
Reward distribution	Disproportionate
Look-ahead	2 Step
Implicit communication	Yes
Exploration and Exploitation Strategy	Constant
Use of action templates	Yes
Knowledge Representation	Shared
Initial state initialisation	90 degrees from the target object
Exploration probability and rate of reduction	Constant

Table 6.2: Control variable summary across all robotic arm experiments

The reinforcement for a complete three joint movement is calculated by an algorithm that takes the two dimensional proximity of the wrist joint to that of the target object. The reinforcement is given as a single value to the overall outcome of the 3-joint movement and thereafter distributed to the respective agents. Key control variables and their configurations for the experiment are listed in Table 6.2.

6.3 Foraging in a Grid World

The grid world experiment was designed to evaluate the core objectives of the research related to emergence, innateness and implicit communication. The experiments were designed to test each aspect of the hypothesis which would be discussed in each respective section. The subsequent sections would explain the design of the grid-world simulation followed by the details of the experiments.

6.3.1 The Simulated Grid-World Environment

A grid world is an area with a restricted boundary as depicted in Figure 6.2. At a given instance there could be one or many participants within the grid that may perform state transitions either to reach the destination Food Source (FS) which is the goal state or else to return back to the nest with the already captured food elements after reaching the goal state. Each participating agent is analogous to an ant in a colony.

A grid world could be experimented along several dimensions such as spatial, temporal and functional. In terms of spatial aspects, the total grid is divided into small squares called as cells. Most of the discussed experiments are based on a 10 x 5 grid, but the same experiments were performed on 20 x 30 and 30 x 40 grid environments to assess the scalability. The movements within the grid are done on temporal clock cycles and the main functions of agents are searching and transporting food. The grid and obstacle layouts are totally configurable using the grid world simulator front-end application.

The participants could travel from one cell to another in a horizontal or vertical direction, but restricted in travel diagonally. A single participant could inhabit a cell at a time during the search stage, though several may travel together while transporting a food unit collectively. However, there could be some cells that are obstructed and impassable by the agent to make the foraging task more realistic.

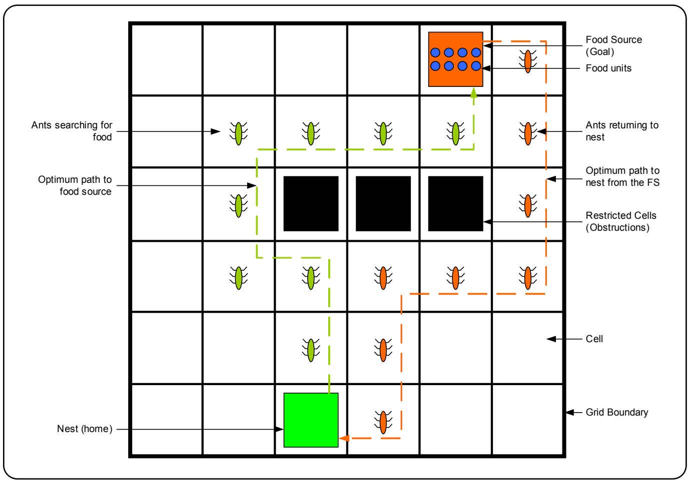


Figure 6.2: Grid world model for the ant foraging simulation

The grid world contains a single cell representing the Nest (the home position) of the ants and another cell representing the food source (FS). The location of FS is called as the "goal". The objective of the participants would be to find and transport the food items from the goal position to the nest. The effectiveness of the participants is gauged by the optimisation of movement during the iterative foraging behaviour. The simulator is developed in a configurable manner so that the dynamics of the grid world discussed above could be adjusted based on the requirements of the experiments.

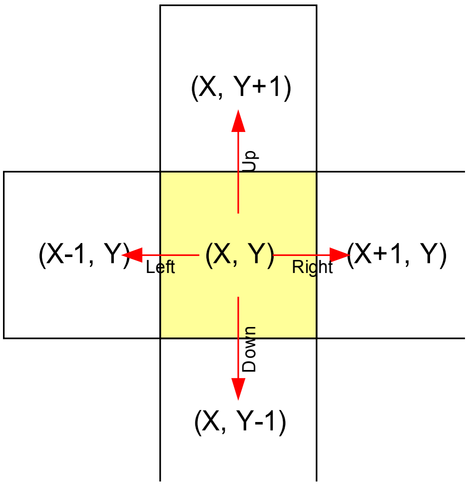


Figure 6.3: Grid state naming convention

The participating software agents capture state characteristics with respect to the neighbouring cells in the grid world. The cell characteristics may convey different semantics to different agents if a standard nomenclature is not adopted. Hence, a neighbouring cell naming convention was adopted to uniquely identify each cell within the community. Cells are referred with respect to the originating cell using X and Y coordinates. Each cell has four neighbours and could be referred using a simple formula that increments and decrements the values of X and Y coordinates as depicted in Figure 6.3.

6.3.2 Grid World Experimental Guidelines

The guidelines that are applied to the entire set of grid-world experiments could be described along the following aspects.

1. Each participating entity in the grid is represented by a single agent instance within the AAANTS framework.

2. The overall agent behaviour within the grid could be segmented to food search and transportation activities.

3. Agents possess inherent capabilities to perform the following actions.

    a. **Search/Move** from cell to cell in four directions - up, down, left and right. Always the boundary of the grid has to be checked before committing to any type of movement. The search mode is dominated by two techniques, namely exploration and exploitation. The exploration and exploitation behaviour in experiment 3 is based on the procedure as explained in chapter 3. The exploitation search is related to the movement to the next cell inline with the strategies listed in section 4.3. Hence, exploitation uses heuristics based on past reinforcements to decide on the best possible movement.

    The other opposing search mode is exploration which uses a stochastic approach. When adapting a stochastic approach, the search mechanism could be called as a blind search where the next selection of state becomes unpredictable. The search approach uses a mix of exploration and exploitation modes which initially gives higher preference to exploration and thereafter moves the preference inclined towards exploitation. The probability of using these two modes could be configured using the SMVC. For all of the experiments, the exploration probability was initialised at 95% which was linearly reduced subsequent to each episode, which results in using higher probability of exploitation after considerable amount of episodes.

b. **Pick** food from the FS.

c. **Carry** food while moving from goal to home.

d. **Drop** food at a location. This could be done before or after reaching the destination. Dropping food prior to reaching the destination is related to reaching a higher level of collective behaviour by segmenting overall grid into controllable segments.

e. **Sense of direction** from source to destination. This is done through a simple gradient calculation. Further, the sense of direction should be improved to calculate with reference to other objects found in the environment. A simple formula to calculate the direction would be to use the following: $D = |\delta y| / |\delta x|$. The direction described is calculated among two fixed points: nest (Home) and food source (Goal). There could be situations where these fixed points may change during a single execution cycle of episodes within the grid world. Therefore, when origination point changes, agents should be able to recalculate the new direction based on the ratio difference of the latter to the former.

4. The search aspect of an agent is summarised below.

   a. Individual agents start from the nest and move from cell to cell in search of the FS.

   b. An agent reaching the FS would receive a reward proportional to the strength of the source, e.g. finding a FS with larger food item density would generate higher rewards than a FS with low food item density.

   c. The agents should be able to perform the search function collectively. The agents could disperse to different segments of the grid and the first to reach the goal should implicitly communicate to others.

   d. After initially locating food, agents should be able to determine the direction of food respective to the nest. This could even be communicated to other agents – implicitly.

5. The transportation of food items from the FS to the nest is summarised below.

   a. The agents that reach the FS should execute following actions in sequence – pick, carry and drop.

   b. Transportation could also be done collectively where part of the colony may drop food halfway along the path to the nest and the rest of the agents may adapt to transport from that point onwards. It is of interest to find the efficiency of collective behaviour against individual effort of transportation.

c. The return path to the nest need not necessarily overlap with that of the reaching path. Since the search path may be the relative optimal path from nest to the FS, the return path too could align with the reaching path though not in an overlapped manner.

6. When an agent reaches the goal state an episode ends, whence the system releases a reward value to be distributed among the states that contributed to reach the goal. The rewards could be distributed in an equal manner or ascending manner from source to destination. The value of a state (V_s), when proportionately distributed is calculated using the formula (6.2). The same rewarding mechanism would happen when reaching the home state; however, the states maintain these two types of rewards in differentiated variables. This would make sure that the agent would refer the correct state values based on the current goal.

$$V_s = V_s + \frac{(R * F)}{C} . -(6.2) .$$

(Where R: Final Reward, F: Food Units at the goal, C: No. of states to reach goal from home location)

When calculating reward per state disproportionately, the formula 6.2 is slightly modified to multiply by a reciprocal based on the distance of each of the states in the converged path to the goal state.

6.3.3 Grid Experiment 1 – Single agent foraging

This experiment was done as a reference experiment to the AAANTS model. This experiment involves a single agent that search for a FS from a fixed home location. The learning methodology is based on Monte-Carlo algorithm and the experiment is void of any type of inter-agent coordination mechanism. The single agent that traverses the grid throughout the experiment could be called as the sole contributor for recognising the optimal path. The experiment is also void of any type of shared memory and implicit communication. The key emphasis is that the inter-agent communication is absent in this experiment and is used as the control experiment to gauge the effectiveness of the proposed coordination model.

6.3.3.1 Scenario 1: One Step Look-Ahead Policy using Monte Carlo (MC) Method with Proportionate Reward Distribution

This is a control experiment based on the traditional reinforcement learning technique of the Monte-Carlo method. The results of this experiment would be subsequently compared with that of the AAANTS model. In this scenario an agent uses the Monte-Carlo based reinforcement learning method with a mixture of exploration and exploitation strategies. All states are initialised to an identical value and the state transition is based on the state values (V_s) of the four neighbouring (left, right, top and bottom) cells. In this experiment, one step look-ahead of adjacent states were performed though in most MC applications multiple look-ahead is done similar to the Dynamic Programming methodology. The uniqueness of this experiment is that the reward distribution is done in a proportionate manner.

The adaptive algorithm of experiment-1: scenario-1 is as follows: (The Exploration Quotient (EQ) refers to the probability of exploration allowed within the community)

```
Initialise all the states to identical values
Agent transit to states until goal reached {
        Decide whether to Explore or Exploit based on the randomness
        and EQ
        if (Exploration Selected) {
                Perform state transition based on a random algorithm
        }
        else  {
        List the highest valued state from the neighbours
                if there are many states with equal state values
                        then, randomly explore and transit to a state out
                        of the selected
                else if a leading state with the highest value is found,
                        then, move to the state with highest value
        }
        Change/Reduce EQ
}
```

This experiment was performed using a single agent for static arrangement of obstacles. For each episode the number of steps to reach the goal is tabulated, and further all the state values of the 10 x 5 grid are tabulated after reaching 100 episodes. The Figure 6.4 depicts the average number of transitions taken to reach the goal state over 40 episodes.

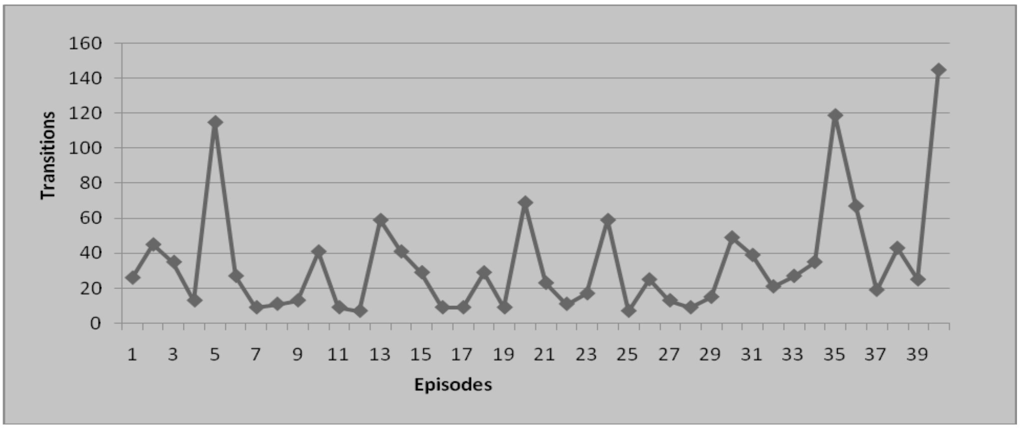


Figure 6.4: State transitions using single agent scenario with equal rewards distribution among states

The Figure 6.5 depicts the state values of the 10 x 5 grid after 40 episodes based on one-step look-ahead MC method. The state value arrangement in this figure shows one peak local optima where as the repetitive experiments conducted with different arrangements of obstacles and grid sizes (20 x 30 & 30 x 40) resulted in various patterns of state value distributions with evident multiple local optima. It was observed that in most experiments the agents get trapped in local optima without ever reaching the goal state.

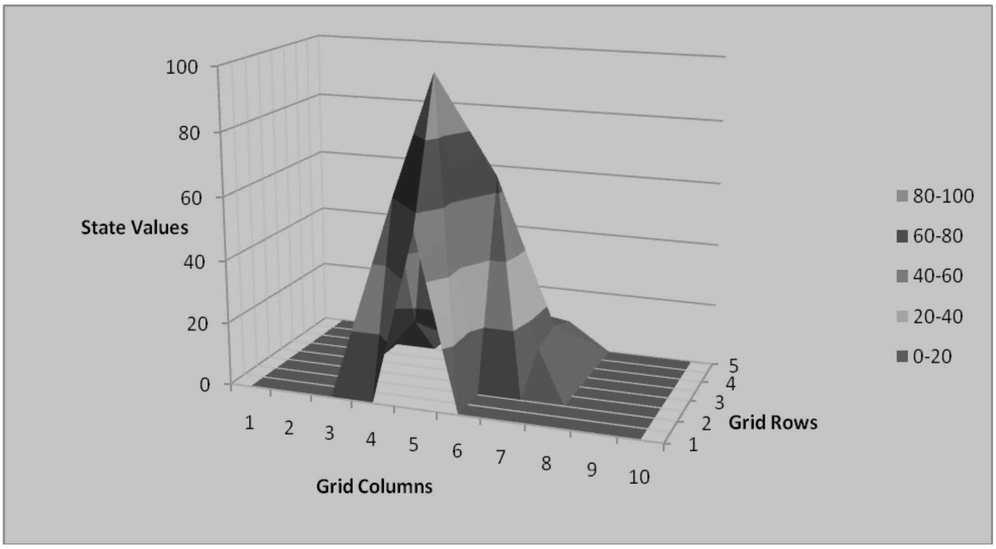


Figure 6.5: Reward distribution among 10 x 5 grid world using a single agent on MC 1 step look-ahead

The effectiveness of a learning method is based on the improvement of the expected behaviour over a period of time while iteratively being reinforced from the environment. In this context, the number of transitions to reach the goal state should reduce over a period of time. However, with reference to the collected data (Figure 6.4) such improvement is not evident from the results of the experiment. This learning method was not able to accurately converge on an optimal path to the goal state.

6.3.3.2 Scenario 2: Disproportionate Reward Distribution among the Participating States

The objective of this experiment is to evaluate the capability of an agent to exploit a path with an incrementing gradient of state values when reaching the goal state. This is similar to the varying concentrations of pheromones laid by the insects to demarcate the closeness to food sources.

The scenario 2 of experiment 1 is based on identical variables to that of scenario 1 which is primarily a Monte-Carlo reinforcement learning algorithm for one-step look-ahead goal search in the grid world. Scenario 2 differentiates from scenario 1 primarily on the reward distribution mechanism which adopts a method of allocating rewards disproportionately among the states that contribute in an episode to reach the goal. A higher reward proportion is given to states closer to the goal state and lesser to the states near the home location, hence in a descending manner.

Consequently, a decreasing trail of rewards is assigned to states from the goal state to the home state. The reciprocal of the above distribution mechanism is applied to rewards given when reaching the home state, however, stored in a separate variable in each state.

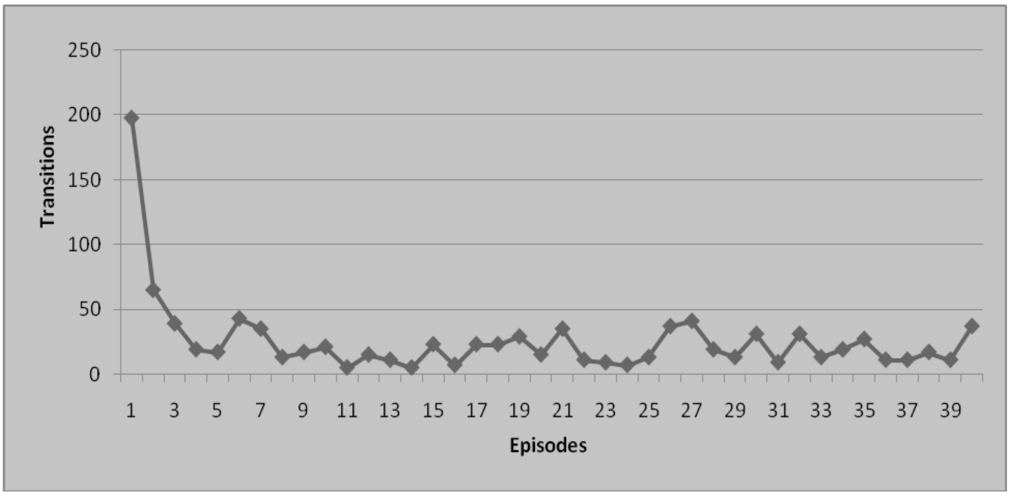


Figure 6.6: State transitions using a single agent with disproportionate reward distribution function

The scenario 2 algorithm demonstrates considerable improvement over the former (scenario 1) in converging to an optimum path. This is evident with reference to Figure 6.6, where the number of transitions to reach the goal state considerably reduces during the initial episodes.

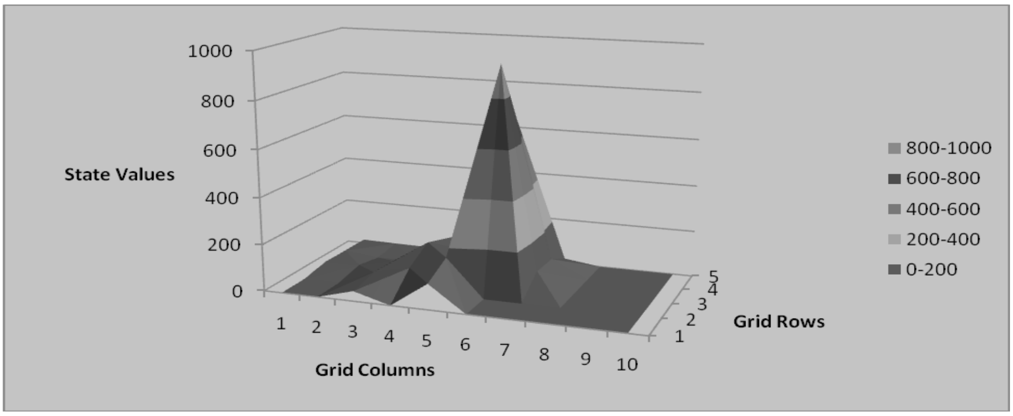


Figure 6.7: Reward distribution among 10 x 5 grid world using a single agent on MC 1-step look-ahead

The reward value distribution among neighbouring states shows clear concentration along the optimum path from home to the goal (Figure 6.7). The probability of generating local optima based on this method was relatively low; however, there were few situations of local optimality based on different obstacle arrangements. During some episodes it was noticed that the agent randomly gets stuck in local optimal situations. This kind of behaviour was not that prominent when considering the entire set of experiments, though it gives a clear indication that the methodology could be further improved. The experiment was extended by changing the goal state after convergence and further obstacle arrangement of the grid. Scenario 2 shows some level of tolerance to these changes whereas scenario 1 was unable to handle these situations in most occurrences.

6.3.4 Grid Experiment 2 – Cooperative Foraging Using Monte-Carlo Method

This is the second control experiment conducted to compare the capabilities of the AAANTS model. The objective of this experiment is to evaluate the effect of concurrency and implicit communication to the standard Monte Carlo algorithm. The scope of this experiment is similar to experiment 1, but with the inclusion of several agents foraging concurrently and further adhering to the disproportionate reward distribution method throughout the experiments. The reinforcement learning is based on the Monte-Carlo algorithm with 1-step look-ahead which is similar to the former experiment.

The enhancement in this experiment is that the reinforcements received by all participating agents are maintained in a shared context. This shared context ensures that experiences of each participant complement the others in the community in an implicit manner. Any improvement related to implicit coordination could be detected by comparing results of this experiment with experiment 1 scenario 2.

6.3.4.1 Scenario 1: 2-Agent Cooperative

In this scenario, two agents were released concurrently to the grid environment. The agents make sure they do not come into overlapping states concurrently. A grid cell locking mechanism was used for this purpose. It was observed in comparison to experiment 1 scenario 2 that the number of state transitions within an episode reduces as the two agents complement each other through the shared context of reinforcements and the agents converged to an optimal path in a relatively lesser number of episodes (Figure 6.8). Hence, the rewards given to one agent has an effect on the other, despite a lack of direct communication among the participants. A satisfactory level of improvement was gained with respect to a single agent contribution.

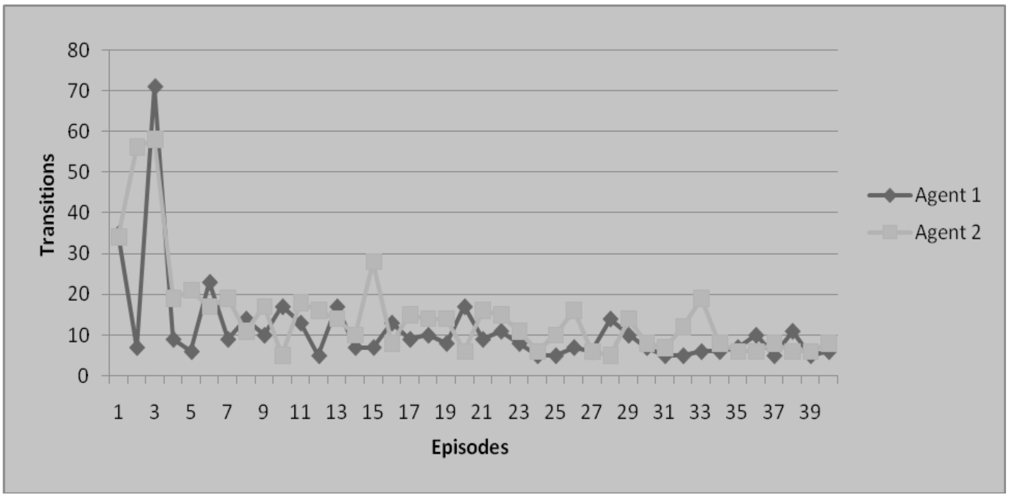


Figure 6.8: State transitions of two agents in a cooperative mode

The rewards distribution depicted in Figure 6.9 shows convergence to the goal state in a gradual manner with more exploration considered than the single agent scenario of experiment 1. This level of reward distribution is very important to overcome situations of getting stuck in local optima.

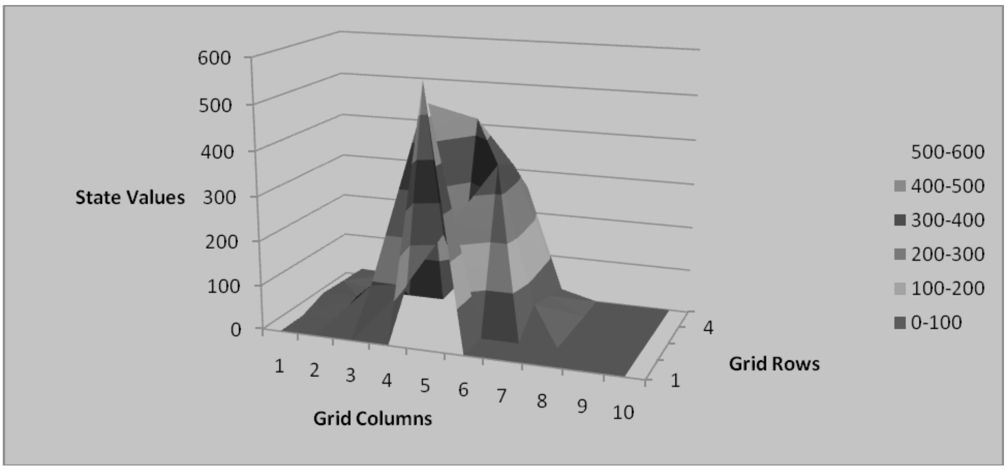


Figure 6.9: Reward distribution in a 10 x 5 grid world based on activities of two agent cooperative scenario

However, this approach has more tendencies to create local optima when compared to the approach taken in experiment 1 scenario 2. This is evident from the comparison of reward distribution of the two approaches. However, there is clear improvement when compared to experiment 1 due to the low number of episodes taken to converge to the optimal path.

6.3.4.2 Scenario 2: 4-Agent Cooperative

This experiment is very similar to scenario 1 of experiment 2, with the difference being the use of 4 agents instead of 2. The assumption being that due to increase in concurrent search capabilities, the number of episodes to reach the optimum path should be improved.

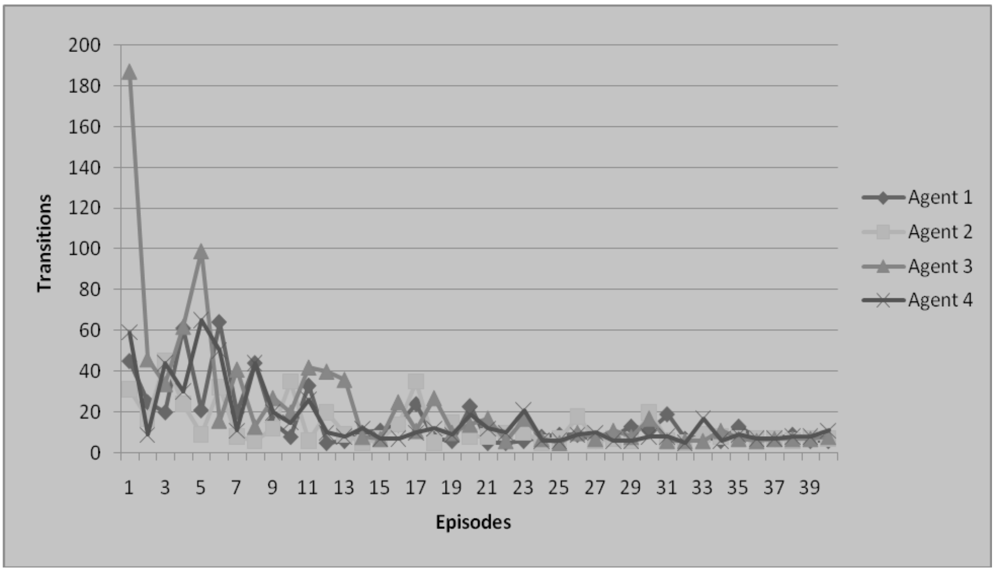


Figure 6.10: State transitions of 4 agents in a non-cooperative mode

The assumption was justified when comparing data depicted in Figures 6.10 and 6.11 where the two agent scenario achieves the optimum solution after an average of 40 episodes and the four agent scenario after 35 episodes on average. However, the proportion of increase in agents does not correlated to the decrease in the episodes for convergence.

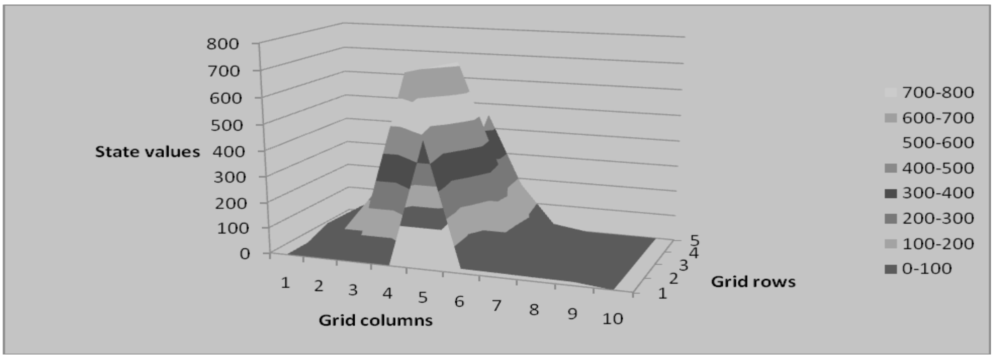


Figure 6.11: Reward distribution among 10 x 5 grid world after activities of 4 agent cooperative scenarios

The comparison of reward distribution values in Figures 6.10 and 6.11 reveals the presence of more exploration activities in scenario 1 than in scenario 2. Therefore, though scenario 2 achieves the optimum path in lesser iterations, there would be less time spent on exploration as the number of agents increase. Logically, it agrees to the fact that when the amount of agents increases, each agent tends to use guidance of others and spend less time on exploration. Therefore, these two experiments suggest that there should be a proper balance of the number of agents based on the complexity of the environment.

6.3.5 Grid Experiment 3 – Collective Foraging Based on the AAANTS Model

The final set of experiments was conducted to evaluate the capabilities of the AAANTS model when compared to the Monte-Carlo method. As discussed in chapter 3, AAANTS model uses a combination of concepts based on emergence heuristics, ATs, behavioural concentres, hub states, temporal sensory frames and reinforcement learning methods. The expectation is that the learning outcome of the AAANTS model should out perform the traditional methods such as the Monte-Carlo method. The experimental variables were kept constant across experiments other than the specific aspects of the algorithm unique to the AAANTS model.

6.3.5.1 AAANTS Experimental Guidelines

This experiment was conducted along the following guidelines to align with the objectives of the research.

1. The AAANTS model uses a learning strategy based on reinforcement learning (Chapter 4) to implement the adaptive nature of agents. The MC method used in the previous two control experiments was adjusted as discussed in chapter 4 to suit the needs of the AAANTS model. The episodic rewards generated were distributed disproportionately among the participating states and were assigned in descending order from the goal to home location – similar to the varying concentrations of pheromones in insects. The state values are modelled in a shared context accessible to all participating agents of the grid world.

2. The agents perform state transitions in the grid world by evaluating the best possible state from the four neighbours. This is called as 1-step look-ahead and was the basis for all the prior control experiments conducted. The AAANTS model introduces the concept of ATs where there could be several actions executed and reinforced together by a community of agents. Hence, depending on the number of actions in the AT the agents could do multi-step look-ahead and each of these actions within an AT is contributed by the coordination of several agents. This experiment uses templates of different number of actions; therefore the agents perform state value evaluation based on the number of actions in the template. For example, if there are 2 actions in the AT, an agent performs two-step look-ahead of all possible states that it could reach and based on the highest possible reward expectation executes the two actions in sequence to reach the next state.

3. The AAANTS model uses the concept of Hubs as described in Chapter 3. Hubs are highly connected states that amalgamate heterogeneous regions. The reward sharing model of agents identifies the Hub states from the rest of the states and uses them to converge to the optimum path. The hubs are demarcated by states that are relatively high in reward value and also could be somewhat similar to local optima states. Hence, the local optima states in the AAANTS model could be used to improve the experimental outcomes. The Hub states could be hierarchically arranged as depicted in figure 4.2 based on the rewards from the environment.

4. These two experiments would demonstrate the capabilities of the AAANTS model in relation to the emergent nature of behaviour. Though there were several agents in experiment 2, they were only coordinated through implicit coordination to individually achieve goals whilst sharing information of the shared context. They did not coordinate the movement of a single entity from home to destination. The AAANTS model based experiments uses the multiple agents to coordinate the actions of the ATs that belong to a single entity that moves within the grid.

5. An agent could sense the environment with respect to each residing cell within the grid. The listed experiments have considered a 3 x 3 matrix of cells surrounding and including the inhabiting cell of an agent. Five clustering centres were selected as the ideal after initial series of experiments. Each cell has two state values, one related to reaching the food source and the other for reaching the home while carrying food. The data values of the 3 x 3 matrix are used to compose a Temporal Sensory Frame unique to each location in the grid (discussed in section 3.4). The use of TSF for navigation is only used in experiment 3 which is based on the AAANTS model.

6. The ATs, sensory templates and agent instances assigned to ATs are configured using the CDT components of the AAANTS framework. First, the list of AAs was defined for the grid-world experiment. The four movements that were identified as important to this experiment were up, down, left and right. Thereafter, altogether eight ATs were defined. Scenario 1 used six ATs using four agent instances and scenario 2 used all eight actions

using same number of agent instances. The agent instances were also defined using the CDT and activated using the SMVC. The agents migrate from inactive to active state (described in section 3.2 using figure 3.1) when instructed by the SMVC and thereafter becomes autonomous until optimum path is reached. After reaching the optimum path, the agents converge to a static path, especially after exploration probabilities gracefully degrade. However, SMVC has the capability to alter the exploration and exploitation blend even during the experiments. Further, all the experimental data are captured through the SMVC.

6.3.5.2 Scenario 1: 4-Agents Using an Action Template of 2 Actions with 6 ATs

This experiment uses an AT with 2 elementary actions each action with 4 possible movements. Each of these actions are coordinated by an agent, hence the experiment require four agents to move a search node from source to destination since a single agent is responsible for a particular type of action. Six instances of ATs were created from the use of AAs related to Forward, Backward, Left and Right. These instances were {Forward, Forward}, {Forward, Left}, {Forward, Right}, {Backward, Left}, {Backward, Right} and {Backward, Backward}. The rest of the experimental method is as per the guidelines listed in section 6.3.5.1. However, the overall guidelines listed in section 6.3.2 are applicable to both scenarios of experiment 3.

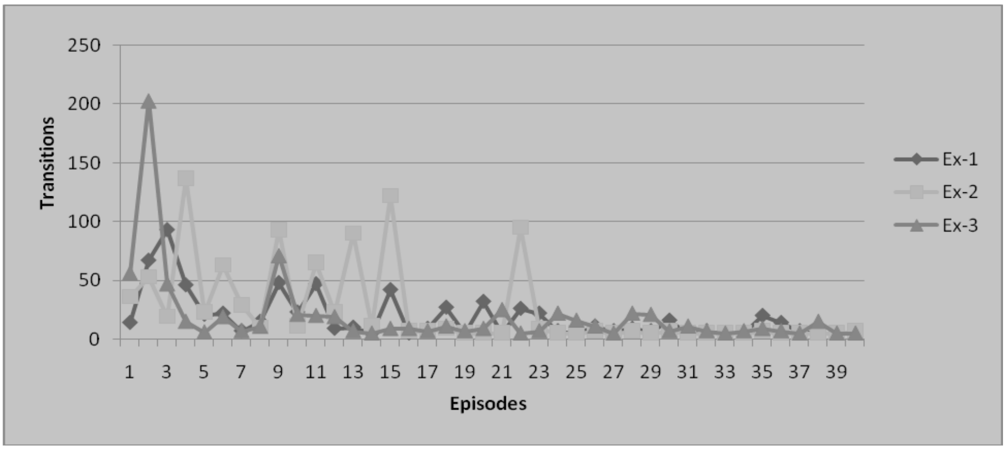


Figure 6.12: State transitions of 2-agent cooperative scenarios across 3 experiments with an action template with one action

A considerable improvement was gained in this experiment with reference to the MC based control experiments (experiments 1 & 2). Three iterations of the same experiment were conducted to gain a general consensus of the approach where all three experiments demonstrated a relatively similar optimisation pattern (Figure 6.12).

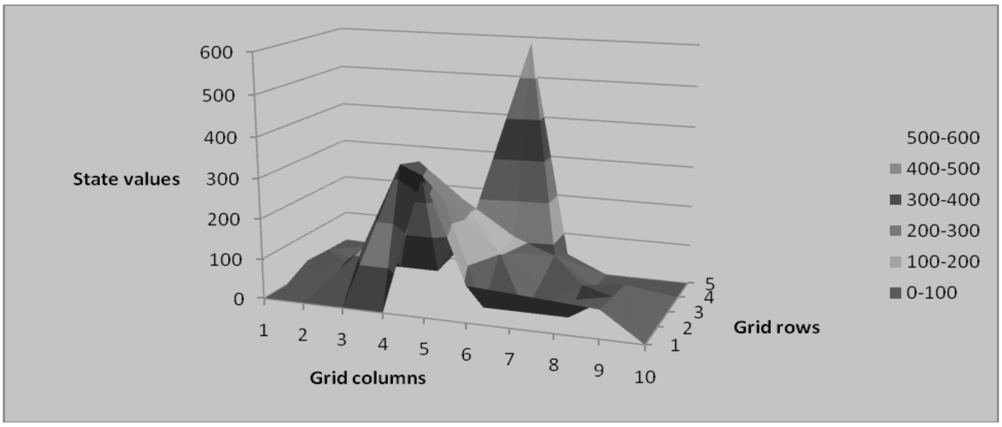


Figure 6.13: State value distribution among 10 x 5 grid world after activities of 2 agent cooperative scenarios with action template with one action

The use of hubs becomes very obvious with reference to reward distribution in figure 6.13. There are two state values peaks in the graph and the secondary peak with relatively lower value concentration could be considered as a hub state. It could also be mistaken as local optima based on the slope towards the goal state but based on the arrangement of the obstacles the secondary peak serves as a guiding path, technically a hub state within the grid. It was noticed that even when obstacles were rearranged, the position of the hub state changes accordingly to connect heterogeneous regions separated by the obstacles.

6.3.5.3 Scenario 2: 4-Agents Using an Action Template of 2 Actions with 8 ATs

This experiment uses an AT with 2 elementary actions each action with 4 possible type of movements. Each of these actions are coordinated by an agent, hence the experiment require 4 agents to move a search node from source to destination since a single agent is responsible for a particular type of action. Eight instances of ATs were created from the use of AAs related to Forward, Backward, Left and Right. These instances were {Forward, Forward}, {Forward, Left}, {Forward, Right}, {Backward, Left}, {Backward, Right}, {Backward, Backward}, {Left, Left} and {Right, Right}.

The rest of the experimental method is as per the guidelines listed in section 6.3.5.1. However, the overall guidelines listed in section 6.3.2 are applicable to both scenarios of experiment 3. The objective of this scenario is to investigate whether there is an improvement due to the increase in the available possibilities of movement (related to increase of ATs from six to eight). It was observed that the number of episodes to reach the optimum path was reduced further in relation to the results in scenario 1 (Figure 6.14).

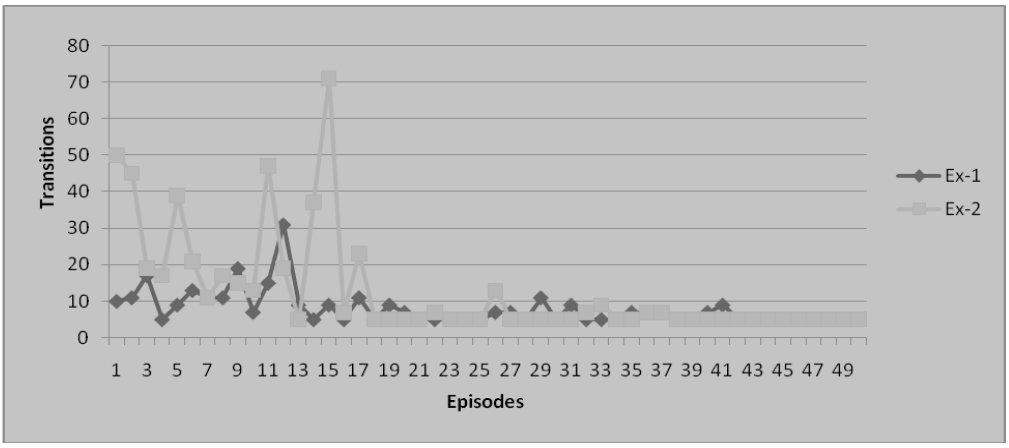


Figure 6.14: State transitions of 2-agent cooperative scenarios across 2 experiments with action template with two actions

The important finding in this experiment was the presence of more hubs in relation to scenario 1 (Figure 6.15). Based on these experimental results it should be noted that the presence of hubs (local optima) contributes towards better convergence when executing the AAANTS algorithm whereas in the previous experiments, the local optima resulted in sub-optimal results.

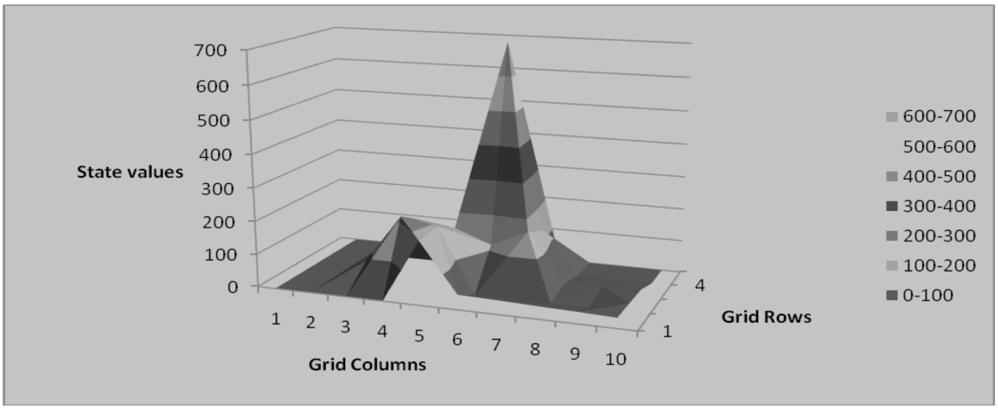


Figure 6.15: State value distribution among 10 x 5 grid world after activities of 2 agent cooperative scenarios with action template with two actions

According to Amy McGovern *et al* [MCGO01], a bottleneck was described as a region in the observation space that an agent tends to visit frequently on successful paths to the goal. It was identified that an option framework could be used to define sub-goals where an option is a temporarily extended action which, when selected by an agent, executes until a termination condition is satisfied [MCGO01]. The use of hubs within the AAANTS model could be directly attributed to the concept of bottlenecks that are defined as sub-goals used in reaching the goal.

6.4 Robotic Arm Experiment

The objective of this experiment is to evaluate whether a static layer of innate behaviour could result in heterogeneous emergent behaviour. The experiment is designed to create two types of behaviour from a robotic arm with 3 joints by using the same set of innate ATs. One expected behaviour is the grabbing of an object and the other is to push an object which is kept at the same distance from the base of the arm.

6.4.1 Experimental Guidelines

The guidelines of the experiment are described below.

1. The robotic arm in the resting mode is stretched at an angle of 90 degrees to the target object.

2. The elbow and wrist joint is restricted to rotate 25 degrees with respect to the resting angle. The rationale was that an angle over this limit contributes to practical movement issues in relation to the robotic arm installation.

3. The state transition of the grid world from goal to the destination is mapped in a similar sense in this experiment. The wrist of the robotic arm is considered as the moving target similar to the movement within a grid from one cell to another. The state that should be given the reward is based on the cells that overlap with the wrist joint while executing a particular AT. An example of a specific AT is depicted in Figure 6.16.

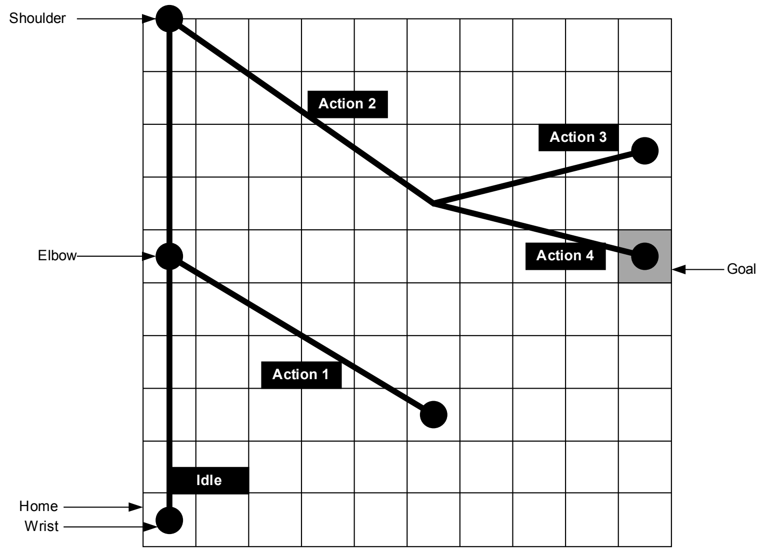


Figure 6.16: Robotic arm AT movement with reference to the grid state transitions

4. There are four types of elementary actions that could be blended to achieve the goal state.

   a. Shoulder move – positive degrees (up)

   b. Elbow move – positive degrees (up)

   c. Elbow move – negative degrees (down)

   d. Wrist move – close to grad the object.

5. Eight AT instances were created using four elementary actions for each AT. Out of these ATs some are related to grabbing the object and the rest related to pushing the object.

6. Credit / Reward assignment was done subsequent to executing the four actions in each AT instance. The credit assignment is implemented in relation to the discussion in section 4.5.2.

6.4.2 Experiment Results

The expected result of the experiment is related to finding the convergence to a solution rather than finding the optimum movement to grab the object. The convergence to a solution, which is related to ultimately been able to both grab and push the target object would justify the capability of a static innate layer of elementary actions producing heterogeneous emergent behaviour.

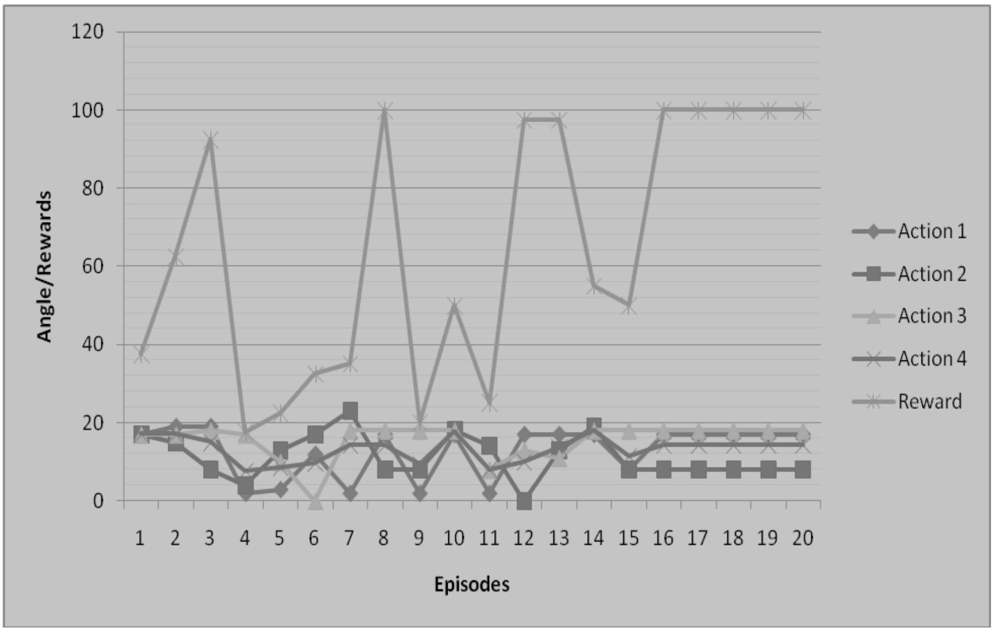


Figure 6.17: Results of the object grabbing experiment – scenario 1

The scenario 1 of the robotic arm experiments converges to grab the target object within an average of 16 episodes with reference to the figure 6.17. Relatively high values of rewards were given to the

ATs that resulted in grabbing the object as opposed to pushing the object. The overall outcome was inline with the reinforcements and in all experimental iterations the solution converged to the grabbing of the object within 16 episodes.

The scenario 2 of the experiment was conducted on the same innate layer of behaviour (8 ATs), but the reinforcements were aligned to the pushing of the object. The arm movement converged to the optimum behaviour of pushing the object as opposed to grabbing the object within an average of 17 episodes (Figure 6.18).

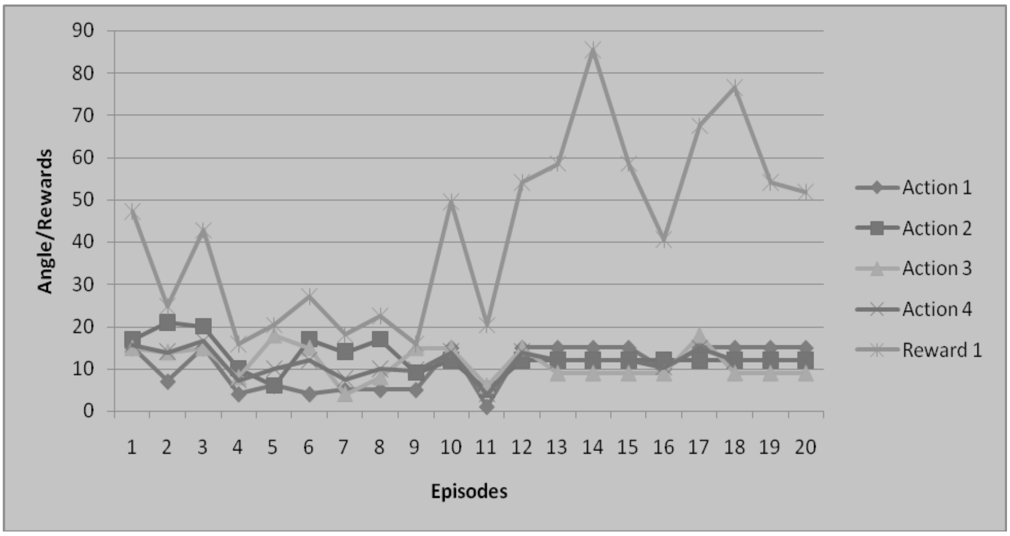


Figure 6.18: Results of the object pushing experiment – Scenario 2

The experimental results confirm the ability of a static innate layer to produce heterogeneous behaviour based on the reinforcements from the environment. The state values of the matrix described in 6.4.1-item-3, could be used to analyse the state values after converging to the optimal solution. It was noticed that the local optima of the state value show unique patterns for each respective movement.

6.5 Overall Experimental Observations and Conclusions

The following sections are dedicated to summarise the observations of all the experiments conducted within this chapter.

6.5.1 Observations of the Grid World Experiments

The following observations of the grid world experiments were identified as important to assess the hypothesis and objectives of this research.

1. When comparing results of experiment 1, scenarios 1 & 2, it is evident that disproportionate distribution of rewards among state values results in better convergence to the optimum path (Figure 6.19). The disproportionate distribution is analogous to pheromone distribution of insects where the concentration is maintained in an ascending rate when reaching the goal state. Even after changing the location of the goals and obstacles in scenario 2, the algorithm was able to readjust the state values to converge to the new path within a reasonable number of episodes.

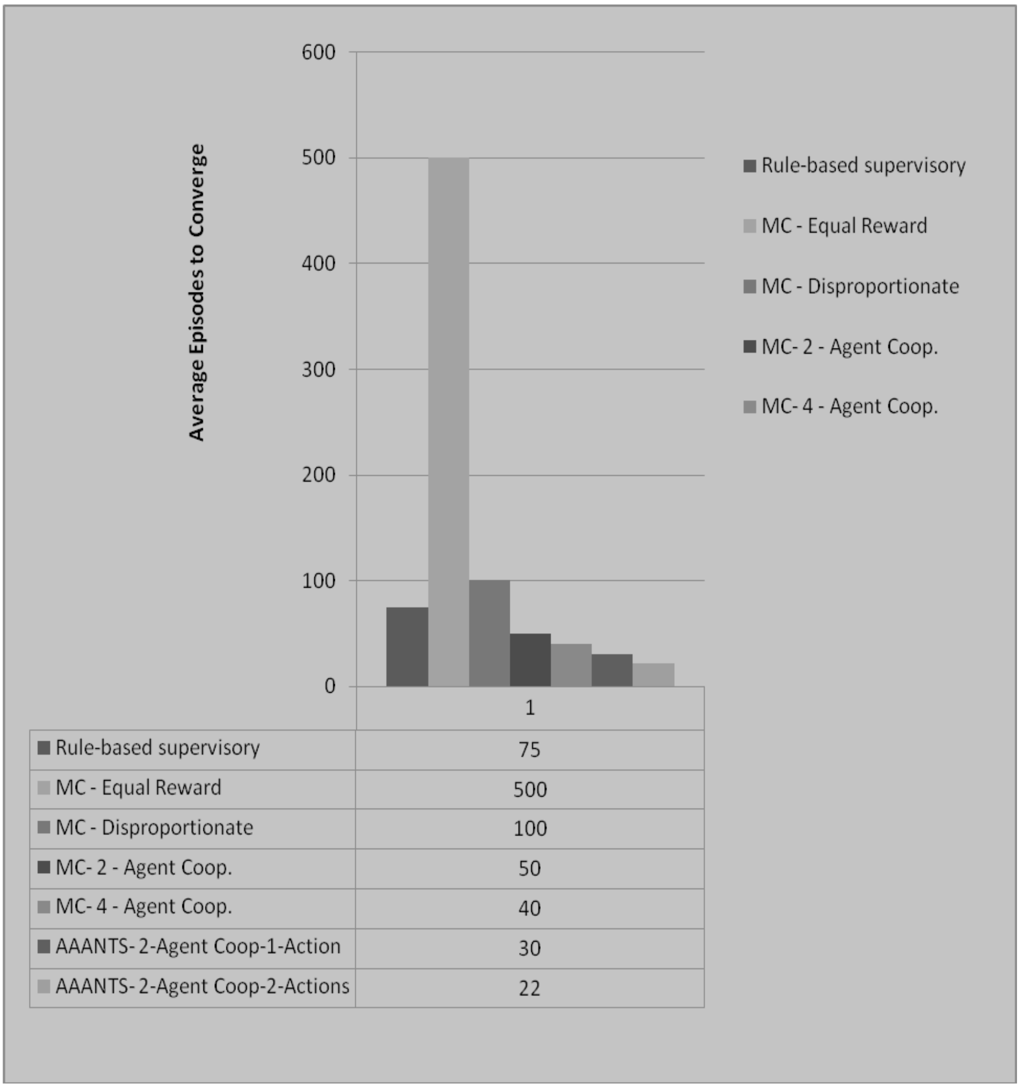


Figure 6.19: Comparison of average episodes taken to converge to the optimum path using the different learning strategies discussed

2. The objective of the experiment 2 is to evaluate the effectiveness of implicit coordination methods using shared contexts on general learning algorithms such as Monte-Carlo. Both scenarios of experiment 2 showed improvements when compared to the results of experiment 1, which the latter is void of any form of coordination. However, several more experiments were carried out with increased agent counts from one to ten. It was noticed that

initial gradual improvements fade away after reaching an optimum threshold of agents which was however variable based on the grid sizes.

3. Among all the Monte-Carlo based experiments (experiments 1 & 2), the 4-agent cooperative method produced the best outcome (Figure 6.19). This was a modification done to the original Monte-Carlo method to include the cooperative aspects with the objective that it could be compared in similar grounds with the AAANTS model.

4. Experiment 3 introduces the full scale features of the AAANTS model. It introduces the capabilities of emergence, innateness and implicit communication. In experiment 3 a key difference when compared to experiments 1 & 2, is that though there are multiple agents, there exists only one search thread at a time. The multiple agents coordinate different elementary actions of the AT to navigate a single search node from source to destination. An AT is executed based on inputs from the environment and each elementary action is contributed by a single agent.

5. The results of the experiment 3 out perform that of experiments 1 & 2, and further demonstrate that capabilities improve when the innate layer contribute several ATs to survive in the environment. Most suitable AT needs to be selected based on the sensation from the environment.

6. Further, it was noticed that when the amount of obstacles were increased within the grid world, the AAANTS method

converges considerably faster than the Monte-Carlo methods. This was due to the fact that AAANTS uses obstacle characteristics as navigation markers during the initial exploration process. These obstacles were described as local-optima and specifically within the AAANTS model referred to as Hubs – special states that bridges regions of cells. For example, when there is a pattern of receiving high reward for moving forward when a certain type of obstacle is in the neighbourhood, the agents detects these situations as Hubs and adapts to executing the appropriate AT whenever such situations were faced.

Observations/ Experiments	Ex 1 – Sc 1	Ex 1 – Sc 2	Ex 2 – Sc 1	Ex 2 – Sc 2	Ex 3 – Sc 1	Ex 3 – Sc 2
Average number of states of the optimum path from source to destination	15	10	9	9	8	8
Presence of local optima	Yes	Yes (relatively low)	Yes	Yes	Yes	Yes
Stability after converging to the optimal path	No	Mostly	Mostly	Mostly	Yes	Yes
Ability to reach the optimal path	No	Mostly	Mostly	Mostly	Yes	Yes
Minimum number of episodes to converge to optimum path	> 100	50-100	30-50	27-40	25-30	20-22
Ability to converge after adjusting the location of the goal subsequent to reaching convergence	No	Mostly	Mostly	Mostly	Yes	Yes
Ability to converge after adjusting obstacle arrangement subsequent to reaching convergence	No	Mostly	Mostly	Mostly	Yes	Yes

Table 6.3: Observation summary of the grid world experiments

7. The summary of the experimental outcomes of all the experiments of the grid world domain is tabulated in Table

6.3. It could be stated that the number of episodes to converge and states to reach the goal state considerably reduces in the AAANTS domain. The final outcome is very stable in the AAANTS model when compared to the rest of the control experiments.

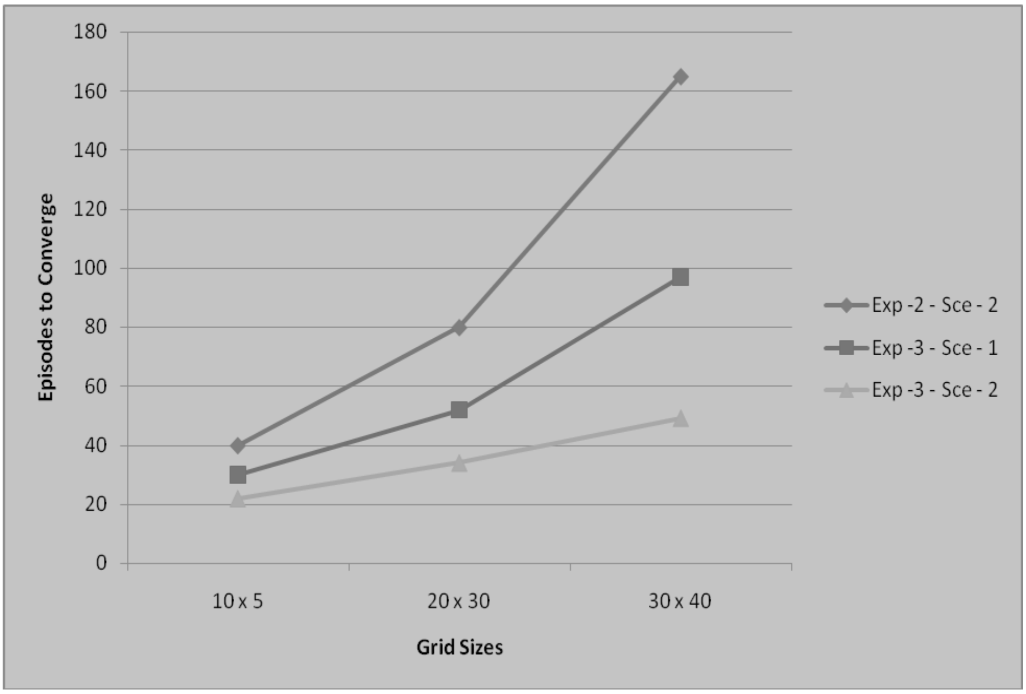


Figure 6.20: Comparison of overall average episodes to converge in extended grid search spaces

8. Figure 6.20 depicts the results of experiments conducted on extended search spaces of 20 x 30 and 30 x 40 grid sizes. The experiment 2-4 agent scenario was taken to represent the MC learning method, which is actually the best performing out of all the MC experiments. The MC method does show convergence to an optimal path, however, the overall number of episodes increases considerably when compared to the AAANTS learning

model. Both experiments related to the AAANTS learning model show superiority in comparison to MC method. Out of the two AAANTS experiments, the method which contained higher number of ATs seems to converge with a relative lower number of episodes and further the ratio of increase is lower. It could be concluded that the AAANTS learning model scales better in complex environments when compared to the MC method. The experiments conducted on the same grid sizes with increased number of obstacles demonstrated even better results in favour of the AAANTS model in comparison to the MC method.

6.5.2 Observations of the Robotic Arm Experiment

As discussed earlier the objective of this experiment is to evaluate whether a static innate actions could result in heterogeneous emergent behaviour. It was observed that both the object grabbing and pushing behaviour emerged from the 8 ATs programmed in the innate layer.

Observations/ Experiments	Scenario 1	Scenario 2
Presence of local optima	Yes	Yes
Stability after converging to the optimal path	Yes	Yes
Ability to reach the optimal path	Yes	Yes
Minimum number of episodes to converge to optimum path	16	17
Ability to converge after adjusting the location of the goal subsequent to reaching convergence	Yes	Yes

Table 6.4: Observation summary of the robotic arm movement experiments

The two scenarios conducted converged to the expected emergent behaviour and it was evident that based on the reinforcements from the environment an innate layer could produce heterogeneous emergent behaviour. The summary of the overall experimental results are tabulated in Table 6.4.

6.6 Chapter Summary

This chapter provides an in-depth description to the experiments conducted within the AAANTS research. The experiments span across two domains with the objective of evaluating the abstractness of the proposed AAANTS learning model. It was evident from the results that in all experiment instances, the proposed model shows early convergence when compared to traditional learning methods such as the Monte-Carlo method.

The grid world experiment was conducted using a simulator program that could be configured for different grid sizes, obstacles arrangements and number of agents. The initial experiments were conducted on a 10 x 5 grid to shorten the experimental periods, but thereafter modelled on 20 x 30 and 30 x 40 grid sizes to detect any anomalies related to the increase in environmental complexity. Three different varieties of experiments were carried out to evaluate the objectives of the research.

The robotic arm experiment was conducted by an actual robotic arm built using a robotic kit. The 3 joint robotic arm was fixed at the shoulder joint and the movement of the overall arm happen in 4 step movements to either grab or push the target object. The experiment was conducted using 8 ATs which contributes to two types of behaviour. The experiments demonstrate the capabilities of the model in achieving heterogeneous behaviour using an innate layer of behaviour.

Chapter 7

Research Conclusions

7.1 Introduction

The purpose of an intelligent entity is to provide the most appropriate and preferred behaviour to the continuous flow of multi-modal sensations from the environment. Decades of research have contributed a breadth of strategies, methodologies, models and theories to this science of creating artificial intelligence. However, the world still awaits a major breakthrough in artificial intelligence research that could produce anthropomorphic levels of intelligence. The best avenue to pick up any clues for research directions in artificial intelligence is most certainly where it started in the first place - the animal kingdom. The research was inspired several years back by investigating the amazing world of ants. Henceforth, the research unravelled itself in several directions, though not deviating from the core objective of achieving emergent behaviour as a result of collective implicit interactions of simple entities.

7.2 Research Approach Summary

The study conducted during the initial phase of the research related to intelligent systems that are both natural and artificial in nature instilled the identification of three aspects that are paramount for realising intelligence. They are coordination, adaptability and representation, which would be referred to as the AI Mix. The first agenda of the research was to build models of these three aspects in amalgamation would result in, a holistic model that could be used to test the hypothesis of this research.

The conceptualised holistic model is referred to as the AAANTS model.

A considerable amount of time was spent on evaluating existing agent platforms that could favourably implement the AAANTS model. However, after much deliberation it was decided to implement a generic agent platform to realise the AAANTS model, because the existing platforms lacked key requirements of the model. The resulting agent implementation was called as the AAANTS platform.

The selection of experimental domains was based on the objectives of the research. The basic requirement of the domain was the ability to produce behaviour that is emergent and complex. Based on these requirements Grid World Foraging, Robotic Arm movement to grab an object and Visual Navigation of a Robotic vehicle were selected as the prospects. Implementation for all three domains were conducted, however, during the final stages of the experiments, the Visual Navigation experiment was discontinued since it was difficult to compare its results with the other two domains.

7.3 Research Conclusions

The conceptualisation of the AAANTS model was based on the hypothesis and the objectives mentioned in the introductory chapter. The model was driven by the concepts of innateness, adaptability, emergence, implicit communication and behavioural congruence. The initial inspiration of the insect colonies motivated the author to build an artificial model that incorporated the above ingredients to reach different grades of intelligence that could be deployed on heterogeneous problem domains.

The essence of emergence is that none of the contributors to the emergent behaviour is aware of the master plan. The grid world experiments 1 and 2 is void of any form of emergence, however, it shows gradual improvements (within the 4 scenarios of experiments 1 and 2) related to the use of shared contexts and implicit communication among the participants. However the grid world experiment 3 focuses on the emergent nature of behaviour with the introduction of the full functionalities of the AAANTS model. The AAANTS model demonstrates considerable improvement over the standard Monte Carlo technique and specially performs exceptionally better in larger grid sizes. Further, it is concluded that dynamic changes in the environments (goal and obstacle location changes) are gracefully handled by the AAANTS model in comparison to the Monte-Carlo learning model. These observations confirm the achievement of congruent behaviour in dynamic environments using the concept of the AAANTS model.

The grid world experiments confirm that the behavioural acts built, based on innate action templates provide better convergence to the optimum behaviour than using a pure adaptation strategy void of innate behaviour, which thereby confirm the respective objective set forth in the introduction. The purely adaptive experiments, especially the grid world simulation, demonstrates that the simulations conducted void of action templates takes relatively more episodes to converge to the optimum path and further intermittently settle down on local optima.

The argument of achieving different grades of intelligence using a static layer of innateness is another objective of this research. The robotic arm experiment was dedicated to evaluate this hypothesis. This experiment solved both problems of object pushing and grabbing using a static collection of ATs. Hence, it could be concluded that innate ATs could be reinforced and adapted to produce heterogeneous emergent behaviour. However, it should be noted that the possibilities of behaviour would be restricted within the capabilities of the innate ATs.

7.4 Future Work

The AAANTS model and the experiments were focused on the fulfilment of the hypothesis and the objectives of the research. A large collection of complementary methods and concepts resulted as a by-product of this research. These aspects that could be used to further improve the capabilities of the AAANTS model is discussed in this section. The author intends to conduct further experiments to improve the capabilities of the AAANTS model and hope to test the capabilities in a more complex problem domain of vision navigation.

One of the highlights of the model is the capabilities in uniquely identifying a sensation and situations using TSFs based on sensory hubs. However, the experimented sensations were primitive in nature and may need further enhancements to apply to a complex sensory modality such as vision. One of the methods of improving this technique is with the proper identification of clustering centres. The ant colony optimisation algorithms

[KANA03a] [KANA03b] are one of the most successful methods that could overcome FCM algorithm's sensitivity to the initial values of clustering centres. The clustering centres filtered through Ant Optimisation Algorithms could be refined using the FCM algorithms. However, the AAANTS model did not pursue this path in order to keep the FCM algorithm less complicated to suit the generic pattern recognition needs of this research. Hence, this could be taken as a method of refining the overall research outcome in the future.

The aspect of building relationships among multi-modal sensory frames is not pursued in this research. This is an aspect that could be researched further to gain insight to the integration of multi-modal sensations. This is similar to the concept of Perceptual Integration [COEN00], where perception layers from multiple modalities are integrated into a holistic abstraction. The McGurk effect [COEN00] is perhaps the most convincing demonstration of the inter-sensory integration where one modality radically changes perceptions in another through perceptual integration. Hence, the integration of multi-modal sensations to identify situations is a useful direction to improve the capabilities of the AAANTS model.

In this research the agents within a colony were defined as cooperative. However, some level of competition among the agents to resolve the overlapping nature of elementary behaviour may produce better results. This would be a similar concept to the cross-exclusion concept [MINS86] which could be used to regulate levels of activities in an agent society. When an agent in a group, contributing to related behaviour is aroused, its signals tend to inhibit others in direct competition. This leads to an avalanche effect, as each competitor grows weaker; its ability to inhibit its challengers also weakens. The result is that even if the initial difference between competitors were small, the most active agent would quickly lock out all the others. Consequently, the most suitable action template or elementary action could be selected by the agents.

With respect to the AAANTS model, agents have the capability to directly interface with the sensations relevant to their role using the sensory bus. However, according to the society of mind theory by Marvin Minsky [MINS86], only a minority of agents are connected directly to the sensors of the outer world, like those that send signals from the eye or skin; most of the agents in the brain detect the events inside the brain. Further, according to Rodney Brooks [BROO92], as much as 50% of the human brain seems to be devoted to perception, the rest could be assumed to play a coordination role with the rest of the community. The AAANTS model could be configured to dedicate a segment of agents for perceptive activities and the rest to be dependant only on inter-agent communication for coordination. This hypothesis could be tested to evaluate whether it provides improved results in comparison to the already conducted experiments.

Common sense reasoning is an attractive domain of research that may have applications to the AAANTS model. The identification of common situations using TSFs and its relationships to common behaviour could be extended to represent a common-sense reasoning model. The author intends to model this aspect using the vision navigation domain of experiments to demonstrate similar behaviour to common situations.

The pattern matching methods applied to identify sensory patterns within the current model could be considered simple and sensitive to noise. More advanced techniques of pattern matching such as frequency analysis using Fourier series and matrix manipulation could contribute in producing more accurate pattern matching models.

It would be a time consuming process to compare the outcomes of the AAANTS model with the myriad of learning models. The Monte Carlo based reinforcement learning method was selected as the representative for the rest of the algorithms due to the reasons stated in the experiment chapter. However, it would be useful to compare these results to other learning models such as Temporal Difference, Neural networks, belief networks, etc., for future research activities. The author intends to publish these results subsequent to the acceptance of this dissertation.

The experiments were made complex by introducing aspects such as obstacles to the grid world environment. However, the environment could be enhanced to further complicate and challenge the capabilities of the AAANTS model. The collective transportation of food elements is one such agenda. The change to the existing food transportation process could be explained as follows. An agent has a limitation to the food chunk size it could transport which could be referred as the Agent Chunk Size. Each agent arriving at a food source should first calculate the chunk size of the food source. If the "agent chunk size" >= "food chunk size"; then the agent could transport food, a chunk at a time. When "agent chunk size" < "food chunk size", a single agent does not have the capacity to carry a food chunk on its own which requires collective effort. The correct number of agents should be grouped to carry out the food transportation. Hence, the important aspect is the calculation of the exact number of agents that is needed to transport a food chunk and the coordination of the selected group in reaching the nest collectively.

Finally, the AAANTS model discussed action templates as fully produced through the innate layer to make the model less complicated to experiment. However, action templates could be produced by the adaptive layer using a stochastic selection of elementary actions. The produced ATs could be selected or destroyed based on the reinforcements from the environment. This process would enable an artificial entity to produce emergent behaviour that are even absent in the innate layer action templates. This is an aspect identified as an enhancement to the initial model of AAANTS.

The above research directions could be identified as separate projects as extensions to the AAANTS model. The AAANTS model could be applied to more complex AI problem domains by incorporating the above mentioned modifications. The author sincerely believes that anthropomorphic behaviour in an artificial entity could be achieved by biologically inspired AI models.

References

Index	Description
[ABON02]	Janos Abonyi, Sandor Migaly, & Ferenc Szeifert, 2002, Fuzzy self-organising map based on regularised fuzzy c-means clustering, Published in advances in soft computing, engineering design and manufacturing, Springer Engineering series.
[ALEX96]	Alexandros Moukas, 1996, Amalthaea: Information Discovery and Filtering using a Multi-agent Evolving Eco System, MIT Media Laboratory.
[ANDE01]	Niels S. Andersen, Mathias N Kjaegaard, 2001, Advanced Programming of the LEGO RCX for Education, Technical University of Denmark, Lyngby.
[ANDR03]	Virgil Andronache, Matthis Scheutz, 2003, Artificial Intelligence and Robotics Laboratory, University of Notre Dame, USA.
[ARAI00]	Sachio Arai, Katia Sycara, 2000, Effective Learning Approach for Planning and Scheduling in Multi-Agent Domain, Robotics Institute, Carnegie Mellon University.
[AYLE98]	Ruth Aylett, Frances Brazier, Nick Jennings, Michael Luck, Hyacinth Nwana and Chris Preist, 1998, Agent Systems and Applications, Result of a panel discussion at the Second UK Workshop on foundations of Multi-Agent Systems (FoMAS'97)
[BABA01]	Ozalp Babaoglu, Hein Meling, Alberto Montresor, 2001, Anthill: A Framework for the development agent-based peer-to-peer systems, University of Bologna, Italy, Norwegian University of Science and Technology, Norway.
[BADI04]	Costin Badica, 2004, Manufacturing and control: putting agents to work, University of Craiova.
[BAKE99]	Albert Baker, Van Dyke Parunak, Kutluhan Erol, 1999, Agents and the Internet: Infrastructure for Mass Customisation.
[BALC97]	Tucker Balch, 1997, Learning roles: Behavioural Diversity in Robot Teams,

Index	Description
	Mobile Robot Laboratory, College of Computing, Georgia Institute of technology, Atlanta Georgia.
[BARA02]	Albert-Laszlo Barabasi, 2002, Linked, How everything is connected to everything else and what it means for business, science, and everyday life, Published by the Penguin Group, ISBN: 0-452-28459-2.
[BARA03]	Albert-Laszlo Barabasi and Eric Bonabeau, 2003, Scale-Free Networks, Scientific American.
[BART94]	Andrew Barto, Michael Duff, 1994, Monte Carlo Matrix Inversion and Reinforcement Leaning, Computer Science Department, University of Massachusetts, USA.
[BAUM00]	C. Baumer, M. Breugst, S. Choy, T. Magadanz, 2000, Grasshoppe – A Univeral agent platform based on OMG MASIF and FIPA standards.
[BAUM99]	C. Baumer and T. Magedanz, 1999, The Grasshopper Mobile Agent Platform Enabling Short term Active Broadband Intelligent Network Implementation
[BECK94]	R. Beckers, O.E.Holland, J.L.Deneubourg, 1994, From local actions to global tasks: Stigmergy and collective robotics, Center for interdisciplinary research, University of Bielefield.
[BEER99]	Martin Beer, Mark d'Inverno, Michael Luck, Nick Jennings, Chris Preist, and Michael Schroeder, 1999, Negotiations in Multi-Agent Systems, A report as a result of a panel discussion at the Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS'98)
[BELL03]	F. Bellifemine, G. Poggi, G.Rimassa, 2003, JADE A white Paper, Telecom Italia Lab.
[BELL99]	F. Bellifemine, A.Poggi, G.Rimassa, P Turci, 1999, An Object-Oriented Framework to realise Agent Systems.
[BOGN99]	Myles Brandon Bogner, 1999, Realising "Consciousness" in Software Agents, PhD Dissertation, The University of Memphis.
[BRAD97]	Jeffery M. Bradshaw, 1997, Software Agents, ISBN: 0262522349.
[BROO92]	Rodney A. Brooks, 1992, Artificial Life and Real Robots, MIT Artificial Intelligence Laboratory, USA.

Index	Description
[BURB04]	Kalle Burbeck, Daniel Garpe,Simin Nadjm-Tehrani, 2004, Scale-up and performance studies of three agent platforms.
[BUSE99]	Paolo Busetta, Ralph Ronnquist, Andrew Hodgson, Andrew Lucas, 1999, Light-Weight Intelligent Software Agents in Simulation.
[CAO95]	Y. Uny Cao, Alex S. Fukunaga, Andrew B. Kahng, and Frank Meng, 1995, Cooperative Mobile Robotics, UCLA Computer Science Department Los Angeles, CA.
[CASS95]	Anthony R Cassandra, 1995, Optimal Policies for Partially Observable Markov Decision Processes, Department Computer Science, Brown University, Providence, RI 02912.
[CAWS97]	Alison Cawsey, 1997, The Essence of Artificial Intelligence, ISBN: 0135717795.
[CHIA95]	Dante Chialvo, Mark Millonas, 1995, How swarms build cognitive maps, Fluctuation and Biophysics Group, The Santa Fe Institute.
[COEN00]	Michael Coen, 2000, Multimodal Integration: A Biological View, MIT Artificial Intelligence Lab, Cambridge, MA 02139.
[COEN97]	Michael H. Coen, 1997, Building Brains for Rooms: Designing Distributed Software Agents, MIT Media Lab, Copyright of American Association for Artificial Intelligence.
[COLL99]	Jaron Collis, Divine Ndumu, 1999, The Application Realisation Guide, The Zeus Agent Building Toolkit, Intelligent Research Systems Group, BT Labs, British Telecommunication plc.
[COLO93]	Marco Colombetti, Marco Dorigo, 1993, Training agents to perform sequential behaviour.
[DAVI93]	Randall Davis, Howard Shrobe, Peter Szolovits, MIT AI Lab, 1993, What is a Knowledge Representation?
[DECK95]	Keith Decker and Victor Lesser, 1995, Designing a family of Coordination Algorithms, Department of Computer Science, University of Massachusetts, Amherst.
[DECK99]	K.Decker, M.Fisher, M.Luck, M.Tennenholtz, and UKMAS'98 Contributors, 1999, Continuing Research in Multi-Agent Systems, This

Index	Description
	report summarises the paper presentations at the Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS'98)
[DIET00]	Thomas G. Dietterich, 2000, Hierarchical Reinforcement Learning with the MAXQ Value function Decomposition, Department of Computer Science, Oregon State University, Corvallis.
[DIET97]	Thomas G. Dietterich, 1997, Machine-Learning Research, Four Current Directions, AI Magazine, The American Association for Artificial Intelligence.
[DIMI98]	Dimitris N. Chorafas, 1998, Agent Technology Handbook, McGrow-Hill, Series on Computer Communications, ISBN-0-07-011923-6.
[DORI93]	Marco Dorigo, Marco Colombetti, 1993, Robot Shaping: Developing Situated Agents through Learning, International Computer Science Institute, Berkeley.
[DROG98]	A.Drogoul, and J.D. Zucker, 1998, Methodological issues for Designing multi-agent systems with Machine Learning Techniques: Capitalising Experiences from RoboCup Challenge.
[ELMA99]	J.L. Elman, E.A. Bates, M.H. Johnson, A. Karmiloff-Smith, D. Parisi, K. Plunkett, 1999, Rethinking Innateness, A connectionist Perspective on Development, ISBN: 0-262-05052-8.
[FARH97]	Faramarz Farhoodi and Peter Fingar, 1997, Developing Enterprise Systems with Intelligent Agent Technology, Featured in Distributed Object Computing Magazine, Object Management Group (OMG).
[FATI01]	S. Shaheen Fatima and Michael Wooldridge, 2001, Adaptive Task and Resource Allocation in Multi-Agent Systems, Department of Computer Science, University of Liverpool, U.K.
[FERB99]	Jacques Ferber, 1999, Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.
[FRAN96]	Stan Franklin and Art Graesser, 1996, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Institute for Intelligent Systems University of Memphis, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag.

Index	Description
[GAMM95]	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 1995, Design Patterns, Elements of Reusable Object-Oriented Software, ISBN: 81-7808-135-0.
[GARC00]	Carlos Gershenson García, Pedro Pablo Gonzalez, 2000, Dynamic adjustment of the motivation degree in an action selection mechanism.
[GARC01]	Carlos Gershenson García, 2001, Artificial Societies of Intelligent agents.
[GARD98]	Natalia Hernandez-Gardiol, Sridhar Mahadevan, 1998, Hierarchical Memory-Based Reinforcement Learning, Department of Computer Science, Michigan State University.
[GENE94]	Michael R. Geneserath, Narinder P. Singh, Mustafa A. Syed, 1994, A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation, Stanford University.
[GIFF00]	Adam Gifford JR., 2000, The Bioeconomics of Cooperation, Department o Economics, California State University.
[GORD99]	Deborah M Gordon, 1999, Ants at Work, How an Insect Society is Organised, ISBN: 0-684-85733-2.
[GRIF99]	Nathan Griffiths, Michael Luck, 1999, Co-operative Plan Selection Through Trust, Department of Computer Science, University of Warwick, U.K.
[GRUB93]	T.R. Gruber, 1993, A translation approach to portable ontologies.
[GULL92]	Vijaykumar Gullapalli, 1992, Reinforcement Learning and its Application to Control, PhD Dissertation, Department of Computer and Information Sciences, University of Massachusetts.
[HARM96]	Mance E. Harmon, Stephanie S. Harmon, 1996, Reinforcement Learning: A Tutorial, Wright Laboratory, Wright State University.
[HAZY04]	James K Hazy, Brian F Tivnan, 2004, On building an organizationally realistic agent-based model of location interaction and emergent network structure, WSC 04 proceedings of the 36^{th} conference on Winter simulation.
[HENR01]	Henry Lieberman, 2001, Attaching Interface Agent Software to Applications, Media Laboratory, Massachusetts Institute of Technology.
[HOGG97]	Lisa M Hogg, Nick R Jennings, 1997, Socially Rational Agents – Some Preliminary Thoughts, Department of Electronic Engineering, Queen Mary

Index	Description
	and Westfield College, University of London.
[HOLL90]	Bert Holldobler and Edward O. Wilson, 1990, The Ants, ISBN: 0-674-04075-9, The Belknap Press of Harvard University Press Cambridge, Massachusetts.
[HOLL94]	Bert Holldobler and Edward O. Wilson, 1994, Journey to the Ants: a story of scientific exploration, ISBN: 0-674-48526-2.
[HONA94]	Vasant Honavar, 1994, Symbolic Artificial Intelligence and Numeric Artificial Neural Networks: Towards a Resolution of Dichotomy, Iowa State University of Science and Technology, Department of Computer Science.
[HU98]	Junling Hu and Michael P Wellman, 1998, Artificial Intelligence laboratory, University of Michigan, USA.
[HUMP96]	Mark Humphrys, 1996, Action Selection methods using Reinforcement Learning, PhD Dissertation, University of Cambridge, Computer Laboratory.
[INVE96]	Mark d' Inverno, Michael Luck, 1996, Understanding Autonomous Interaction, 12th European Conference on Artificial Intelligence, John Wiley & Sons, Ltd.
[INVE97]	Mark d'Inverno, Michael Fisher, Alessio Lomuscio, Michael Luck, Maarten de Rajike, Mark Ryan and Michael Wooldridge, 1997, Formalisms for Multi-Agent Systems, Result of a panel discussion at the First UK Workshop of Multi-Agent Systems (Fo-MAS'96).
[INVE97b]	Mark Inverno, David Kinny, Michael Luck, Michael Wooldridge, 1997, A formal specification of dMARS.
[JENN94]	Nick R Jennings, 1994, Commitments and Conventions: The Foundations of coordination in Multi-agent systems, Department of Electronic Engineering, Queen Mary and Westfield College, University of London, UK.
[JENN95a]	N.R. Jennings, Michael Wooldridge, 1995, Intelligent Agents: Theory and

Index	Description
	Practice, Knowledge Engineering Review Oct. 1994, Revised 1995.
[JENN95b]	N.R. Jennings, 1995, Coordination Techniques for Distributed Artificial Intelligence, Queen Mary and Westfield College, University of London, UK.
[JENN96]	N. Jennings, 1996, Applying Agent Technology, Plenary presentation at PAAM'96.
[JENN98]	N. R. Jennings, M. Wooldridge, 1998, Application of Intelligent Agents, Queens Mary and Westfield College, University of London.
[JENN99]	Nicholas R. Jennings and Michael Wooldridge, 1999, Agent-Oriented Software Engineering, Department of Electronic Engineering, Queen Mary & Westfield College, University of London.
[JOHN02]	Steven Johnson, 2002, Emergence- The connected lives of ants, brains, cities and software, ISBN: 0-684-86876-8.
[KAEL91]	L.P. Kaelbling, 1991, A situated automata approach to the design of embedded agents, SIGART Bulletin
[KAEL96]	Leslie Pack Kaelbling, Michael L. Littman, 1996, Reinforcement Learning: A Survey, Brown University Providence and Carnegie Mellon University.
[KANA03a]	Parag M Kanade, Lawrence Hall, 2003, Fussy ant clustering by centroid positioning, Computer Science & Engineering Department, University of South Florida.
[KANA03b]	Parag M Kanade, Lawrence Hall, 2003, Fussy ant as a clustering concept, Computer Science & Engineering Department, University of South Florida. NAFIPS'03 Conference.
[KARP93]	Perter D. Karp, 1993, The Design Space of Frame Knowledge Representation Systems, Artificial Intelligence Centre, SRI International, SRI AI Centre Technical Note 520.
[KEIL03]	David Keil, Dina Goldin, 2003, Indirect interaction and decentralised coordination, Computer science and Engineering Department, University of Connecticut, USA.
[KENN01]	James Kennedy, Russell C Eberhart, 2001, Swarm Intelligence, ISBN: 1-55860-595-9.

Index	Description
[KIFE95]	Michael Kifer, Georg Lausen and James Wu, 1995, Logical Foundations of Object-Oriented and Frame-Based Languages.
[KINN92]	David Kinny, Magnus Ljungberg, Anand Rao, Gill Tidhar, Eric Werner, 1992, Planned team activity, 4th European workshop on modelling autonomous agents in a multi-agent world, Rome, Italy.
[KOES06]	Mary Koes, Illah Nourbakhsh, and Katia Sycara, 2006, Constraint optimisation coordination architecture for search and rescue robotics. Carnegie Mellon Robotics Institute.
[LABR94]	Yannis Labrou, Tim Finin, 1994, A semantics approach for KQML- A general purpose communication language for software agents, Computer Science Department, University of Maryland, Baltimore County, Baltimore MD 21228.
[LACE00a]	Timothy Lacey, Scott A. DeLoach, 2000, Automatic Verification of Multi-agent Conversations Eleventh Annual Midwest Artificial Intelligence and Cognitive Science Conference, University of Arkansas, Fayetteville.
[LACE00b]	Timothy Lacey, Scott A. DeLoach, 2000, Verification of Agent Behavioural Models, The International Conference on Artificial Intelligence (IC-AI'2000), Monte Carlo Resort, Las Vegas, Nevada.
[LASS01]	Ora Lassila, Deborah McGuinness, 2001, The Role of Frame-Based Representation on the Semantic Web Software Technology Laboratory, Nokia Research Centre, Knowledge Systems Laboratory, Stanford University.
[LASS90]	Ora Lassila, Frames or Objects, ot Both ?, 1990, Laboratory of Information Processing Science, Helsinki University of Technology, Finland.
[LEON97]	Leonard N. Foner, 1997, What is an agent, anyway? A sociological case study.
[LESZ04]	Rafal Leszczyna, 2004, Evaluation of agent platforms.
[LUCK96]	Michael Luck, Mark d'Inverno, 1996, Engagement and Corporation in Motivated Agent Modelling, Proceedings in the First Australian Workshop on Distributed Artificial Intelligence.

Index	Description
[LUCK97]	M.Luck, M. d'Inverno, M.Fisher, 1997, Foundations of Muit-Agent Systems: Techniques, Tools and Theory.
[LUCK99]	Michael Luck, 1999, From Definition to Deployment: What Next for Agent-Based Systems? Department of Computer Science, University of Warwick, U.K.
[MAES87]	Pattie Maes, 1987, How to do the right thing, AI-Laboratory, Vrije Universiteit Brussel and AI-Laboratory, Massachusetts, Institute of Technology.
[MARA00]	Alec Marantz, Yasushi Miyashita, and Wayne O'Neil, 2000, Image, Language, Brain: Papers from the First Mind Articulation Project Symposium, ISBN: 0-262-13371-7.
[MARK98]	Mark d' Inverno, David Kinny, Michael Luck, Michael Wooldridge, 1998, A Formal Specification of dMARS, Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages, Springer-Verlang.
[MATA94a]	Maja J Mataric, 1994, Interaction and Intelligent Behaviour, Ph.D. Dissertation, MIT, USA.
[MATA94b]	Maja J Mataric, 1994, Learning to Behave Socially, MIT Artificial Intelligence Laboratory, Cambridge, MA.
[MCGO01]	Amy McGovern, Andrew G. Barto, 2001, Automatic discovery of sub-goals in reinforcement learning using diverse density, Proceeding of the eighteenth international conference of machine learning.
[MEHR97]	Anshu Mehra, Mark Nissen, 1997, Case Study: Intelligent Software Supply Chain Agents using ADE.
[MFAC97]	Joint submission Crystaliz Inc, General Magic Inc., GMD FOKUS, IBM Corporation, 1997, Mobile Agent Facility Specification.
[MICH99]	Michael van Lent and John Laird, 1999, Developing an Artificial Intelligence Engine, Artificial Intelligence Lab, University of Michigan.
[MINS74]	A Framework For Representing Knowledge, 1974, Marvin Minsky, Massachusetts Institute of Technology, A.I. Laboratory

Index	Description
[MINS86]	Marvin Minsky, 1986, The Society of Mind, A Touchstone book published by Simon and Schuster, New York, ISBN 0-671-65713-5.
[MIZO95]	Riichiro Mizouchi, Johan Vanwelkenhaysen, Mitsura Ikeda, 1995, Task Ontology for Reuse of Problem solving Knowledge, Osaka University Japan.
[NASC00]	S Nascimento, B Mirkin and F Moura-Pires, 2000, A Fuzzy clustering model of data and fuzzy c-means, CENTRIA, departamento de informatica, Portugal.
[NASH50]	John Nash, 1950, Non-Cooperative Games, PhD, Dissertation, University of Princeton.
[NGUY02]	NguyenG, Dang T.T, Hluchy L, Laclavik M, Balogh Z, Budinska I, 2002, Agent platform evaluation and comparison.
[NICO06]	Vincenzo Nicosia, Cancetto Spampinato and Corrado Santoro for Eurobot DIIT Team, Universita di Catania, Italy.
[NORT01]	Nortel Networks Corporation, 2001, FIPA-OS Developers Guide
[NWAN96a]	Hyacinth Nwana & Divine Ndumu, 1996, An Introduction to Agent Technology, Intelligent Systems Research Applied Research and Technology, BT Labs.
[NWAN96b]	H.S. Nwana, L. Lee, N.R. Jennings, 1996, Coordination in software agent systems.
[NWAN98]	Hyacinth S Nwana, Divine Ndumu, Lyndon Lee, 1998, ZEUS: An advanced tool-kit for engineering distributed multi-agent systems, Intelligent Systems Research, Advanced Research and Technologies Department, BT Laboratories, London.
[OHTA00]	Shinya Ohtani, Masahiro Ishido, Katsunori Shimohara, 2000 Multi-Agent based Neural Network as a Dynamic Brain Model.
[OXFO88]	The Concise Oxford Dictionary, 1988, 7th Edition, ISBN: 19-561491.
[PAOL97]	Ezequiel A Di Paolo, 1997, An Investigation into the evolution of communication, School of Cognitive and Computing Sciences, University of Sussex, UK.
[PAOL99]	Ezequel Aleiandro Di Paolo, 1999, On the evolutionary and behavioural

Index	Description
	dynamics of social coordination: models and theoretical aspects, PhD Dissertation, University of Sussex.
[PARR97]	Ronald Parr, Stuart Russell, 1997, Reinforcement Learning with Hierarchies of Machines, Computer Science Division, UC Berkeley, CA 94720.
[PARU94]	H. Van Dyke Parunak, 1994, Applications of Distributed Artificial Intelligence in Industry, Industrial Technology Institute, U.S.A.
[PARU96]	H. Van Dyke Parunak, 1996, "Go to the Ant": Engineering Principles from Natural Multi-Agent Systems, Center for Electronic Commerce, ERIM.
[PARU96b]	Van Dyke Parunak, 1996, Workshop Report: Implementing Manufacturing Agents, The Industrial Technology Institute, UK.
[PARU97a]	Van Parunak, John Sauter, and Steve Clark, 1997, Toward the Specification and Design of Industrial Synthetic Ecosystems, Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).
[PARU97b]	H. Van Dyke Parunak, Raymond S. VanderBok, 1997, Managing Emergent Behaviour in Distributed Control Systems, Presented at ISA-Tech '97, Anaheim, CA.
[PARU98]	H. Van Dyke Parunak, 1998, Practical and Industrial Applications of Agent-Based Systems Industrial Technology Institute.
[PATT91]	Pattie Maes, 1991, The Agent Network Architecture, SIGART Bulletin.
[PATT94]	Pattie Maes, 1994, Modelling Adaptive Autonomous Agents, MIT Media Laboratory.
[PEDD02]	Taraka Peddireddy, Jose Vidal, 2002, Multi-agent Network Security System using FIPA-OS.
[PERE00]	Pedro Pablo González Pérez, José Negrete Martínez, Carlos Gershenson García, Ariel Barreiro García, 2000, A Model for Combination of External and Internal Stimuli in the Action Selection of an Autonomous Agent.
[PETR95]	Charles J Petrie, Teresa Webster, Mark Cutkosky, 1995, Using Pareto optimality to coordinate distributed agents, AIEDAM special issue on conflict management vol. 9.
[PETR96]	Charles J. Petrie, 1996, Agent-Based Engineering, the Web, and Intelligence, Stanford Centre for Design Research, December 1996 issue of IEEE Expert.

Index	Description
[PFEI01]	Rolf Pfeifer, Hanspeter Kunz, Marion M. Weber 2000, Artificial Life.
[PFEI02]	Rolf Pfeifer, 2002, On the role of embodiment in the emergence of cognition Grey Walter's turtles and beyond, AI Laboratory, Department of Information Technology, University of Zurich, Switzerland.
[PIER94]	Pierguido V C Caironi, Marco Dorigo, 1994, Training and Delayed Reinforcements in Q-Learning Agents, Universite' Libre do Bruxelles.
[PIRJ99]	Paolo Pirjanian, 1999, Behaviour coordination mechanisms – state of the art, USC Robotics Research Laboratory, University of Southern California, USA.
[PISO03]	John Pisokas and Ulrich Nehmzow, 2003, Experiments in sub symbolic action planning with mobile robots, Department of computer science, University of Essex, UK.
[POLL90]	Jordan B. Pollack, 1990, Recursive Distributed Representations, The Ohio State Unidersity.
[PRIC03]	Bob Price, Craig Boutilier, 2003, Accelerating Reinforcement Learning through Implicit Imitation, Journal of Artificial Intelligence Research 19 (2003) 569-629.
[PYNA02]	David V. Pynadath, Milind Tambe, 2002, The communicate Multi-agent Team Decision Problem: Analysing Teamwork Theories and Models, Information Sciences Institute and Computer Science Department, University of Southern California.
[RANA02a]	R.A.C. Ranasinghe, A.P. Madurapperuma, 2002, "AAANTS – An Intelligent Synthetic Ecosystem", IITC Colombo, Sri Lanka.
[RANA02b]	R.A.C. Ranasinghe, A.P. Madurapperuma, 2002, "AAANTS – Distributed Mobile Component Architecture for an Ant Colony Based Synthetic Ecosystem", MATA-2002, Spain Barcelona, Short Paper.
[RANA03a]	R.A.C. Ranasinghe, A.P. Madurapperuma, 2003, "Non-Intentional Cooperative Behaviour for an Agent Based Intelligent Environment", IITC Colombo, Sri Lanka.
[RANA03b]	R.A.C. Ranasinghe, A.P. Madurapperuma, 2003, "AAANTS-A Distributed

Index	Description
	Agent Framework for Adaptive and Collective Intelligence", Mobile Agents for Telecommunication Applications (MATA'2003), IEEE/IFIP, Marrakech, Morocco.
[RANA03c]	R.A.C. Ranasinghe, A.P. Madurapperuma, 2003, "Enhanced Frame Based Representation for an Intelligent Environment", (KIMAS'2003), International Conference on Integration of Knowledge Intensive Multi-Agent Systems, IEEE Boston Section, Cambridge MA, USA.
[RANA05]	R.A.C Ranasinghe, A.P. Madurapperuma, 2005, "Learning Coordinated Actions by Recognising State Patterns with Hubs", Eighth International Conference on Human and Computers, University of Aizu, Japan.
[RANA99]	R.A.C. Ranasinghe, A.P. Madurapperuma, 1999, "Distributed Service Utilisation on Heterogeneous Networks using Mobile Agents", IITC, Colombo, Sri Lanka.
[REIC91]	Han Reichgelt, 1991, Knowledge Representation: an AI perspective, ISBN: 0-89391-590-4.
[RESN94]	Mitchel Resnick, 1994, Turtles, Termites, and Traffic Jams, Explorations in Massively Parallel Microworlds, ISBN: 0-262-18162-2.
[REYN02]	Stuart Ian Reynolds, 2002, Reinforcement Learning with Experience, PhD thesis, School of Computer Sciences, The University of Birmingham, UK.
[RICO00]	Pierre-Michel Ricordel and Yves Demazeau, 2000, LEIBNIZ Laboratory France.
[RUI02]	Rui Rocha, 2002, Fostering Cooperation among Intelligent Machines, ISR – Universidad de Marrocos.
[RUMM95]	Gavin Adrian Rummery, 1995, Problem Solving with Reinforcement learning, PhD Dissertation, Cambridge University Engineering Department, Cambridge, England.
[RUSS95]	Stuart Russell, Peter Norvig, 1995, Artificial Intelligence A Modern Approach, ISBN: 81-7808-554-2.
[SALA00]	Isaac Salazar-Ciudad, Jordi Garcia-Fernandez, and Ricard V Sole, 2000, Gene networks capable of pattern formation: From induction to reaction-diffusion, Complex Systems Research Group, Department of Physics, FEN-UPC Campus Nord, Barcelona Spain.

Index	Description
[SAND95]	Tuomas W Sandholm, and Robert H Crites, 1995, Multi-agent Reinforcement Learning in the Iterated Prisoner's Dilemma, University of Massachusetts at Amherst, Computer Science Department, Amherst.
[SAUT99]	John A. Sauter, H. Van Dyke Parunak, 1999, ANTS in the supply chain, The original version of this paper was presented at the Workshop on Agent based Decision Support for Managing the Internet-Enabled Supply Chain, Agents 99, Seattle, WA.
[SCHA95]	Andrea Schaerf, Yoav Shoham, Moshe Tennenholtz, 1995, Adaptive Load Balancing: A study in multi-agent learning, Journal of Artificial Intelligence Research pg. 475-500.
[SCHN98]	Jeff Schneider, Wong-Keen Wong, Andrew Moore, Martin Riedmiller, 1998, Distributed Value Functions, Carnegie Mellon University, Pittsburgh, USA.
[SING95]	Narinder P. Singh and Mark A. Gisi, 1995, Coordinating Distributed Objects with Declarative Interfaces, Stanford University California, Hewlett-Packard Laboratories, Palo Alto.
[SOLE01]	Ricard Sole and Jose Montoya, 2001, Complexity and fragility in ecological networks. The Royal Society.
[STAN95]	Stanley P. Franklin, 1997, Artificial Minds, The MIT Press, ISBN: 0-262-06178-3 (HB).
[STON00]	Peter Stone, 2000, Layered Learning in Multi-Agent Systems, A winning approach to robotic soccer, The MIT Press, Massachusetts Institute of Technology, ISBN 0-262-19438-4.
[STON98]	Peter Stone, 1998, Layered Learning in Multi-Agent Systems, School of Computer Science, Carnegie Mellon University, Pittsburgh.
[SUMP00]	David Sumpter, 2000, From Bee to Society: An agent-based investigation of honey bee colonies. PhD Thesis, Department of mathematics, University of Manchester Institute of Science and Technology.
[SUTT98a]	Richard S. Sutton and Andrew G. Barto, 1998, Reinforcement Learning: An Introduction MIT Press, Cambridge, MA, A Bradford Book
[SUTT98b]	Reinforcement Learning: Introduction, 1998, R.S. Sutton, A.G. Barto, Course Notes.
[SYCA99a]	Katia Sycara, Onn Shehery, 1999, The RETSINA Communicator, The

Index	Description
	Robotics Institute, Carnagie Mellon University, IBM Research Lab.
[SYCA99b]	Katia Sycara, Matthias Klusch, Seth Widoff, Jianguo Lu, 1999, Dynamic Service Matching Among Open Information Environments, The Robotics Institute, Carnegie Mellon University, U.S.A. Computer Science Department, University of Toronto, Canada.
[TANG02]	Poj Tangamchit, John M Dolan, Pradeep K Khosla, 2002, The Necessity of Average Rewards in Cooperative Multi-robot Learning, Dept. of Electrical Engineering, The Robotics Institute, CMU, USA.
[TOTH00]	Tihamer T. Toth-Fejel, 2000, Agents, Assemblers, and ANTS Scheduling Assembly with Market and Biological Software Mechanisms, Environmental Research Institute of Michigan, Centre for Electronic Commerce, Seventh Foresight Conference on Molecular Nanotechnology.
[TUNG01]	An Electronic Market-Maker, 2001, Nicholas Tung Chan and Christian Shelton, Massachusetts institute of technology-Artificial Intelligence Laboratory.
[VENN97]	Bill Venners, Java World, 1997, The architecture of Aglets.
[VIDA02]	Jose Vidal, Paul Buhler, 2002, A generic Agent Architecture for Multi-Agent Systems.
[WANG03]	Xiao Ying Wang, Jon Garibaldi, Turhan Ozen, 2003, Application of the Fuzzy C-Means clustering method on the analysis of non pre-processed FTIR data for cancer diagnosis, The University of Nottingham, UK.
[WEBS88]	The Merriam Webster Dictionary, 1988, ISBN: 0-87779-508-8.
[WEIS00]	Gerhard Weiss, 2000, Multi-agent systems, A Modern Approach to Distributed Artificial Intelligence, ISBN: 0-262-23203-0.
[WHIT00]	Darrell Whitley, 2000, An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls, Computer Science Department, Colorado State University.
[WINS92]	Patrick Henry Winston, 1992, Artificial Intelligence, 3rd Edition, ISBN: 81-7808-506-2.
[WOLP00]	David H Wolpert, Kagan Tumer, 2000, NASA Ames Research Centre, Moffet Field, CA 94035.

Index	Description
[WOOL00]	M. Wooldridge, 2000, Reasoning about rational agents, The MIT Press, Cambridge, Massachusetts, London, ISBN: 0-262-23213-8.
[WOOL94]	Michael J. Wooldridge, Nicholas R. Jennings, 1994, Agent Theories, Architectures, and Languages: A Survey, Manchester Metropolitan University, Queen Mary & Westfield College.
[WURR03]	Alfred Wurr, 2003, Robotic team navigation in complex environments using stigmergic clues, MSc thesis, Department of computer science, The University of Manitoba, Manitoba.

Appendices

Appendix A: Definitions of Software Agents

Definition 1: "An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives"[JENN99].

Definition 2: "An agent is a system that tries to fulfil a set of goals in a complex, dynamic environment. An agent is situated in the environment; it can sense the environment through its sensors and act upon the environment using its actuators. Autonomous agents are systems that inhabit a dynamic, unpredictable environment in which they try to satisfy a set of time dependent goals or motivations. Agents are said to be adaptive if they improve their competence at dealing with these goals based on experience" [PATT94].

Definition 3: "Agent can be defined as a computer system that is either conceptualised or implemented using concepts that are more usually applied to humans. Agent can be denoted as hardware or software based computer system that enjoys the properties such as autonomy, social ability, reactivity, and pro-activeness" [JENN95a].

Definition 4: "Agents can be defined as intelligent toolboxes with a primary purpose of providing active assistance to their environment" [DIMI98].

Definition 5: "An agent is a computer program that acts autonomously on behalf of a person or organisation. Each agent has its own thread of execution so that it can perform tasks on its own initiative" [MFAC97].

Definition 6: "An intelligent agent is software that can take independent actions on behalf of a user's goals without explicit interaction by the user" [HENR01].

Definition 7: "An agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" [FRAN96].

Definition 8: "A software entity is an agent if it has the data and code encapsulation of a software object, its own thread of control (making it an active object), and the ability to execute autonomously without being invoked externally (thus proactive rather than reactive)" [PARU98].

Appendix B: Existing Agent Platforms and Architectures

Many agent platforms were investigated to evaluate whether the AAANTS model could be realised through these implementations. Among these platforms, the Agent Network for Task Scheduling (ANTS) architecture [SAUT99], was identified as an agent platform that is inspired by both human institutions and insect colonies which is similar to that of the AAANTS model. This platform was designed for the domain of supply chain management where large populations of simple agents exhibit robust behaviour in scheduling supply chains. Further, the ANTS is a agent based system that could assist and supplement human-based interaction and decision making in a supply chain without the need for centralised or top-down management schemes. However, it should be reiterated that this platform is not of generic nature and specific to the domain of supply chain management.

Anthill [BABA01] is modelled from inspiration from ant colony behaviour similar to the AAANTS model. The Anthill model is based on two logical entities: nests and ants. Each nest is a peer entity capable of performing computations and hosting resources. Nests handle requests originating at users by generating one or more ants. Ants interact indirectly with each other by modifying their environment through information stored in the visited nests. Additionally, Anthill pushes the analogy with natural systems even further by "evolving" ant algorithms to better adapt to certain tasks. This is accomplished through evolutionary computing techniques such as genetic algorithms within a simulation environment.

Scatterbrain [COEN97] [COEN98] is a distributed collection of software agents related to the domain of Intelligent Environments. It describes an existing prototype space known as the intelligent room, which is created to experiment with different forms of natural, multi-modal human-computer interactions. In Scatterbrain, the complexity of the overall system comes from the interactions of a collection of agents which is similar in concept to the emergent nature of the AAANTS model. It consists of distinct intercommunicating agents with the primary task of linking various components of an intelligent room such as tracking cameras, speech recognition systems and to connect them to internal and external stores of information.

The above discussed platforms have direct relationship with the AAANTS platform due to similarity of the foundation concepts. However, there were several other platforms evaluated during the initial stages which were based on different agent models such as Cognitive, Reactive and BDI. RETSINA [SYCA99a], MadKit [RICO00], Zeus [RICO00] and ADE [ANDR03] were based on the cognitive agent paradigm. RETSINA, MadKit and Zeus are multi-agent systems implemented on high-level communications languages for agent collaboration. ADE is also a multi-agent system developed in Java with the facility of agent mobility. All these platforms were built of generic platforms that could be configured for different cognitive agent implementations. Due to the hybrid nature of the AAANTS model, the above listed platforms were not considered.

The Belief, Desire, Intention (BDI) [BUSE99] [MARK98] [GRIF99] [WOOL00] [PYNA02] is another popular agent model accepted by the research community to model intelligent software artefacts in which an agent's beliefs correspond to information the agent has about the world which may be incomplete and or incorrect. AgentBuilder [RICO00], Jack [RICO00] and dMARS [RICO00] are conceptualised on the DBI agent model. BDI model also falls into the cognitive agent paradigm and not considered for the AAANTS research.

Appendix C: Frames: Structure, Control, Transformation and Manipulation

An important intuition underlying a frame system is that people cope with new situations by relating the new information similarity to previous experiences [MINS74]. A frame is a data structure that is typically used to represent a single object or a class of related objects, or a general concept or predicate [KARP93] [REIC91] [LASS01]. Frames are typically arranged in a taxonomic hierarchy in which each frame is linked to one or many parent frames. Therefore a collection frames in one or more inheritance hierarchies could be called as a knowledge base [KARP93].

The slots of a frame describe attributes of the entity represented by that frame, and could also describe binary relations between that frame and another frame [KARP93]. A slot usually consists of two parts: A slot name, which describes an attribute, and a slot-filler, which describes either a value for that attribute or a restriction on the range of possible values [KARP93]. In most frame systems we can identify two types of frames [KARP93]. The first type of frame, called a class-frame, is a description of a class of entities in the world. The second type of a frame is the instance frame, which is an intentional description of an individual entity in the world.

A collection of frames could be linked together into frame-systems [MINS74]. The effects of important actions are mirrored by transformations between the frames of a system. The frame-systems are linked, in turn, by an information retrieval network. When a proposed frame cannot be made to fit reality, this network provides a replacement frame [MINS74]. These inter-frame structures make possible other ways to represent knowledge about facts, analogies, and other information useful in understanding [MINS74].

Once a frame is proposed to represent a situation, a matching process tries to assign values to each frame's terminals, consistent with the representation of each situation. The matching process is partly controlled by information associated with the frame and partly by knowledge about the system's current goals [MINS74]. Frame system reasoning may sometimes be incomplete [DAVI93] and frame systems do not typically make guarantees about the computational tractability of their inference [LASS01]. There are several techniques for finding and organising frames such as Pattern Matching Process, Clustering Theory, and Similarity Network [MINS74].

Each frame has terminals for attaching pointers to substructures. Different frames could share the same terminal, which could thus correspond to the same physical features as seen in different views. This permits us to represent, in a single place, view independent information gathered at different times and places [MINS74].

Appendix D: Credit Assignment Techniques

The basic problem of any learning system is to deal with the Credit-Assignment Problem (CAP), that is, the problem of properly assigning feedback-credit or blame for an overall performance change (increase or decrease) to each of the system activities that contributed to that change. The CAP for MASs could be usefully decomposed into two sub problems [WEIS00]: Inter-agent CAP – the assignment of credit or blame for an overall performance change to the external actions of the agents and intra-agent CAP – the assignment of credit or blame for an overall performance change to the internal actions of the agents.

According to Sachiyo Arai [ARAI00c], there are two credit assignment procedures based on boot-strapped and non-boot-strapped methods. The boot-strapped method is inspired by dynamic programming and attempts to satisfy Bellman equations relating to the values of successive states to make the agent behave optimally [ARAI00c]. The non-boot-strapped method is inspired by classifier systems [DIET97] and does not attempt to estimate the value of all rules that cover the state space, but just accumulates the weight on successful rules based on the agent's experience [ARAI00c].

According to Vijaykumar Gullapalli [GULL92], the structural credit assignment has two aspects: the hidden component and multiple action elements. In the hidden component, CAP involves assigning credit to those components that do not directly interact with the environment. The multiple action elements arise when the learning system's actions are multi-dimensional and the learning system has to determine the relative impact of each action element in various situations to apportion credit among the action elements for the ensuing evaluation [GULL92]. The Gradient methods could be used for credit assignment in hidden component aspects. The gradient methods indicate the degree of influence of each unit on the criterion function, and hence each unit is assigned credit proportional to the magnitude of the gradient [GULL92].

Another technique is based on a measure of worth of a network component [GULL92]. The measure of worth is used to determine how to streamline the structure of the learning system by discarding useless components and by adding useful new components

[GULL92]. The worth of a component is estimated by evaluating the contribution it makes to the output of the learning system. A commonly used measure is the overall output of the learning system. The credit assignment based on contribution could be achieved by assigning sensitivity values to individual action elements in each context [GULL92]. These sensitivity values are used to scale the learning rates when adjusting the action elements. Action elements with low sensitivity values undergo relatively smaller adjustments than action elements with high sensitivity, thereby reducing superstitious learning in the elements that are inactive in a given context. The partial derivatives are used to calculate the sensitivities of action elements [GULL92].

Appendix E: Background on Evolutionary Techniques

Anthill [BABA01] is a simulation environment that uses evolutionary techniques such as genetic algorithms in designing ant algorithms. Ant algorithms are based on the behaviour of ants that uses pheromones to find the optimal path whether based on the deterministic or probabilistic methods. The Anthill simulation environment has been extended to enable the definition of a collection of such parameters and the selection of the fittest set of parameters for a particular task.

Evolutionary agents could also be used to improve capabilities of an agent community. MAB-Net [OHTA00] creates a new artificial neural network model with a dynamic structure. In order to put dynamic structure into practice, in MAB-Net, neurone's functions as agents while they grow, cut connections, self-replicate, and evolve. The key idea is that neurones work not only as a neural network but also as evolutionary agents. As an evolutionary agent, each neurone has a gene that works as a strategic program; and based on the strategic program, neurones execute these behaviours, such as grow, absorb, move, turn, and so on. Each neurone has a gene and energy. A gene is decoded into a strategic program and energy is reduced every time when commands on the strategic program are executed. These are the commands on the strategic programs. When the Cell-Divide command is executed, a new agent is generated. If energy runs out, an agent is killed.

Another solution is the Amalthaea system [ALEX96] that also focuses a lot on the evolutionary aspects of the agents. Amalthaea describes evolutionary agents that are handled by two elements: their individual fitness and the overall fitness of the system. Only a variable number of top ranked performers of the whole population are allowed to produce offspring. The rank of an agent is based solely on its fitness. The number of the agents that will be allowed to produce offspring is linearly related to the number of agents that will be purged because of poor performance. If the overall fitness diminishes, then the evolution is increased in search for quicker adaptation of the users' new interests. If the overall fitness increases the evolution is kept at a constant configurable rate to allow the system to slowly explore the search space for better solutions.

Another two popular concepts used in this sphere of interest are Evolutionary Programming (EP) Genetic Algorithms (GA) [KENN01]. Evolutionary programming is derived from the simulation of adaptive behaviour in evolution: GA is derived from the simulation of genetics [KENN01]. The difference is perhaps subtle, but important. Genetic algorithms work in the genotype space, while evolutionary programming (EP) emphasizes the phenotype space of observable behaviours [KENN01]. EP therefore is directed at evolving "behaviour" that solves the problem at hand; it mimics "phenotype evolution" [KENN01].

Appendix F: Hamilton's Formula

This formula describes the balance between the urge for reproduction and dedication to the community of partners. The humans and insects have considerable differences when it comes to reproduction. Humans tend to be very self-centred and act in a selfish manner to participate in sexual activities whereas insects sacrifice their reproductive rights to the success of the immediate family. We may find rare instances in the human society where sister and brothers scarifies their sexual interests to take care of offspring's of kin. However, insects have this altruistic capability embedded in their genes.

The purpose of reproduction is to pass down genes to the next generations. There are two ways for alleles to be passed to future generation: personal reproduction and promoting genes of common decent or collateral relatives [HOLL90]. Both these scenarios could be described with a Fitness Indicator. The measure of the personal reproductive success is known as Classical Fitness and Inclusive Fitness incorporates both Classical Fitness and its influence on the reproduction of collateral relatives [HOLL90]. The following formulas describe these two indicators.

Classical Fitness $(W) = \dfrac{E(RS)}{A(RS)}$

(Where E(RS) is the average direct reproductive success of individuals possessing the genotype of interest which measures the number of offspring the individual injects into the population, in comparison with the remainder of the population; A(RS) is the average reproductive success of a population)

Inclusive Fitness $(IF) = \dfrac{E(RS) + \sum b_j E(RS)}{A(IF)}$

(Where $\sum b_j E(RS)$ is the effect on the reproduction of all of the collateral relatives; b_j is the coefficient of relatedness, the probability that the relative j of the focal individual also possesses the allele of interest, A(IF) is the average inclusive fitness of the population)

Hamilton's rule says that the benefit to relatives is discounted by the degree of relationship, as a result, lesser the relatedness, the greater the benefit must be to counter balance the cost [HOLL90].

Hamilton's Rule: $C(Cost)/B(Benifit) < b(\mathrm{Pr} - relative - having - same - allele)$

(Where C is the loss in expected personal reproductive success through self-sacrificing behaviour; B is the increase in the relatives' expected reproductive success, b is the probability that the relatives have the same allele)

When part of the group sacrifices their reproductive rights for the others, then the reproductive group should be able to perform that function and benefit the overall community considerably better than themselves getting involved in reproduction [HOLL90].

Appendix G: Research Publications

Paper 1: Learning Coordinated Actions by Recognising State Patterns with Hubs

R.A. Chaminda Ranasinghe, Dialog Telekom Ltd, Sri Lanka,

A.P Madurapperuma, University of Moratuwa, Faculty of IT, Sri Lanka.

Chaminda.Ranasinghe@dialog.lk

Ajith@itfac.mrt.ac.lk

Abstract

The AAANTS (Adaptive Autonomous Agent Colony Interactions with Network Transparent Services) is a multi-agent system conceptualised to model distributed collective intelligence by coordinated actions of a colony of agents. The model is inspired by the fascinating accomplishments of colony based insects such as Ants. Each individual in a colony is so simple and insignificant, though they collectively accomplish complicated tasks. Hence, the selection of the learning methodology, knowledge representation and coordination technique is of prime importance to the success of this model.

Our conscious lives can be regarded as a gradual movement from one state to another within the environment. During this state transition, the decision to select the next state could be learnt from pervious experience or could be based on spontaneous and purely stochastic exploration. The AAANTS model describes a technique based on Reinforcement Learning that maintains a partial model of the environment. Further, the definition of Hub states is of special importance to this model. A Hub state which can also be called as sub-goals is a critical state that directly impacts the achievement of the final goal. It is the use of Hub states that enable a collection of agents to perform concurrent and coordinated actions. The coordination is facilitated by pheromone based communication among a collection of homogeneous agents which enables the clustering around Hub states. This methodology was tested using experiments on foraging, robotic arm movement and vision based navigation. The preliminary results of these experiments have shown considerable improvements over techniques based on traditional Reinforcement Learning techniques.

Introduction

All entities of living and non-living nature show signs of emergence. It is accepted that objects are constructed by minute atomic particles, though different patterns of such particles creates a fascinating diversification in the environment. Similarly, complex behaviours are emerged from atomic actions generated by a myriad of objects. The secret lies in learning the execution of the optimum concurrent and coordinated action pattern by all participants. Further, the evidence of emergence is witnessed in theories such as creation of mind from mindless elements [MINS86] and colony behaviour of Ants where each Ant is very basic and insignificant [HOLL94].

Adaptations to overcome state changes within the environment could happen as genetical improvements within a species or by active learning. Lower level animals such as insects tend to overcome state manipulation mainly based on genetically implanted information. The genetics tend to implement in-born capabilities to map correct actions to environmental perceptions. Even higher level animals such as humans tend to posses capabilities such as reflex actions that are genetically defined. Obviously, the animals with shorter life-span tend to thrive on genetical information since there is no reasonable time to learn either by supervision or reinforcement. Therefore, genetical mutation is a very useful mechanism for the survival of many species where reinforced and supervised learning techniques enable survival during a single lifespan.

The AAANTS model was conceptualised for simulating collective intelligence with simple and incomplex software elements. The software elements are described as software agents where each agent is autonomous, adaptive and communicates with others in the community [JENN95]. The model blends the techniques of reinforcement learning, frame-based knowledge representation and pheromone based communication among insects. The model also encompasses the process of assigning the learning ability to a distributed collection of software agents. The valuable lesson learnt from this exercise is the emergence of complex behaviour as a result of coordinated actions of simple agents.

A valuable concept being invested in this paper is the use of "Hubs". In any network, the network elements are connected through links. Networks with connectors (nodes with an enormous number of links) are present in very diverse complex systems [BARA02]. These concentrated and highly connected nodes are called as "Hubs" [BARA02]. It is assumed that the optimum path through a network can be found by traversing across the most concentrated Hubs. Hence, Hubs be could regarded as intermediary states or sub-goals for reaching the final goal and regarded as very crucial to the success of the learning model. The Hub concept also has an orthogonal effect of demonstrating a layered learning process. These layers could be expanded or collapsed with the effects of exploration and exploitation strategies.

Architecture of the AAANTS Learning System

The Architecture of a system should encompass the building components and their interconnectivity. It should also facilitate the functionality expected out of that entity. There is wide variety of learning system architectures that has evolved in the natural world. The purpose of the learning architecture of an animal is to facilitate the survival in a particular type of environment. The demands of the environment and the purpose of the species dictate the overall learning structure.

An important objective of the AAANTS architecture is to facilitate a general purpose learning platform that can be adapted to different situations. This paper presents several experiments to justify this objective - foraging in a grid world, robotic arm movement and vision based navigation. Another objective of an architecture is to facilitate collective learning. Hence, the AAANTS model can be described as a distributed learning system as opposed to a centralised cognitive system. In a distributed architecture the system tasks are mapped to a series of atomic behavioural modules and are interconnected to build a complete learning system [COLO94] [COLO93]. A distributed architecture could be represented as flat or hierarchical where AAANTS model could be described as hierarchical which may be also debated as a layered model.

The hierarchical architecture gives way to a layered model. There is lot of alignment from many other research publications about the use of layering and modular decomposition. According to [MINS86], the development of the human mind from infancy to adulthood is achieved in terms of stages where each stage acts as a teacher to the next stage by providing guidance and assistance. MAXQ method [DIET00] provides a similar strategy to that of AAANTS where it decomposes a Reinforcement Learning problem into a set of sub-problems. This is further confirmed by arguments of [GULL92], where a high level complex task may be decomposed into a sequence of lower-level tasks and thereafter the same activity performed on the sub-tasks until atomic functions are reached. HAM (Hierarchical Abstract Machines) [PARR97] is another similar concept where nondeterministic finite state machines are organised in a hierarchy and higher level abstractions invoke lower level machines [PARR97]. Further, machines for HAMs are defined by a set of states, a transition function, and a start function that determines the initial state of the machine.

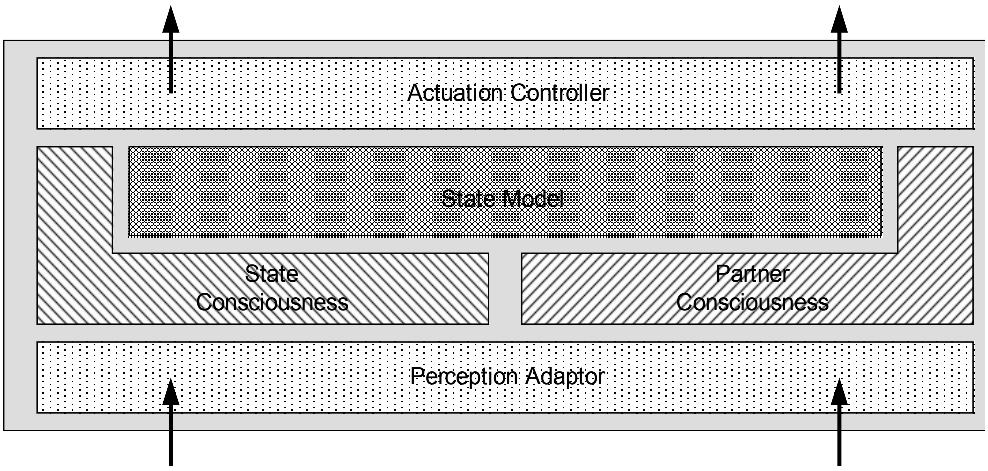


Figure 1: Learning Architecture of an Agent

AAANTS model is best described as a colony of agents where each agent contributes to the overall success of the system. Agents maintain a limited model of the environment and coordinate actions to achieve global objectives. The overall architecture of the AAANTS model consists of agents, together with sensory and actuation adaptors [RANA02] [RANA03]. The architecture of an individual agent with the emphasis on the main modules that facilitate learning is depicted in figure 1. Agents interface to the outer world (consisting of other agents, sensations and actuations), with the use of Perception Adaptor and Actuation Controller. The State Model maintains a summary of states that is of interest to the agent which will always represent a subset of global states. It maintains a repository of information about state-action values, hub states and optimal / sub-optimal paths. The Consciousness modules keep references to the on-going states of each individual agent as well as to the others in the colony within the perimeter of a homogeneous group. It is with the use of the Consciousness elements that an agent can take actions that are coordinated with others.

The Learning Method

AAANTS learning method uses Reinforcement Learning, Frames, Agents and Hubs as ingredients. A goal can be achieved with a properly coordinated sequence of actions by a community of agents. Inter-agent communication is a facilitator for this coordination. During the initial agent interactions with the environment, each agent tends to maintain a flat structure of state sequences and as the agents iterate though the environmental states with the objective of finding the global optimum, collection of states separated through Hubs are arranged into a layered hierarchy (figure 2). With reference to (figure 2), the states represented by L1S1, L1S2, L1S3 ending with sub-state L1S4 in layer 1, can be represented by a single state (L2S1) in layer 2.

A higher level state represents a link between two very important lower level states – the Hub states. This is to some extent similar to Nearest Sequence Memory (NSM) where raw experiences are recorded as a linear chain and the choice of the next action is evaluated based on the nearest neighbours in the experience chain [GARD98]. It is also researched that organising past experiences hierarchically scales better to problems with long decision sequences than organising past experiences as a linear chain of primitive observations and actions [GARD98]. Further, each layer is represented by different instances of agents where the higher level agents trigger lower levels to perform actions. The hierarchy is not pre-defined, but dynamically expanded and collapsed with the iterative adaptations to a changing environment.

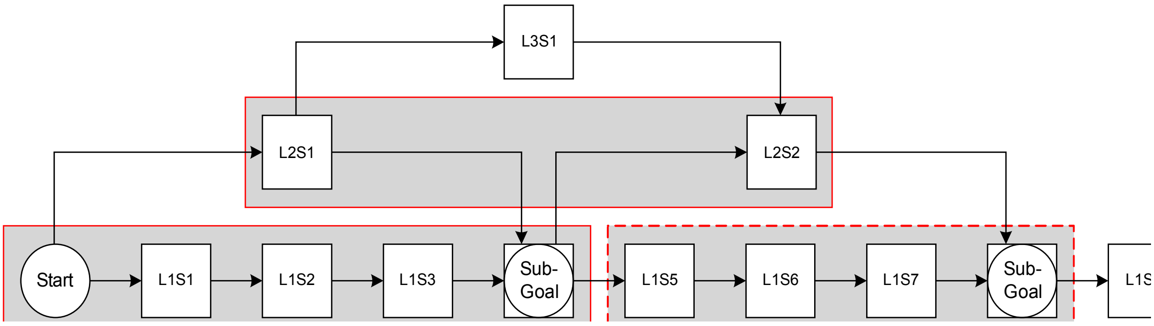


Figure 2: Hierarchical layers based on hubs

The two highly researched methods in RL so far by the research community are Monte-Carlo (MC) and Temporal Difference (TD) [SUTT98]. These methods can be further enhanced and also combined in flavour with the use techniques such as approximation, eligibility, models, and active/passive learning. This thesis concentrates on a learning methodology based on TD based "Sarsa" while extending it with the use of models. Further, a reinforced methodology can be on-policy and off-policy. On-policy methods evaluate and improve the same policy that is used to make decisions. In off-policy methods, the policy to generate behaviour (behaviour policy) is different to the policy that is evaluated and improved (estimation policy) [SUTT98]. AAANTS invests on on-policy method that is evaluated episodically.

We use the concept of frames as described by Minsky [MINS86] to implement the state representations. Each agent will capture its share of state instances as experience of the environment grows. These frames are attached to each other in a manner of representing the experience with the environment. The links among the frames are strengthened or weakened with the help of episodic reinforcements given to the agents. Over a period of time, an agent may accumulate a vast collection of state instances where in a complex environment may become impractical to maintain. In order to overcome such situations humans settle down on standard averages to define similar states and also use heuristics to overcome complex situations. We call this as common-sense reasoning. In a learning methodology, function approximation techniques and non-monotonic logic are used to implement heuristics.

A complex task can be achieved through the execution of elementary actions. Implementing sequential execution of a collection of elementary actions to implement a complex task is of trivial nature. However, most tasks require overlapped concurrent execution of actions by individual control units, which in AAANTS concept is represented by agents. The key issue with concurrency is in finding the optimum coordination strategy. The rest of the sections in this paper discuss this coordinated division of labour with the use of Hubs. Agents treat hubs as goals where overall activity can be collectively carried out by a group of agents.

AAANTS model proposes social learning as a useful technique for improvement. Social or observational learning is the process of acquiring new behaviour patterns in a social context, by learning from conspecifics [MATA94]. Social learning could be implemented though imitation and mimicry where it is useful to differentiate each other. Though both mimicry and imitation observe and repeat the behaviour of another agent, in mimicry the mimicking agent does not understand the goal of the behaviour or the internal state of the agent being mimicked [MATA94]. Social facilitation is another social learning method which refers to the process of selectively expressing a behaviour which is already a part of the animal's species-specific repertoire [MATA94]. A society can develop social rules based on individual learning if the agents are able to estimate other agents' reinforcement and their individual reinforcement is positively correlated with their conspecifics [MATA94].

The AAANTS model implements social learning by introducing an agent with an optimum state sequence to achieve the goal. It is this technique that blends the supervised and reinforcement learning methodologies. Usually, agents converge towards the optimum path where in insects this is represented by a higher level of pheromone concentration. This supervisory agent disseminates information related to the optimal path to others in a homogeneous community. However, it should be mentioned that this technique does not implemented direct supervision, but acts as an encouragement to the others to reach the optimum convergence. It further reduces the danger of a culprit agent misdirecting a whole community of agents which is quite obvious when considering history of human leaders.

Exploration and Exploitation Strategy for the AAANTS Model

An action takes agents from one state to another. In a given state there can be many actions that transfer an agent to other different states. Normally, an agent may tend to take the action with the highest expected reward as per the previous experience. If an agent adopts this type of strategy more frequently, we can call it as greedy and non-exploratory. The research done in this area expose that greedy actions usually contribute towards local optima. Therefore, an agent has to exploit what it already knows in order to obtain rewards, but it also has to explore in order to make better action selections in the future.

The model concentrates on a strategy that employs the proper blend of exploration and exploitation. The dilemma is that neither exploitation nor exploration can be pursued exclusively without failing at the task [SUTT98] and further according to [KAEL96] there are no good, formally justified approaches to this problem either. For example, when a group of agents are released to a grid world defined for foraging, a heuristic can be developed to initially encourage exploration and thereafter converge towards an exploitation policy. However, this strategy would not succeed in an environment where the dynamics change, which would require periodic exploration.

Exploration of actions can be done using methods such as Boltzman distribution (actions selected randomly), pseudo-stochastic choice (Best action or random action chosen), and pseudo-exhaustive choice (Best action or least recently chosen). The experiments of [PIER94] conclude that Boltzman distribution produced worst results in terms of steps to converge to the optimal solution. It is confirmed that on a stochastic task, each action must be tried many times to reliably estimate its expected reward [SUTT98]. Hence, the AAANTS learning model formulates a methodology of using a blend of stochastic and common-sense based patterns to address the exploration-exploitation dilemma.

In a robot experiment problem, [TANG02] confirms that exploration can be made more efficient by dividing the area into sub-areas and having the robots disperse to explore those sub-areas, which in tern induce cooperation. Hence, exploration among spatially distributed collection of agents when converged would provide improvements to the global optimum with reduced effort. We could also apply Nash Equilibrium to restrict unilateral deviation of agents and would cause each agent's choice to be in synergy with all others [HU98].

Action selection strategy for exploitation and exploration are quite different. In exploitation it is much simple where the action with the highest reward in that state is the priority of execution. This action is selected either by remembering past actions or by using a function approximation technique. However, action selection for exploration is a difficult matter altogether. The following is a list of strategies used for exploration in the AAANTS model.

1. The most frequently used actions within the state space are a suitable measurement for selecting an action for exploration. We also need to attach the preference of actions to a given context. This is due to the fact that within the same state space, depending on the context or more precisely the immediate goal state, the most preferred action could change.

   Each action $(a_i \in A)$ in total state space (S) and Context (C)

   $E_c(a_i)$: accumulated expected reward for an action, and $V_c(a_{ix})$: the reward for a specific type of action instance, and therefore $E_c(a_i) = \sum_{x=0}^{n} V_c a_{ix}$ and $[\max E_c(a_i)]$ is the reward for the most globally preferred action. Hence, $[\max E_c(a_i)]$ can be regarded as the most preferred action within the context C.

2. Actions taken to reach the immediate Hub state are a good heuristic for selecting actions for exploration. When a_{ix} refers to a list of actions that can be used to reach the immediate Hub state, $[\max[V_c(a_{ix})]]$ is the most preferred action out of that list. Since Hub states are major decision points for the global success, the

respective actions to reach them can be assumed as highly probable actions to reach the goal.

3. An action can be selected stochastically either from the actions that are available in a given state or from rest of the actions from (1) and (2) excluding the most suitable.

4. Some states can represent a degree of similarity based on different aspects. However, two states in the environment are very unlikely to be exactly similar, though we can only define a degree of similarity. The last strategy is for an agent to invest on the actions taken from states similar to that of the current state.

The AAANTS learning model describes several ways of evaluating the similarity of states. The following strategies are adapted to assess state similarity.

1. Similarity based on action patterns

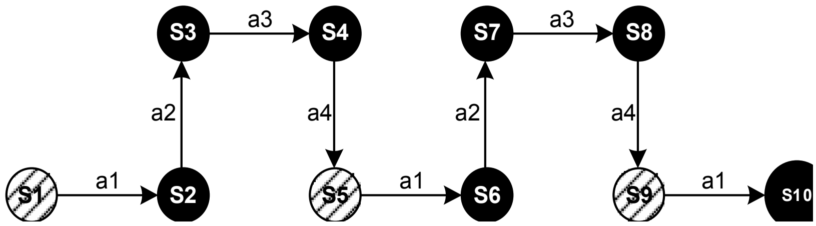


Figure 3: State similarity with action patterns

{(S1, a1), (S2, a2), (S3, a3), (S4, a4)} = {(S5, a1), (S6, a2), (S7, a3), (S8, a4)}

When two or more states initiate a sequence of actions that are similar in the pattern of execution, we could call the initiating and terminating states as similar. With reference to the above diagram (figure 3), states S1, S5 and S9 could be called as similar due to the action pattern {a1, a2, a3, a4}. This type of pattern recognition is a trivial process when considering search space of a single agent.

However, it becomes complicated when considering coordinated concurrent actions of a group of agents. Therefore, the current experiments focus only on individual agent state space.

2. Similarity based on feature patterns

States can be similar based on the attributes inherent to a state (e.g. temperature, pressure, radiation levels, light intensity, etc) or based on the spatial arrangement of objects with respect to the neighbouring states and entities. The neighbouring states can be used to identify spatial and attribute based patterns. The spatial arrangement of neighbouring states of S1 and S2 (Figure 4) based on angle of separation can be used to assess the similarity of states ($Q1 \approx Q2$).

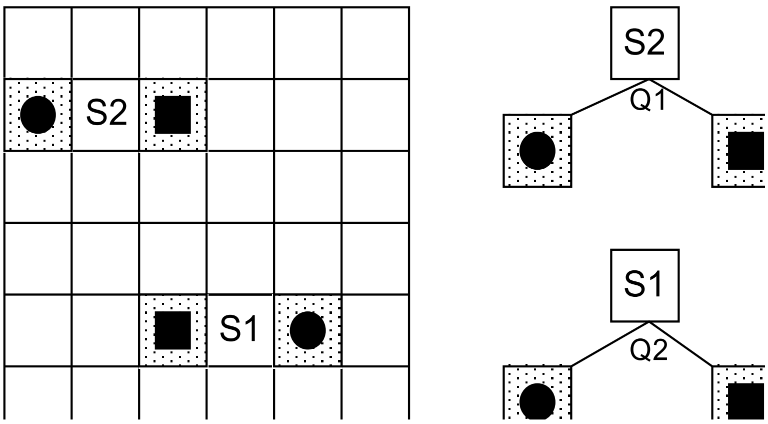


Figure 4: State similarity due to spatial and attribute patterns

3. State values based on rewards

Another successful way to evaluate the similarity of states is by the use of reward values received during the past experiences with the environment. The state values are numerical accumulations of reinforcements received during past experiences within a given domain. States can be grouped into homogeneous categories with the use of value bands.

Coordinated Rewards and Learning Patterns

The previous section discussed the usefulness of Hubs to converge to a globally optimum state pattern. In this section, the Hub concept is further extended for the purpose of conceptualising coordinated behaviour in a community of agents. Therefore, the recognition of Hub states is of special interest to the AAANTS learning model. A Hub is a special state that is recognised as important when compared to its neighbours. A Hub can be created due to reasons such as high state values resulted from episodic reinforcements, special interim rewards before reaching the final goal, local optima, and lastly and most importantly, states that are used by agents to coordinate dependant action execution.

An agent needs to model the behaviour of other agents to contribute to the global well-being. Modelling others in the environment is a complicated task which needs to maintain large volumes of state information. The problem is two folded. Large amount of data is difficult to maintain and further latency of processing such data will hinder behaviour expectations in a real-time environment. The suggestion of the AAANTS model is to maintain the Coordination States or Hub States, of agents in a homogeneous group.

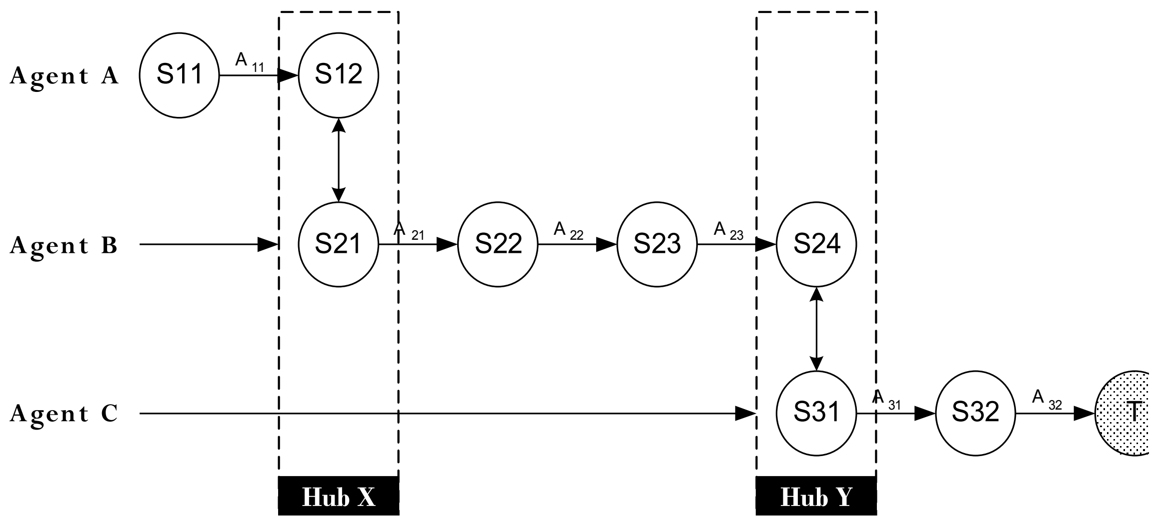


Figure 5: Use of Hub states for agent coordination

When an initiator broadcasts its objectives to achieve a Coordinated State, the agents that need to synchronise at that state would start executing a series of actions to reach it. The above diagram (figure 5) shows a situation where three agents A, B, and C perform a series of actions individually but in a synchronised manner. Continuous experience in the environment has enabled the agents to learn that state pairs S12 and S21 together with S24 and S31 synchronise the elementary actions of the three agents to perform a complex task.

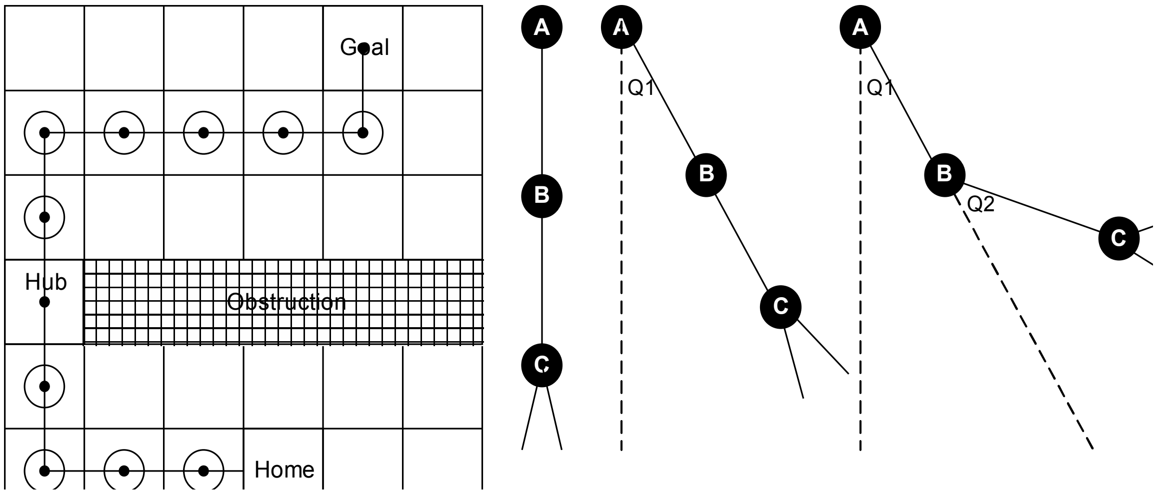


Figure 6: Use of hub states in heterogeneous environments for coordination

The above diagram (figure 6) accurately depicts the use of Hubs by agents in heterogeneous environments. In the grid world the agent moves from home to goal state though a natural obstruction while receiving a reward from the environment. This reward is taken as a special situation and the respective state is marked as a Hub state by the agent. When considering the robotic arm, reference angles among the agents A, B and C, is used as the sensation for coordination. Agent A can move the upper arm and by iterative experience, Q1 angle is taken as a Hub state by agent B to trigger its movement of the lower arm and thereafter, Q2 is considered by Agent C as a Hub state.

The incarnation of a complete multi-cellular infant starting from a single fertilised egg seems like a heavenly secret to all of us. It is the initial set of genes in a fertilised egg that helps a simple cellular growth to be morphed into a complex combination of organs found in a complete animal. It is amazing that every cell contains a complete footprint of all genes found in the initial cell and only represents a single instance of the overall pattern. According to [SALA00], it is considered that there is a pattern if different cells express the same gene at different levels, which can be called as the pattern gene. Thus, most patterns are in fact combinations of a small number of basic patterns [SALA00].

The retention of state information within a homogeneous group of agents could also be described in relation to information patterns in genes. A series of three dimensional cell growths in a predefined pattern could be represented by a chunk of information within a gene. Hence, a gene may contain a sequence of such information chunks that generate cell patterns. It is the combination of cell patterns that create useful organs. The creation of an organ may be done by concurrent creation of cell patterns connected with each other to create greater patterns. The temporal synergy of starting and ending of each pattern within a collection of overlapped execution is of key importance.

Hubs as described earlier are important states using which the achievement of the final goal can be meaningfully descretised. They basically define the start and end of a pattern. A high level task can be achieved by executing a hierarchy of basic patterns that are built on each other. The temporal synchronisation of the concurrent execution of patterns to generate a meaningful higher-level pattern is an immense challenge. The genes have gone though such level of pattern optimisation; however it has taken millions years of evolution for realisation.

The perception layers of a colony of agents could be integrated into a holistic abstraction to have the effects of Perceptual Integration. This is quite different to the coordination discussed in the above sections where the agents perceive the environment individually and coordinate with the use of subsets of state information correlated through Hubs. The McGurk effect is perhaps the most convincing demonstration of the inter-sensory integration where one modality to radically change perceptions in another [COEN00]. Post-perceptual and multi-modal integration are two popular techniques for perceptual integration. Post-perceptual integration occurs in systems where the modalities are treated as separately processed, increasingly abstracted pipelines and the outputs of these pipelines are integrated in a final integrative step where as in multi-modal integration perceptual events are separated from the specific sensory mechanisms that generate them and integrated into higher-level representation [COEN00]. The integration of modalities to produce a unimodal perception that produce a series of hierarchical patterns of sensations is of interest to the AAANTS model.

Implementation

The objective of formulating a generic learning methodology has been a prime objective of this research as mentioned earlier. The concept of AAANTS describes the learning methodology starting from the basic sensation of a state. A sensation can be of various nature such as sound, vision, taste, smell and touch. We should be able to further introduce any new sensation as far as it can be captured and quantified. Complexity is introduced to the AAANTS system with the introduction of a colony of agents that execute actions concurrently for the benefit of the colony. Though each agent being simple and similar in structure, the model tries to prove that collective actions of simple agents paves the way to emergent behaviour. The learning methodology performs an important role of embedding cooperative actions within agents with relation to the states of the environment.

In order to prove the generic nature of the methodology, implementation of several experiments were formulated by the authors. Foraging in a grid world, robotic arm movement and vision controlled navigation are the three major experiments and out of which only foraging related experimentation results were being analysed. Foraging in a grid world was used for two types of experiments, namely, searching for the optimum path for a food source and transporting food items as a collective effort. The robotic arm experiment uses an arm with three joints where each joint is controlled by a single agent. To reach an object, all three joints should be moved in a coordinated manner. The vision navigation experiment uses a camera and two motors to control the direction of movement of a vehicle. Its objective is to reach a destination while avoiding obstacles.

During the foraging experiment, Monte Carlo based single agent optimum path experiment was performed as a reference. The collective search outperforms the single agent generic reinforcement learning experiment in number of iterations for convergence. When the grid world was introduced with obstacles, the use of Hub states makes the collective search superior to single agent search. Further, the advantage of using Hub states becomes apparent when obstacles were altered without modifying the goal state. The agents converge towards the goal state within reasonably less iterations when compared to the generic experiments.

We are currently in the process of conducting analysis on the other two experiments. In the robotic arm movement experiment, the angular movement of each joint with respect to the vertical axis and the adjoining joint is taken as the reference. Therefore, the sensation of this experiment is based on angle in degrees. The same learning methodology used for foraging was applicable to this domain and the early results give a positive picture. The last experiment based on vision navigation is still in the process of implementation and would require considerable amount of time to setup the computer vision related analysis.

Conclusions

The initial implementation of the AAANTS model was experimented without the use of Hubs. The experiments performed on the three domains with the use of Hubs (foraging, robotic arm and vision navigation) have shown considerable improvements when compared with the previous results. The most valuable finding was in the use Hubs for coordination. Though our experiments were restricted for pattern search within individual agents (anyway partner consciousness elements do keep a restricted repository of information related to others in the community), it has shown remarkable improvement over cognitive approaches.

As mentioned earlier, further experiments need to be conducted to confirm the applicability of Hubs to heterogeneous domains. Further, a methodology to identify Hubs across multiple agent instances of a synchronous group is a major improvement expect out of the AAANTS model in the future. Together with this, exploration and exploitation strategy has to be fine tuned to facilitate speedy convergence to dynamic changes in the environment.

References

[BARA02] Albert-Laszlo Barabasi, 2002, Linked, How everything is connected to everything else and what it means for business, science, and everyday life, Published by the Penguin Group, ISBN: 0-452-28459-2.

[COEN00] Michael Coen, 2000, Multimodal Integration – A Biological View, MIT Artificial Intelligence Lab, Cambridge, MA 02139.

[COLO93] Marco Colombetti, Marco Dorigo, 1993, Training agents to perform sequencial behaviour.

[COLO94] Marco Colombetti, Marco Dorigo, 1994, Training agents to perform sequential behaviour, Adaptive Behaviour, MIT Press, 1994.

[DIET00] Thomas G. Dietterich, 2000, Hierarchical Reinforcement Learning with the MAXQ Value function Decomposition, Department of Computer Science, Oregon State University, Corvallis.

[GARD98] Natalia Hernandez-Gardiol, Sridhar Mahadevan, 1998, Hierarchical Memory-Based Reinforcement Learning, Department of Computer Science, Michigan State University.

[GULL92] Vijaykumar Gullapalli, 1992, Reinforcement Learning and its Application to Control, PhD Dissertation, Department of Computer and Information Sciences, University of Massachusetts.

[HU98] Junling Hu and Michael P Wellman, 1998, Artificial Intelligence laboratory, University of Michigan, USA.

[HOLL94] Bert Holldobler and Edward O. Wilson, 1994, Journey to the Ants: a story of scientific exploration, ISBN: 0-674-48526-2.

[JENN95] N.R. Jennings, Michael Wooldridge, Intelligent Agents: Theory and Pratice, 1995, Knowledge Engineering Review Oct. 1994, Revised 1995.

[KAEL96] Leslie Pack Kaelbling, Michael L. Littman, 1996, Reinforcement Learning: A Survey, Brown University Providence and Carnegie Mellon University.

[MATA94] Maja J Mataric, 1994,Interaction and Intelligent Behaviour, Ph.D. Dissertation, MIT, USA.

[MINS86] M.Minsky, 1986,The Society of Mind, ISBN 0-671-65713-5, A Touchstone book, Published by Simon and Schuster, New York.

[PARR97] Ronald Parr, Stuart Russell, 1997, Reinforcement Learning with Hierarchies of Machines, Computer Science Division, UC Berkeley, CA 94720.

[PIER94] Pierguido V C Caironi, Marco Dorigo, 1994, Training and Delayed Reinforcements in Q-Learning Agents, Universite' Libre do Bruxelles.

[RANA02] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2002, AAANTS – Distributed Mobile Component Architecture for an Ant Colony based Synthetic Ecosystem, (Short Paper) MATA'02, Universitat Pompeu Fabra. Barcelona, Spain.

[RANA03] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2003, Enhanced Frame based Knowledge Representation for an Intelligent Environment, KIMAS 2003, 1-3 October 2003, Boston Cambridge MA.

[SALA00] Isaac Salazar-Ciudad, Jordi Garcia-Fernandez, and Ricard V Sole, 2000, Gene networks capable of pattern formation: From induction to reaction-diffusion, Complex Systems Research Group, Department of Physics, FEN-UPC Campus Nord, Barcelona Spain.

[SUTT98] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction MIT Press, Cambridge, MA, 1998, A Bradford Book.

[TANG02] Poj Tangamchit, John M Dolan, Pradeep K Khosla, 2002, The Necessity of Average Rewards in Cooperative Multirobot Learning, Dept. of Electrical Engineering, The Robotics Institute, CMU, USA

Paper 2: AAANTS - A Distributed Agent Framework for Adaptive and Collective Intelligence

R.A. Chaminda Ranasinghe
Department of Computer Science,
University of Colombo,
Colombo, Sri Lanka.
Tel No: +94-777-347220
chaminda@interblocks.com

Ajith P. Madurapperuma
Department of Computer Science,
University of Colombo,
Colombo, Sri Lanka.
Tel No: +94-1-581245
ajith@cmb.ac.lk

Keywords
Multi-agent systems Emergent behaviour Intelligent adaptive systems Distributed agent architectures Ubiquitous computing Synthetic ecosystems.

Abstract

Our work with the AAANTS (Adaptive, Autonomous, Agent colony interactions with Network Transparent Services) project attributes to developing methodologies for demonstrating intelligent behaviour in agent-based synthetic ecosystems. The model gains wisdom from a very successful community life style found in the animal kingdom – the Ants. The proposed model conceptualises and implements a colony of agents that actively interact with a collection of distributed services in order to provide adaptive behaviour. We have implemented the AAANTS model on an Intelligent Environment related project and built a prototype for an intelligent room that actively adapts the environmental conditions for individual behavioural patterns. This paper discusses the architectural and design aspects of the AAANTS model where the prime objective is to provide an agent framework to facilitate and sustain a distributed component based agent colony that depict intelligent and adaptive behaviour.

Introduction

The AAANTS model uses the community life style of insects as the core inspiration and metaphor with further inspiration from "The Society of Mind" theory [MINS86]. The colony of agents are conceptualised by extracting features from a very successful community life style found in the animal kingdom – the Ants. Initially, an agent can be defined as an entity with perceptions, goals, cognition, actions, and domain knowledge, situated in an environment [STON98]. Ants together with many other insect species, is the centre of attraction in the study of artificial life due to their individual simplicity combined with relatively complex group behaviour [PARU97]. Ant colonies have evolved means of performing collective tasks, which are far beyond the capacities of their constituent components. They do so without being hard-wired together in any specific architectural pattern, without central control. The consensus is that comprehension of emergent complexity in insect colonies such as Ants will serve as a good foundation for the study of emergent, collective behaviour in more advanced social organisms, as well as leading to new practical methods in distributed computation [BABA01] [GARC01].

AAANTS model is primarily defined by a distributed colony of agent components that interact with a collection of services. We understand components as an independently deliverable package of software operations that can be used to build applications or larger components [KNAP98]. A service can be any type of a component bundle consisting of hardware and software that is networked with a defined access interface. The agents in AAANTS system are given sensory and actuator capabilities by these distributed heterogeneous services. The services are called heterogeneous since they can be responsible for handling sensory and actuator data of different types such as video, audio, chemical, motion etc. Therefore, the colony can be envisaged as a distributed environment where collection of agents works in synergy while being mobile to specific colony locations during their life cycle [RANA02].

It is clear that a typical implementation of AAANTS model would need to focus on distribution, management and interfacing of agents, facilitators, and distributed services. The discussion in this paper concentrates on the architectural and design aspects of the

AAANTS model where the prime objective is to provide an agent framework to facilitate and sustain a distributed colony of agents. The agent framework concentrates in providing services to the agent colony such as life-cycle management, agent reproduction, colony evolution, fault-tolerance, load balancing and mobility.

Design Considerations

The use of Design Patterns and Frameworks are essential for building systematic execution environments that consist of distributed myriad of components. Design Patterns are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context [GAMM95]. Further, the rationale for using frameworks is to build cooperating collection of entities that make up a reusable design for a specific application domain where it dictates the architecture of applications. A framework defines the overall architecture, segmentation into objects, object collaboration, predefines design parameters common to its application domain and emphasizes design reuse over code reuse. We use Patterns to construct primitive building blocks that are required to build a framework to facilitate design objectives of the AAANTS system.

We now explain the design variables that were taken into consideration for the AAANTS framework. Our selection of design variables was based on those presented by [FARH97] as important variables of an agent system.

<u>Distribution model</u> – The method of distributing application functionality among different types of agents that constitute the agent system.

<u>Internal structure of agents</u> – The internal constituents of an agent and their interoperability to accomplish the desired functionality of an agent.

<u>Control, co-ordination and co-operation</u> – The issues of inter-agent and intra-agent activities, process management and how agents achieve community wide goals.

<u>Communication and knowledge sharing among agents</u> – Inter-agent communication to share knowledge and to exchange information.

Agent environment – Environment that is complemented by the agent system. The sensory information from the environment is useful for agent decisions and their respective actions should be performed on the environment.

Adaptation and continuous improvement – Improvements to the decision making process of agents using the sensory information to provide a better service to the human users.

The Distribution model of the AAANTS system depicts hybrid characteristics of existing multi-agent systems. Each agent in the system is not a fully functional component, though each would contribute towards the overall functionality. The distribution model of AAANTS system also adheres to the concept of "levels of organisation" [FERB99] in which one level can be embedded in another. We shall use the term *elementary component (module)* to refer to units at the lowest level of decomposition (in terms of atomic functionality), and the term *multi-agent system* to refer to a high-level organisation not forming part of an organisation at a higher level than itself *(Figure 1)*. A module is called a lower level agent and in almost all occasions, a homogeneous group of such modules are responsible for a specific category of behaviour (e.g. opening a door, switch on a light and generating alerts).

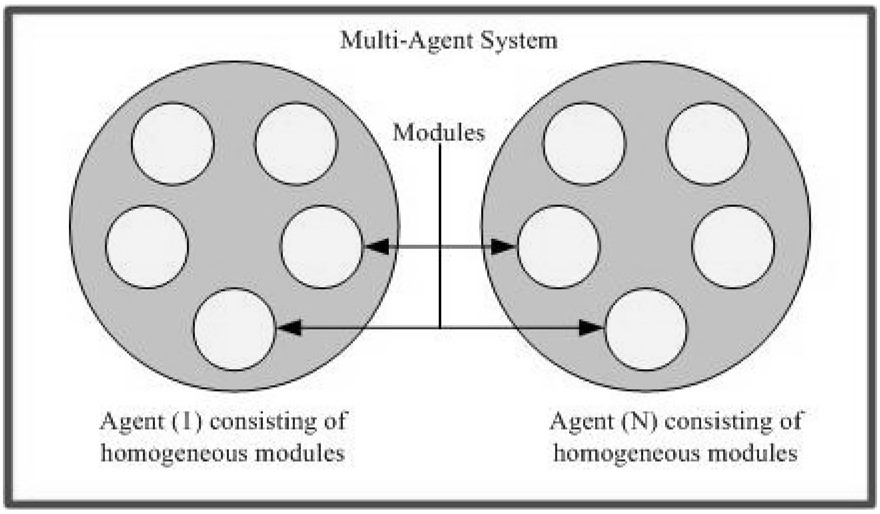


Figure 1: Agent component organisation

Societies considered as being complex, such as colonies of bees or ants, should not be considered as individuals in their own right if we wish to understand their organisation, regulation and evolution phenomena. In terms of multi-agent systems, this means that an organisation can emerge from the juxtaposition of individual actions, without the necessity to define a specific objective [FERB99]. In the AAANTS model, the components are arranged in a manner that the outcome for a defined set of sensor information is generated by complex interactions among agent components.

The environment in which agents operate is represented by a list of distributed services. These services are embedded in the environment, which can be used as neural extensions by the agents for their sensory and actuator functionalities. For example, there are services that perform voice synthesis and recognition, video capturing, motion sensory and robotic services. These heterogeneous services announce their capabilities within the network and interested agents could use them in their goal-driven activities.

In AAANTS framework, communication sub-system is a key necessity for the agents to co-ordinate and work as a group by communicating with each other to exchange current state of execution. It is through communication that a group of agents can arrive at a final set of actions for a particular situation defined by a collection of sensory information.

The design of the AAANTS system has taken into consideration the applicability of cognitive and reactive paradigms. As described in [FERB99], there is actually a continuum between the pure reactive agent, which reacts only to stimuli, and the entirely cognitive agent, which has a symbolic model of the world which it updates continually and based on which it plans all its actions. AAANTS can be described as a hybrid agent system that derives wisdom from both reactive and deliberative architectures.

AAANTS Framework

An agent framework should initially facilitate an environment for inhabitant components to exhibit expected agent characteristics. Accordingly, AAANTS framework provides support for the basic properties of an agent such as autonomy, communication, adaptiveness, goal-orientation, mobility and persistence.

There is already quite a promising collection of architectures conceptualised and implemented in the academic and commercial institutions related to BDI (Belief-Desire-Intension), Blackboard, Subsumption, etc [FERB99]. We have the option of either reusing an existing architecture or designing a new agent framework altogether. When the design objectives of the AAANTS model were considered, it became apparent that existing frameworks could not fully support the design requirements of the model under consideration. Consequently, we have formulated a new architectural framework especially designed to achieve design objectives of the AAANTS model.

A conceptual view of AAANTS architecture is depicted in *figure 2*. The architecture has adopted a layered approach with three functional layers: the Distributed Services layer (DSL), Service Adaptation Layer, and Colony of Software Agents. Each of these layers is composed of an egalitarian collection of components that interface in a dynamic manner. An obvious hurdle is to overcome interfacing among adjacent layers of the architecture while maintaining simplicity and dynamism. As a solution to this, we have resorted to the use of communication middleware that support heterogeneous distributed components to interface with one another in simple and a dynamic fashion. Each of the three layers of this architecture is described below.

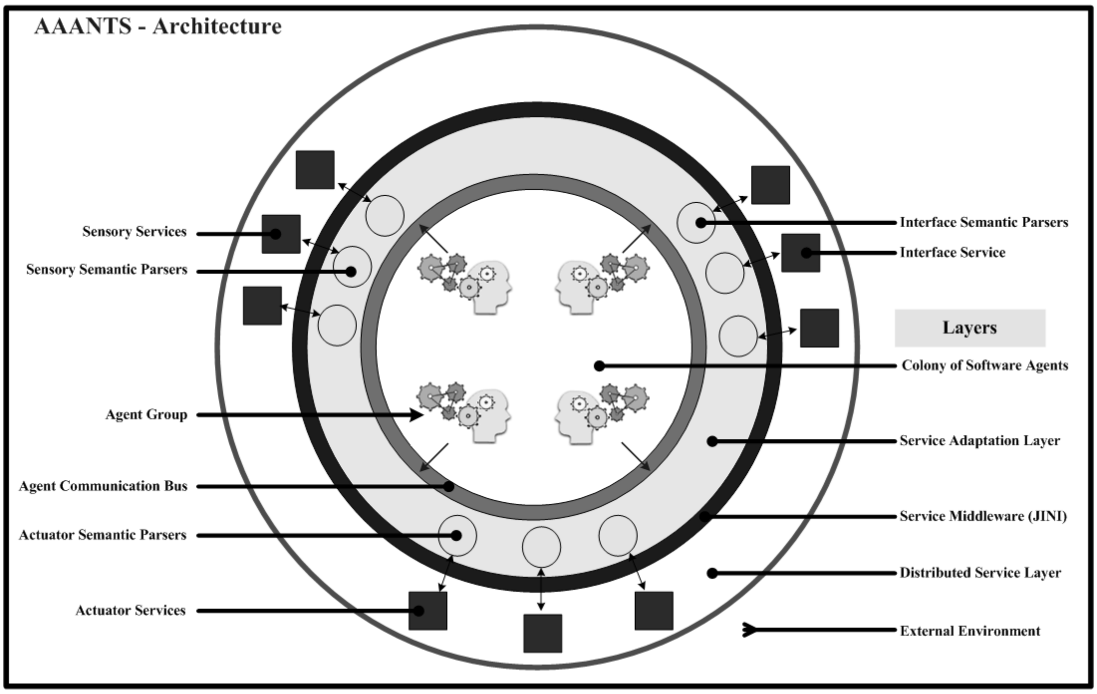


Figure 2: Layered architecture of the AAANTS model [RANA01]

Distributed Services Layer (DSL)

The outer most layers in the AAANTS architecture is called as the Distributed Services Layer (DSL). It is defined by a collection of components that directly interface with the external environment. The DSL contains all the components necessary to sense and affect the natural environment. There are three categories of components in the DSL.

Sensory Services: The architecture uses these components to feed sensory information to the inner layers of the architecture. The sensations are intercepted from different natural mediums such as audio, video, touch, temperature and motion (e.g. Camera, a thermometer, GPS etc.).

Actuator Services: The actuator components are used to enforce agent behaviour on the external physical environment. These components can perform actions that may change the state of different environment mediums (e.g. Motorized robot, wireless devices, home appliances).

Interface Services: The interface components enable the human users to interact and configure the internal state of the runtime model.(e.g. PDA, WWW browser, wireless phones)

The above-mentioned service categories are heterogeneous in nature with reference to their functionality and execution platform (hardware and system software) [RANA99]. This is a prerequisite since the natural environment demands such flexibility and robustness. Heterogeneity introduces many problems during interfacing the DSL to the Colony of Software Agents. These problems are addressed in the Service Adaptation Layer and are discussed next.

Service Adaptation Layer (SAL)

The Service Adaptation Layer deals with the issue of interoperability among the agents and their external environment. It is mainly composed of three component categories: service middleware, service semantic parsers and a distributed shared communication bus. Service middleware concentrates on managing the existence of such as sensory, actuator and interface services. Service middleware is very important for the robustness of the overall architecture since it mange the well-being of unreliable services in noisy environments.

The semantic parsers of service adaptation layer sits in the middle of service middleware and agent communication bus, and performs semantic mapping of the descriptions in both directions (i.e. Colony of Software Agents and DSL). This is a requirement due to the heterogeneity of the sensory, actuator and interface services as described. There exist separate semantic parsers for each sensation, actuator and interface service categories described in DSL.

The agent communication bus is a distributed and shared communication channel built on Message Oriented Middleware (MOM) [TETI99]. It is used to facilitate agent-to-agent and agent-to-service messaging. The content of information exchanged mainly consists of messages addressed to a subject among publishers and subscribers. We define subject as categorisation of an information bundle exchanged among services and agents. The publishers are responsible for formulating a self-describing message and thereafter broadcasting on a predefined subject whereas the consumers listen on a subject to intercept messages of relevance.

Colony of Software Agents

The **Colony of software agents** inhabits the innermost layer of the AAANTS architecture. The agent layer comprises of modules, knowledge toolkits, visualisation tools, utility agents and agent definition toolkit. The Agent Colony Container provides the runtime environment for the agents. Each agent module keeps its own resources and communicate with others using simple self-contained messages via the MOM. Agents that belong to a homogeneous group publish and subscribe messages using predefined set of subjects. Further, with the help of the services in the Agent Colony Container, the agents can periodically improve module fitness by using optimization techniques such as reproduction and evolution.

Implementation

The implementation of the AAANTS framework consists of distributed collection of Agent Colony Containers connected via the communication middleware. The container is a run-time environment that contains and executes agent related components and provides a standard set of services such as component life-cycle management, performance and process distribution management, directory services, deployment management and communication service interfaces.

We have found that creational and structural design patterns such as the Abstract Factory, Adapter, Facade and Builder [GAMM95] are very useful during the design of a container framework where it helps to create a system, independent of how objects are created, composed, and represented. Therefore, in an Agent Co`lony Container, primitive agents of different types are instantiated using a series of factory classes.

We now explain the different segments of components found in a typical AAANTS implementation. ***System Definition Components (SDC)*** is a broad term representing a collection of application tools that can be used to define an initial agent colony. Using the SDC an administrator can initially create an agent colony for a specific purpose and later change the definition to introduce new features to the system. ***System Execution and Control Components (SECC)*** represents the core of the implementation consisting of containers, agents, communication sub-system, and distributed services. ***System Monitoring and Visualisation Components (SMVC)*** *(figure 3)* is a set of tools used by the administrators and end users to interact with an active agent system. The agents found in SECC also use SMVC for user notification and feedback.

The user interaction with AAANTS implementation uses heterogeneous interfaces such as stand-alone clients, web browsers, and PDA clients. These interfaces allow the agent colony to actively communicate with the user for notification, confirmation, and configuration. Users are also responsible for maintaining some of the mobile services such as Global Positioning Systems (GPS), mobile phones and PDAs', in a ubiquitous manner are used by agents for decision-making.

The *(figure 4)* shows a photograph of the AAANTS prototype in operation in an Intelligent Environment (IE) condition for an individual human being. The AAANTS system is implemented in Java on Windows platform. CORBA was used as the communication middleware to facilitate communication among distributed containers and services. Initially we tested the system with JESS and Fuzzy JESS toolkits for implementing adaptive and decision-making properties of the agents. Our final objective is to extend the system so that it can be used to provide an intelligent environment for a group of people that inhabit a common geographical location. Here we have to take into consideration the heterogeneous behaviour of individuals under the same environmental conditions (sensations) and how the AAANTS system can adapt to provide intelligent changes to the environment that are most appropriate to the majority of people.

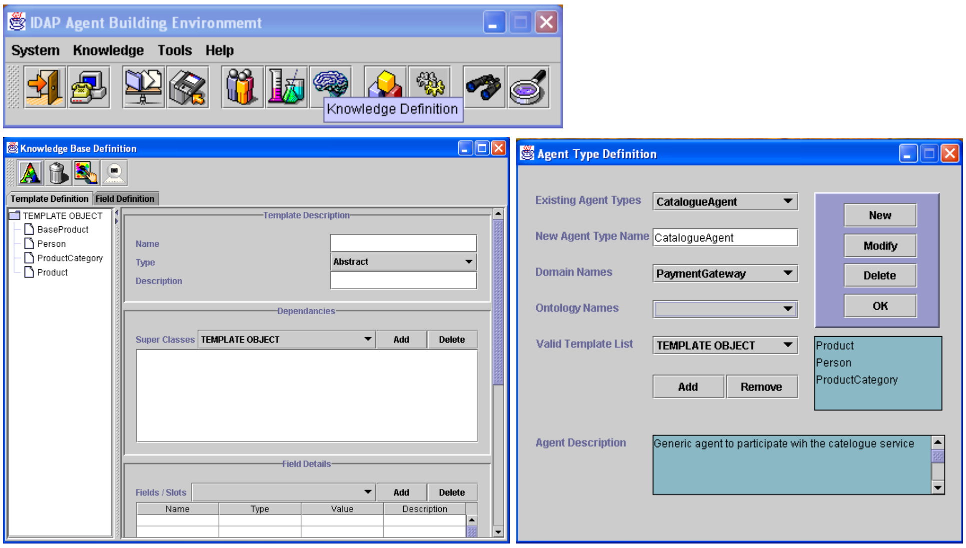


Figure 3: A view of AAANTS SMVC

Related Systems and Technology

Several simulation platforms have similar objectives to the framework presented in this paper such as JADE [BELL98], MadKit [FERB02], Anthill [BABA01], Hive [MINA99] and Amalthaea [ALEX96]. JADE is a software framework that facilitates the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [BELL98]. AAANTS framework uses containers, communication sub-system and facilitator services in a manner similar to that used by JADE but differs in aspects such as the use of CORBA over RMI for communication, use of a simpler inter-agent communication language and the use of reproductive and evolutionary services for periodic optimisations in the overall agent system.

MadKit implementation too has some similarities to AAANTS in terms of its use of containers, communication sub-system and facilitator services. But it is built around the concept of "micro kernel" and "agentification of services" [FERB02]. The MadKit kernel is rather small, but agents offer the important services one needs for its own application such as distribution and remote message passing and monitoring and observation of agents. Most of these services are provided by the Agent Colony

Container (similar to the micro kernel) in the AAANTS model and uses CORBA as distributed middleware instead of proprietary "MadKitdistribution" mechanisms.

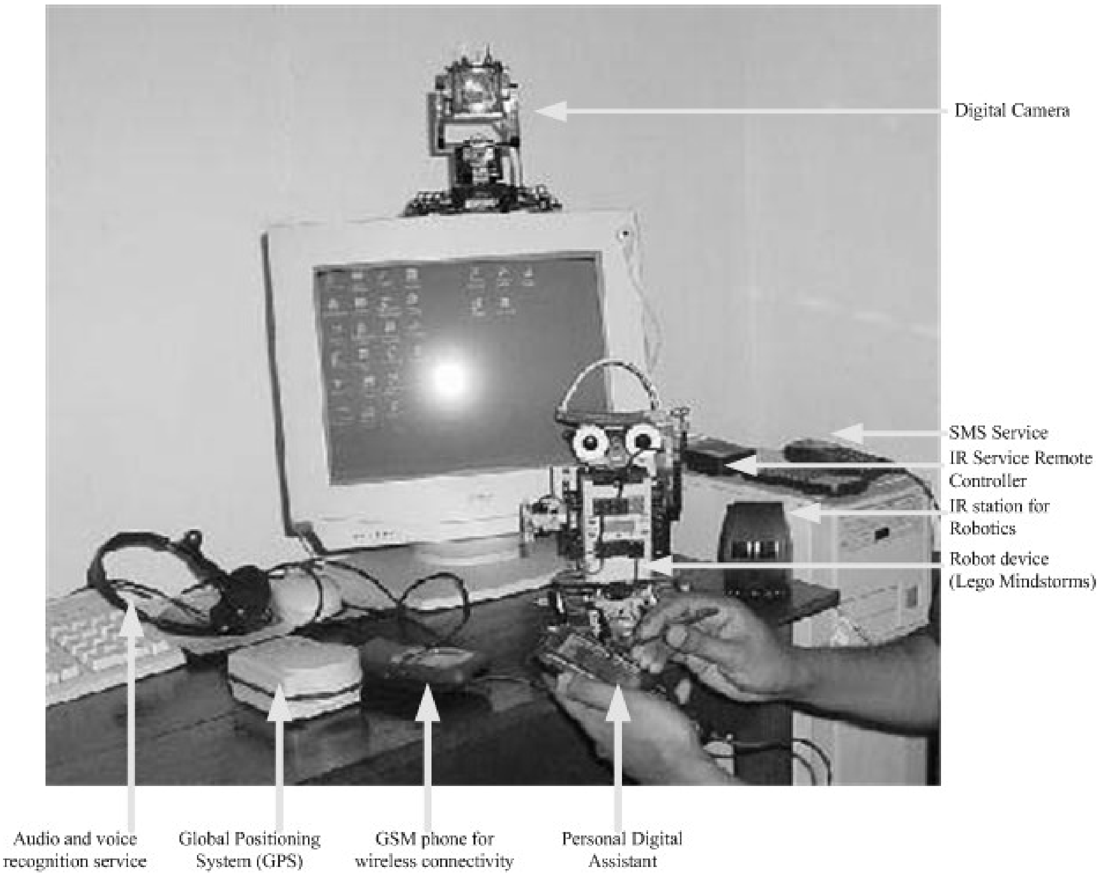


Figure 4: AAANTS prototype for an intelligent environment

Other related implementations such as "Anthill" [BABA01] conceptualises a nest, which is similar to that of the Agent Colony Container. However, the AAANTS container does not distinguish a single instance of a container as a nest, since a single nest can be distributed over multiple containers, which are linked by a communication sub-system.

Conclusions and Implications

In this paper, we have presented the architectural design of the AAANTS model that is inspired from natural ant colony. AAANTS is a general-purpose hybrid agent model since it has the capability to interact with heterogeneous services and has shown remarkable improvements over other functional monolithic agent models in terms of adaptability and knowledge/component reusability.

The AAANTS framework, with the use of a component container will be useful in providing services such as life-cycle management, agent reproduction, colony evolution, fault-tolerance, load balancing and mobility to the agent components. Another advantage of the framework is the separation of common and redundant functionality from agent components to a single layer for common usage. The Agent Colony Container was successful in providing an environment to the agent components to depict characteristics such as autonomy, adaptability, mobility, discourse and responsiveness.

The implementation phase has proven that the use of the Service Adaptation Layer for interfacing has helped to overcome the inter-operability issues among agents and services. It was apparent from the implementation that this layer excludes the need for brokering and matchmaking services present in traditional deliberative architectures, since it offers subject based self-describing messages.

The AAANTS model has helped us to observe emergent behaviour similar to that of a natural ant colony. These agents sense the environment and communicate with others using primitive message constructs to offer emergent adaptive behaviour as a community.

References

[ALEX96] Alexandros Moukas, 1996, Amalthaea: Information Discovery and Filtering using a Multi-agent Evolving Eco System, MIT Media Laboratory.

[BABA01] Ozalp Babaoglu, Hein Meling, Alberto Montresor, 2001, Anthill: A Framework for the development agent-based peer-to-peer systems, University of Bologna, Italy, Norwegian University of Science and Technology, Norway.

[FARH97] Faramarz Farhoodi and Peter Fingar, 1997, Developing Enterprise Systems with Intelligent Agent Technology, Featured in Distributed Object Computing, "DOC" Magazine By Object Management Group (OMG).

[FERB99] Jacques Ferber, 1999, Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.

[GAMM95] Erich gamma, Richard Helm, Ralph Johnson, John Vlissides, 1995, Design Patterns- Elements of Reusable Object-Oriented Software, ISBN: 81-7808-135-0.

[GARC01] Carlos Gershenson García, 2001, Artificial Societies of Intelligent agents**.**

[KNAP98] Michael Knapik, Jay Jhonson, 1998, Developing Intelligent Agents for Distributed Systems, McGraw-Hill, ISBN 0-07-035011-6.

[MINA99] Nelson Minar, Mathew Gray, Oliver Roup, Raffi Krikorian and Pattie Maes, 1999, Hive: Distributed agents for Networking Things, MIT Media Lab.

[MINS86] M.Minsky, 1986, The society of mind, Simon and Schuster, New York, New York. ISBN 0-671-65713-5.

[PARU97] Van Parunak, John Sauter, and Steve Clark, 1997, Toward the Specification and Design of Industrial Synthetic Ecosystems,Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).

[RANA01] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2001, Distributed Network Service Utilisation with Adaptive Ecology of Mobile Agents, INET 2001 the 11th Annual Internet Society Conference, Sweden – Poster Session.

[RANA02] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2002, AAANTS – Distributed Mobile Component Architecture for an Ant Colony based Synthetic Ecosystem, MATA'02 Fourth International Workshop on Mobile Agents for Telecommunication Applications, October 2002, Universitat Pompeu Fabra. Barcelona, Spain.

[RANA99] R.A. Chaminda Ranasinghe, A.P Madurapperuma, October 1999, Distributed Service Utilisation on Heterogeneous Networks using Mobile Agents, International Information Technology Conference, Colombo, Sri Lanka.

[STON98] Peter Stone, 1998, Layered Learning in Multi-Agent Systems, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[BELL98] Fabio Bellifemine, Agostine Poggi, Giovanni Rimassa, 1998, JADE – A FIPA-compliant agent framework.

[FERB02] Jacques Ferber, Olivier Gtknecht, 2002,Madkit User's Guide, version 3.1, http://www.madkit.org.

[TETI99] Frank Teti, Unix Insider, 1999,Understanding Java Messaging and JMS.

Paper 3: Enhanced Frame-based Knowledge Representation for an Intelligent Environment

R.A. Chaminda Ranasinghe, chaminda@interblocks.com

Ajith P. Madurapperuma ajith@cmb.ac.lk

Department of Computer Science, University of Colombo, P.O.Box 1490, Colombo, Sri Lanka. Tel. + 94-1-581245

Abstract

Our work with the AAANTS (Adaptive, Autonomous, Agent colony interactions with Network Transparent Services) project attributes to developing methodologies for demonstrating collective intelligent behaviour in agent-based synthetic ecosystems. The model gains wisdom from a very successful community life style found in the animal kingdom – the Ants. The proposed model is a multi-agent system that conceptualises and implements a colony of agents that actively interact with a collection of distributed services in order to provide adaptive behaviour. We have implemented the AAANTS model on an Intelligent Environment related project and built a prototype for an Intelligent Room that actively adapts the environmental conditions based on user behavioural patterns. The paper discusses the knowledge representation based on "Frames" that harmonise with continuous adaptation based on Reinforcement Learning techniques.

Introduction

Multi-agent systems consist of agents that work in harmony and synergy to achieve community wide goals. An agent can be considered as an entity with perceptions, goals, cognition, actions, and domain knowledge, situated in an environment [18]. The colony of agents discussed in this paper is conceptualised after a very successful community life style found in the animal kingdom – the Ants. Ants together with many other insect species, occupy a central place in artificial life due to their individual simplicity combined with their relatively complex group behaviour [13]. Ant colonies have evolved to perform collective tasks, which are far beyond the capacities of their constituent components. They do so without being hard-wired together in any specific architectural pattern, without central control. The consensus is that comprehension of emergent complexity in insect colonies such as Ants will serve as a good foundation for the study of emergent, collective behaviour in more advanced social organisms [2] [10], as well as leading to new practical methods in distributed computation.

The Society of Mind theory [12] further portrays the mind as a collection of mindless components that enter into competition and interaction to provide reasoning and meaning to the surrounding world. Here too we encounter collective group behaviour and emergent complexity as found in the natural insect colonies.

Knowledge representation is one of the most important considerations while designing and implementing multi-agent systems. There are several successful approaches conceptualised and implemented for knowledge representation such as cognitive, connectionist and interactionist paradigms [23]. Among them the interactionist hypotheses postulates the knowledge being represented inside individual agents within a multi-agent system. The AAANTS model uses a knowledge representation that is derived from the concept of a frame. A Frame is defined as a structure for representing a stereotyped situation or structure that represents knowledge about a limited aspect of the world [22]. An important intuition underlying frame systems is the belief that people cope with new situations by retrieving information that was stored based on previous experiences in situations that were in some sense similar to the present situation [25].

AAANTS is a multi-agent system where each colony is modelled, as a collection of heterogeneous agents distinguished by their differences in ontology, behaviour, knowledge and goals. Heterogeneity creates the problem of defining new agent behaviour for each agent introduced to the existing agent society. This is solved by building a framework for rationalising agent engineering so each agent does not have to be constructed from scratch in ad-hoc ways. In our approach, we have used reusable architectural components with reusable agent behaviours to construct new agents.

Knowledge Representation

Agents within the AAANTS model are responsible for maintaining knowledge specific structures and content required for their individual behaviour. AAANTS knowledge representation methodology has combined frame-based Uniframers and Accumulators [12] to complement the learning achieved through Reinforcement Learning (RL) techniques. Agents keep frames representing the different states of activation. Each state relates to a value function that indicates the expected future rewards that initiates from this state. A correct mapping of a state signal from the environment would trigger an action of the highest expected reward.

An individual agent consists of a Communication Manager, a Knowledge Manager and a Frame-based Knowledge Structure as depicted in *(figure 1)*. The knowledge representation adopts an interacting framework where knowledge is embedded within the individual agents. Each frame represents a stereotype situation or a structure in the environment. Frames are related to each other through object-oriented relationships such as containment, aggregation and inheritance. These relationships are described during the knowledge definition stage and thereafter maintained by Frame Relation Manager during the execution stage.

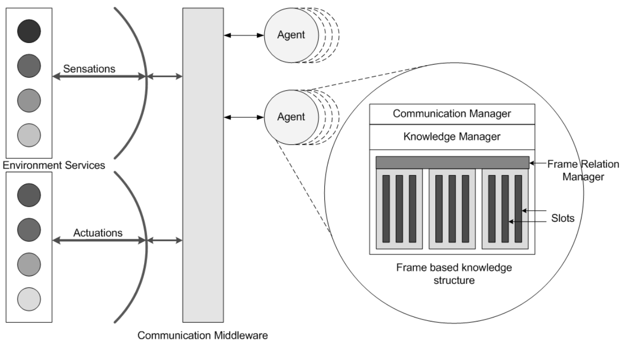


Figure 1: Conceptual knowledge representation using an enhanced frame based representation

Non-Intentional Cooperative Behavior

Agents within the AAANTS model are described as an adaptive and autonomous entity that can sense and act on the environment. The agents in the AAANTS Colony are segmented into groups that share common behaviour and ontologies or in other words, a group of agents is responsible for a spectrum of homogeneous behaviour. For example, the behaviour of activating a lamp can be undertaken by a collection of agents that may be triggered by different environment conditions such as darkness, during user trying to read a book, intruder identification. The state of the environment sensed through the embedded services is responsible for activating suitable agent behaviour [16]. Therefore it become obvious that interaction among agents especially within a homogeneous group is necessary for adaptation and appropriate selection of actions.

Communication is the key facilitator for interaction within entities of an agent system. According to [6], typical agent communication language can be divided into three layers consisting of Content Layer, Message Layer and Communication Layer. We also have used a similar segmentation to handle complexity during agent interoperability *(figure 2)*. Initially a Communication Layer is used for interaction among components such as agents, services, administration and monitoring. This layer combines distributed locations in the agent colony and services embedded within the environment.

The Message Layer consists of encapsulated message packets that contain a header and content information. Sensory signals published by distributed services and actuator signals published by the agents are disseminated in the network on a predefined subject [14]. The subject together with other Meta information is represented in the header portion of the message where a subject simply represents a homogeneous collection of sensations and behaviours that is attached to each message. The agents and services publish, subscribe and intercept messages on a subject of interest. This concept adheres to the Observer Pattern as described in [9].

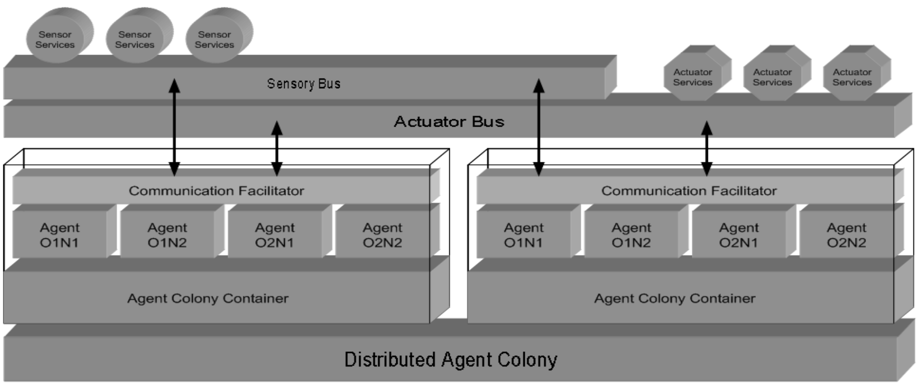


Figure 2: Shared Communication Channel among agent communities

We define the cooperative behaviour of agents in the AAANTS system as non-intentional, due to non-existence of intentional direct communication among agents using an accepted agent communication language. Agent interaction is facilitated by message exchanges disseminated in the network where the interested agents are responsible for intercepting and processing published messages to exhibit further behaviour. This methodology enables information sharing among a group and the ability to influence behaviour on others without explicit knowledge about the participants: thereby making the interactions, non-intentional. We adopted this methodology with reference to the interaction mechanisms found in insect colonies with the use of chemicals such as pheromones in Ants.

Adaptation Techniques

The environment of the AAANTS system approximates Markov state [24]. The objective of achieving Markov state is impractical due to the complexity of sensory information found in a natural environment.

Initially we need to consider the interactions between the agent and the environment. As observed in [3], KQML and FIPA ACL use may be too complicated for the kinds of applications that are envisaged to be present in the non-cognitive based systems as they do not need speech acts and logic to carry out their negotiations. Consequently, we have observed that this argument is applicable to the proposed methodology of interaction in the AAANTS system as well.

Ontology plays a prime role during agent interactions to facilitate proper semantics among the participants. In a multi-agent system, the term Ontology can be given a more practical definition as the declarative knowledge that represents the significant concept attributes and values within the application domain [5]. Ontology defines the basic terms and relations comprising the vocabulary of a topic area or meta-data dictionary, as well as the rules for combining terms and relations to define extensions to the vocabulary [7] [20].

Multi-agent systems based on the reactive paradigm do not make use of ontological basis of knowledge sharing to the extent used by cognitive / deliberative paradigms. The proposed AAANTS model though being simple in terms of depth, flavour and nature of inter-agent communication, use ontological commitments during message exchange among homogeneous groups.

We have used Reinforcement Learning techniques to provide adaptive nature of behaviour to the AAANTS model. In the AAANTS model, the agent-environment interaction is broken down to episodes that are demarcated by an Initial and a Terminal state. There can be several simultaneous on going episodes at a given time. For example, there can be an episode that is responsible for adapting the environment during a user entry to the Intelligent Room. The Initial state is defined by capturing of a geometric figure of a human or unique identification of a user by finger print scanner. Subsequent to the Initial state, the group of agents may depict behaviour to activate lights, air conditioning and other appliances. The Terminal state of the episode may be defined by recognising the user being seated at his/her working position. After the Terminal state, the user can value the actions taken by the system and provide a scalar value to indicate the level of satisfaction. This value acts as the reinforcement to the group of agents that were responsible for the activity. Reinforcement is not always mandatory since it could become a nuisance for the user to supply feedback after each episode. The system is designed to exponentially reduce the interaction frequency for reinforcement with the user.

The sensory signals and respective actions need to be traced in order to assign rewards during reinforcement phase. Therefore, each message signal is assigned a unique identifier for later recognition. However, a given sensory signal can be complemented with a collection of actions executed linearly, contributed by a collection of agents. This situation gives birth to the problem of assigning reward to agents for each atomic activity.

The distribution of actions among several agents and the application of the reinforcement function to democratically distribute reward among them, introduce some complexity to the AAANTS model. We have initially implemented a Pure Delayed Reward class of reinforcement function [19] to assign the average scalar reward to all participant agents for their individual actions after a Terminal state has reached. The Action History Service (AHS) implemented in the AAANTS framework keeps track of all the information related to state, action and reinforcement of each episode. The Reward Assignment Service (RAS) consults the AHS to gather the agent identifications deserving reward since there exists a homogeneous agent group responsible for particular category of behaviour.

The User Interaction Subsystem (UIS) facilitates the user to reward a total episode after the terminal state has reached. Furthermore, the use can provide reward for individual actions carried out with a defined episode. For example, the activation of the air conditioner during evening (assuming the natural conditioning is favourable) can be given negative reward to discourage its inhibition during future interaction of the same episode of entering a room.

Implementation

Implementation of the AAANTS model can be broadly categorised into the development of a distributed agent framework, communication subsystem, ubiquitous services, user interaction modules and knowledge representation and adaptation toolkit.

The Framework and Communication Subsystem

The implementation of the AAANTS framework consists of a distributed collection of Agent Colony Containers connected via communication middleware. The container is a run-time environment that executes agent related components and provides a standard set of services such as component life-cycle management, performance and process distribution management, directory services, deployment management, mobility and communication service interfaces.

We have found that creational and structural design patterns such as the Abstract Factory, Adapter, Facade and Builder [9] are very useful during the design of the AAANTS framework. These patterns help to create a system, independent of how objects are created, composed, and represented. Therefore, within the container, primitive agents of different types are instantiated using a series of factory classes. A single instance of a container is analogous to an insect colony found in the natural world. The model is further extended to distribute a colony among several distributed locations to achieve benefits of a distributed system.

The agents within the AAANTS model are location transparent due to the presence of a communication subsystem. The communication subsystem, as discussed earlier is segmented into Content, Message and Communication Layers together with some system level services. The Communication Layer is implemented using UDP multicasting as it enables information publishers to disseminate a single message to multiple subscribers thereby eliminating redundant retransmission of messages found in a Unicast protocol such as TCP/IP. We have used two multicasting groups to separate messages into sensory and actuator origin. This separation has been intentionally done due to the high traffic rate expected in the sensory channel.

Ubiquitous Services and User Interaction

The environment is represented by services written in different programming languages such as C, C++ and Java due to the availability of APIs. We have selected some basic level sensory services such as Motion Sensor, Sound Sensor, and Infrared Sensor for remote control commands and a Voice Recognition Engine developed on IBM ViaVoice. Each of these services, though heterogeneous in functionality, merges into a single level of interaction through a XML based messaging. We have also developed few actuator services based on a robotic toolkit (Lego Mindstorms), Voice Synthesis Engine (based on IBM ViaVoice), Text message sender for GSM mobile phones and X10 based electronic appliance controller.

The user interaction with AAANTS implementation uses heterogeneous interfaces such as stand-alone, Internet, and PDA clients. These interfaces allow the agent colony to actively communicate with the user for notification, confirmation, and configuration. In the future the users may also be responsible for maintaining some of the services such as Global Positioning Systems (GPS), PDA, and mobile phones in a ubiquitous manner, that are used by agents for decision making and assistance.

Knowledge and Adaptation

We have developed a Toolkit using Java for facilitating RL techniques and frame-based representations. The Toolkit is used by each agent instance within the AAANTS colony. Furthermore, the framework implements some services that complement the adaptive behaviour of agents. The AHS and RAS help during real-time adaptations and the Knowledge Summary Daemon (KSD) periodically summarises the knowledge into Uniframes. KSD is a framework level service that triggers based on memory consumption of frame instances.

Our experiments have shown that an accumulation of frame instances over a period of time has hindered performance and response time and further deviated from real-time feedback. Therefore, we have devised a periodic evolutionary mechanism (KSD) that summarises the accumulated state of each agent to Uniframes.

The Prototype

We have developed a prototype of the AAANTS model to simulate an intelligent environment – “Intelligent Room” (Figure 3). The prototype consists of sensory and actuator services. The testing phase initially used a tool (Pattern Definition Tool) to define different sensory patterns into a pattern script, which could be used by the Pattern Simulation Engine to simulate past user behaviour.

Thereafter, we tested the system within natural environment conditions of a room for a single user interaction. We initially developed an action plan of the behaviours to be carried out by the user together with expected adaptive behaviour from the environment.

We were able to gain a higher level of correlation to forecasted behaviour on real-time user interactions with the system. We have identified that the redundancy of similar frame instances among several homogeneous agents seems to hinder the accuracy and performance of the overall system. We are in the process of improving the functionality of the KSD service to periodically enhance the suitability of agents within the colony by eliminating unfit agents and thereby reducing conflicting frame instances.

Our research also focuses on extending the system to demonstrate a distributed intelligent environment with the interaction of a mobile user. We have developed a prototype of a mobile user interaction application on a PDA with wireless access to the agent system. We have interfaced a Global Positioning System (GPS) to the mobile application to feed location-based sensory information to the agent system. Our objective is to test AAANTS capabilities in providing intelligent behaviour based on past activities performed within defined locations.

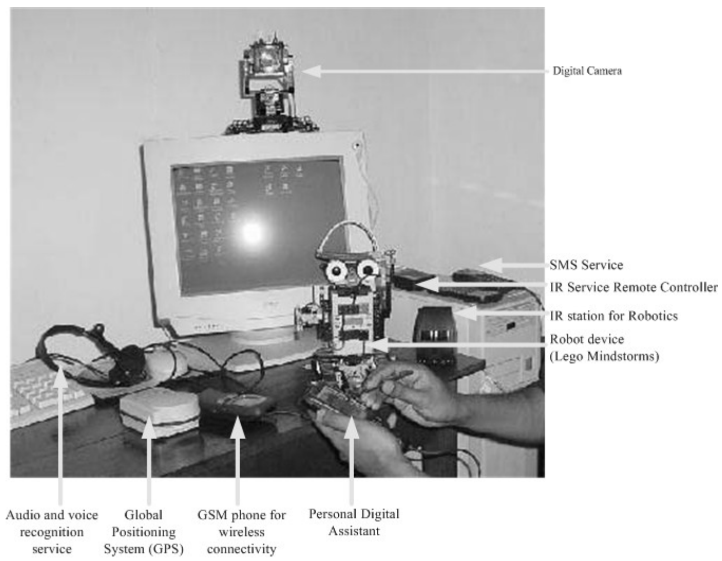


Figure 3: AAANTS simulation and testing environment

Related Work

[1] describes an agent selection mechanism similar to the strategy adopted within the AAANTS model. Here the agents that compose the ecosystem operate under a penalty/reward strategy; supported by the notion of "credit" that is assigned indirectly by the user depending on the system performance. The user gives a feed back on the suitability of an item in the object. The system relates this feedback to the filtering agent that propose the item and the discovery agent that retrieved it and assigns the credit. The AAANTS model is extended further so that the end-user is able to rank actions of many other agents other than the fittest. The argument is that there can be better suggestions from the agent other than the best. But it would be impractical for the end-user to rank all the participating agents every time. Hence the user should be given the facility to change the threshold of the number of agents that he/she can handle. For example a user may decide to interact with only the top three ranked agents during a busy meeting and later decide to evaluate more agents as time permits.

Agent Network for Task Scheduling (ANTS) [17][21], is a distributed agent-based system that can assist human decision making efficiently, allowing material flow and task scheduling to emerge in a manufacturing assembly environment. ANTS uses techniques inspired both by free market economics and insect colonies, specifically a contract net that uses a new mechanism called least Commitment Scheduling that defers decisions on process sequences until the last possible moment.

Several simulation platforms have similar objectives to the framework presented in this paper such as JADE [4], MadKit [8], Anthill [2], Hive [11] and Amalthaea [1]. JADE is a software framework that facilitates the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [4]. AAANTS framework uses containers, communication sub-system and facilitator services in a manner similar to that used by JADE but it differs in aspects such as the use of multicast technique for communication, use of a simpler inter-agent communication language and the use of reproductive and evolutionary services [15] for periodic optimisations in the overall agent system.

MadKit implementation too has some similarities to AAANTS model in terms of its use of containers, communication sub-system and facilitator services. But it is built around the concept of "micro kernel" and "agentification of services" [8]. The MadKit kernel is rather small, but agents offer the important services one needs for its own application such as distribution and remote message passing and monitoring and observation of agents. Most of these services are provided by the Agent Colony Container (similar to the micro kernel) in the AAANTS model and uses multicast based distributed middleware instead of "MadKitdistribution" mechanisms.

Other related implementations such as "Anthill" [2] conceptualises a nest, which is similar to that of the Agent Colony Container. However, the AAANTS container does not distinguish a single instance of a container as a nest, since a single nest can be distributed over multiple containers, which are linked by a communication sub-system.

Conclusions and Implications

In this paper we have presented the theory, design and implementation of a colony of agents in terms of knowledge representation, adaptation and architecture. AAANTS is a general-purpose hybrid agent model that has the capability to interact with ubiquitous services embedded in the environment. AAANTS model has shown remarkable improvements over other functional monolithic agent models in terms of adaptability and knowledge/component reusability.

The core implementation is based on a component based distributed framework. The agent components in the AAANTS model are designed to interact with information sources from heterogeneous domains. We were able to model the information sources as heterogeneous services that actively interface with the core implementation with the help of message based communication middleware.

In the AAANTS architecture, we used an adaptation layer that conceptualises pheromones that act as the sole communication medium in a typical insect colony such as the ants. The implementation has proven that the use of the adaptation layer functionality for interfacing has helped to overcome the conflicts faced in communication among agents and services. It was apparent from the implementation that the adaptation layer excludes the need for brokering and matchmaking services that are present in traditional deliberative architectures.

The AAANTS framework implementation is a distributed component based model that facilitates the well-being of a myriad of agent components. The framework was successful in providing services such as life-cycle management, agent reproduction, colony evolution, fault-tolerance, load balancing and mobility to the agent components. Another advantage of the framework is the separation of common and redundant functionality from agent components to a single layer for common usage.

The current implementation of the AAANTS model has succeeded in distributing knowledge and linearly sequencing actions within an episode among several agents. The user ability to reward an individual action within an episode has enabled to properly adjust the value function of a state so that the sequence of actions adapt to the optimal pattern over a period.

However, we have identified during qualitative testing that there are anomalous behaviour during execution of simultaneous episodes. We have identified that state conflicts during activation of same agent instance contribute to such behaviour. We are in the process of incorporating further evolutionary behaviour that can periodically enhance the agent colony activities. Further we are enhancing the PDA based UIS to introduce a location based service with a help of a GPS.

Finally, AAANTS model has helped us to observe emergent behaviour similar to that of a natural ant colony. These agents sense the environment and communicate with others using primitive message constructs to offer emergent adaptive behaviour as a community.

References

[1] Alexandros Moukas, 1996, Amalthaea: Information Discovery and Filtering using a Multi-agent Evolving Eco System, MIT Media Laboratory.

[2] Ozalp Babaoglu, Hein Meling, Alberto Montresor, 2001, Anthill: A Framework for the development agent-based peer-to-peer systems, University of Bologna, Italy, Norwegian University of Science and Technology, Norway.

[3] Martin Beer, Mark d'Inverno, Michael Luck, Nick Jennings, Chris Preist, and Michael Schroeder, 1999, Negotiations in Multi-Agent Systems, A report as a result of a panel discussion at the Workshop of the (UKMAS'98).

[4] Fabio Bellifemine, Agostine Poggi, Giovanni Rimassa, 1998, JADE – A FIPA-complient agent framework.

[5] Jaron Collis, Divine Ndumu, 1999, The Application Realisation Guide, The Zeus Agent Building Toolkit, Intelligent Research Systems Group, BT Labs. British Telecommunication plc.

[6] Dimitris N. Chorafas, 1998, Agent Technology Handbook, McGrow-Hill series on Computer Communications, ISBN-0-07-011923-6.

[7] Faramarz Farhoodi and Peter Fingar, 1997, Developing Enterprise Systems with Intelligent Agent Technology, Featured in Distributed Object Computing, "DOC" Magazine By Object Management Group (OMG).

[8] Jacques Ferber, Olivier Gtknecht, 2002, Madkit User's Guide, version 3.1, http://www.madkit.org.

[9] Erich gamma, Richard Helm, Ralph Johnson, John Vlissides, 1995, Design Patterns - Elements of Reusable Object-Oriented Software, ISBN: 81-7808-135-0.

[10] Carlos Gershenson García, 2001, Artificial Societies of Intelligent agents.

[11] Nelson Minar, Mathew Gray, Oliver Roup, Raffi Krikorian and Pattie Maes, 1999, Hive: Distributed agents for Networking Things, MIT Media Lab.

[12] Marvin Minsky, 1986, The society of mind, Simon and Schuster, New York, New York, ISBN 0-671-65713-5.

[13] Van Parunak, John Sauter, and Steve Clark, Toward the Specification and Design of Industrial Synthetic Ecosystems, 1997, Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).

[14] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2002, AAANTS – Distributed Mobile Component Architecture for an Ant Colony based Synthetic Ecosystem, (Short Paper) MATA'02, Universitat Pompeu Fabra. Barcelona, Spain.

[15] R.A. Chaminda Ranasinghe, 2002, AAANTS – An Intelligent Synthetic Ecosystem, IITC 2002, Colombo Sri Lanka.

[16] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 1999, Distributed Service Utilisation on Heterogeneous Networks using Mobile Agents, IITC 1999, Colombo, Sri Lanka.

[17] John A. Sauter, H. Van Dyke Parunak, 1999, ANTS in the supply chain, The original version of this paper was presented at the Workshop on Agent based Decision Support for Managing the Internet-Enabled Supply Chain, Agents 99, Seattle, WA.

[18] Peter Stone, 1998, Layered Learning in Multi-Agent Systems, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[19] Richard S. Sutton and Andrew G. Barto, 1998, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, A Bradford Book.

[20] Katia Sycara, Matthias Klusch, Seth Widoff, Jianguo Lu, 1999, Dynamic Service Matching Among Open Information Environments, The Robotics Institute, Carnegie Mellon University, U.S.A. Computer Science Department, University of Toronto, Canada.

[21] Tihamer T. Toth-Fejel, 2000, Agents, Assemblers, and ANTS Scheduling Assembly with Market and Biological Software Mechanisms, Environmental Research Institute of Michigan, Center for Electronic Commerce, Paper for the Seventh Foresight Conference on Molecular Nanotechnology.

[22] Marvin Minsky, 1974, A Framework For Representing Knowledge, Massachusettes Institute of Technology, A.I. Laboratory.

[23] Jacques Ferber, 1999, Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.

[24] S.D. Whitehead, L. J. Lin, 1994, Reinforcement learning in non-Markov environments, Artificial Intelligence.

[25] Han Reichgelt, 1991, Knowledge Representation: an AI perspective. ISBN: 0-89391-590-4.

Paper 4: Non-intentional Cooperative Behaviour for an Agent Based Intelligent Environment

R.A. Chaminda Ranasinghe
University of Colombo, School of Computing,
No. 35, Reid Avenue,
Colombo 7, Sri Lanka.
+94-777-347220
chaminda@interblocks.com

Ajith P. Madurapperuma
Department of Computational Mathematics,
Faculty of IT, No 100/A, D S Senanayake Mawatha, Colombo 8, Sri Lanka.
+94-74-619773
ajith@itfac.mrt.ac.lk

Keywords: Multi-agent systems, Emergent behaviour, Intelligent Adaptive Systems, Distributed agent architectures, Ubiquitous computing, and synthetic ecosystems.

Abstract

Our work with the AAANTS (Adaptive, Autonomous, Agent colony interactions with Network Transparent Services) project applies knowledge gathered from the study of natural community life styles, for e.g. Ants, to developing methodologies for intelligent behaviour in agent-based synthetic ecosystems. This paper discusses the feasibility and implications of using non-intentional interactions among entities in a multi-agent system to coordinate collective behaviour as opposed to agent interaction techniques adapted in common deliberative multi-agent systems. The implementation of AAANTS model within the framework of an Intelligent Environment has confirmed the ability of multitude of loosely coupled egalitarian collection of agents to depict adaptive cooperative behaviour using a non-intentional communication model.

Introduction

Ants like many other insect species, occupy a central place in artificial life due to their individual simplicity combined with their relatively complex group behaviour [12]. Ant colonies have evolved means of performing collective tasks, which are far beyond the capacities of their constituent components. They do so without being hard-wired together in any specific architectural pattern, without central control.

According to [7], the amazing success of the Ants is due to the swiftly applied and overwhelming power arising from the cooperation of colony members. Ants, like humans, succeed because they talk so well [7]. Further, an Ant colony can be regarded as a super organism where it can be analysed as a coherent unit and compared with the organism in design of experiments, with individuals treated as the rough analogues of cells [7].

AAANTS model conceptualises a multi-agent system, i.e. a *Colony of Agents*, consisting of autonomous software components resembling agent characteristics that work in harmony and synergy to achieve community wide goals [13]. At present, an introductory level definition can be given to an agent as an entity with perceptions, goals, cognition, actions, and domain knowledge, situated in an environment [17]. AAANTS model uses the community life style of insects as a metaphor with further inspiration from "The Society of Mind" theory [10]. The components in the AAANTS model can be broadly segmented into a collection of agents and a distributed collection of embedded services. The services act as a neural extension to the agents in providing real-time sensory information from the natural environment.

We have positioned AAANTS as a hybrid model due to the presence of features from both deliberative and reactive paradigms. The hybrid nature of the system is prominently demonstrated in areas of knowledge representation, agent interaction, and adaptive nature in terms of learning and periodic evolution.

The AAANTS model has gone through several stages of modelling, framework development, knowledge-representation techniques, learning methods and cooperation strategies, during the past. We have found out that the coordination strategy remains as the core focus during any team work environment and other methodologies should be moulded to complement it. In the AAANTS model, we try to make the interaction simple by eliminating explicit active communication adapted by most deliberative agent models [14]. In the rest of the paper, we describe the nature of communication found in the AAANTS model and how it contributes to the overall cohesiveness and synergy expected by a multi-agent system.

Reasons for Cooperation?

Cooperative behaviour among collection of individuals has been the cornerstone for the success of human beings' ability to conquer complexity. This is evident when analysing many important historical moments ranging from wars to innovative designs. A multi-agent system too, is composed of several units of autonomous entities that interact to achieve a collective goal. Without cooperation, an agent is merely an isolated individual, closed into its perception-deliberation-action loop [4]. Therefore, co-operation is an important factor to the success in a multi-agent system.

Further, we need to clarify the ambiguity of the terms co-operation and co-ordination. Co-ordination is a process which agents engage in order to ensure a community of individual agents act in a coherent manner. Co-ordination, in turn, may require co-operation; but it is important to emphasize that co-operation among a set of agents would not necessarily result in co-ordination; indeed, it may result in incoherent behaviour [11]. According to [11], the reasons for co-ordination are, preventing anarchy or chaos, meeting global constraints, distributed expertise, resources or information, dependencies among agent actions, and efficiency.

Mechanisms for Cooperative Behaviour in Multi-agent Systems

Communication facilitates sharing of intelligence, negotiations, collaboration and co-ordination. Software agents use a communication language for similar purposes. The main reason for communication may vary depending on the purpose of an agent's existence. The main substance of agent communication is defined in an Agent Communication Language (ACL) [8]. An ACL enables software agents with ontological [3] [18] similarities to communicate with each other via an extensible set of "performatives" expressing beliefs and attitudes towards some information elements. A performative specifies the format of any given message and dictates how an agent should respond to messages. Two popular communication languages are the Knowledge Query and Mark-up Language (KQML) and Agent Communication Language (FIPA ACL) [8] [9] [2].

The break down of predefined tasks found in cognitive agents can be managed by centralising the allocation process or by distributing it among all the agents concerned. The centralised and distributed approaches are concerned with the allocation of tasks by cognitive agents capable of intentionally communicating with each other [4]. In contrast, reactive agents use the concept of signals, which are non-intentional forms of communication, sent by diffusion and propagation into the environment. The proposed AAANTS model conceptualises its communication model based on these elementary form of communication found in reactive paradigms.

According to [6] [16], Agent Communication Languages can best be thought of as consisting of three parts - its vocabulary, an ``inner language'' such as KIF (Knowledge Interchange Format), and an ``outer'' language such as KQML or FIPA-ACL. For example an ACL message can be a KQML expression in which the ``arguments'' are terms or sentences in KIF formed from words in the ACL vocabulary.

According to [1], KQML and FIPA-ACL use may be too complicated for some kinds of applications that do not need speech acts and logic to carry out their negotiations. We embrace this observation for the proposed methodology of interaction in the AAANTS model. The AAANTS model possesses capabilities to simplify the coordinated interaction by eliminating explicit active communication adapted by most deliberative agent models.

Multi-agent systems based on the reactive paradigm do not make use of ontological basis of knowledge sharing to the extent used by cognitive / deliberative paradigms. The proposed AAANTS model though being simple in terms of depth, flavour and nature of inter-agent communication, uses ontologies during communication.

Non-Intentional Cooperative Behaviour

The agents in the AAANTS Colony are segmented into groups that share common behaviour and ontologies. In other words, a group of agents is responsible for a spectrum of homogeneous behaviour. For example, the behaviour of activating a lamp is undertaken by a collection of agents that may be triggered by different environment conditions such as darkness, during user trying to read a book or intruder detection. Therefore, the state of the environment sensed through the embedded services is responsible for activating suitable agent behaviour.

Message Structure

According to [2], a typical agent communication language can be divided into three layers consisting of Content Layer, Message Layer and Communication Layer (Figure 1). In the AAANTS model too, we have used similar segmentation to handle complexity during agent interoperability. Initially a communication layer is used for interaction among components such as agents, services and administrator/monitoring tools. This layer combines distributed locations within the agent colony and services embedded in the environment.

The Message Layer consists of encapsulated message packets that contain a header and content information. Sensory signals published by distributed services and actuator signals published by the agents are disseminated in the network on a predefined subject. The subject together with other meta information is represented in the header portion of the message. The agents and services could publish, subscribe and intercept messages on a subject of interest. This concept adheres to the Observer Pattern as described in [5]. A subject simply represents a homogeneous collection of sensations or behaviours that is attached to each message. Subjects are organised in a hierarchy so that a consumer listening to a parent subject may intercept all inherited messages classified under the parent and can be formulated as listed below.

$S1 = \{x: x \text{ is a subject}\}$

$S2 = \{x: x \text{ is a subject}\}$

$S2 \subseteq S1 \Leftrightarrow (\forall x, x \in S2 => x \in S1)$

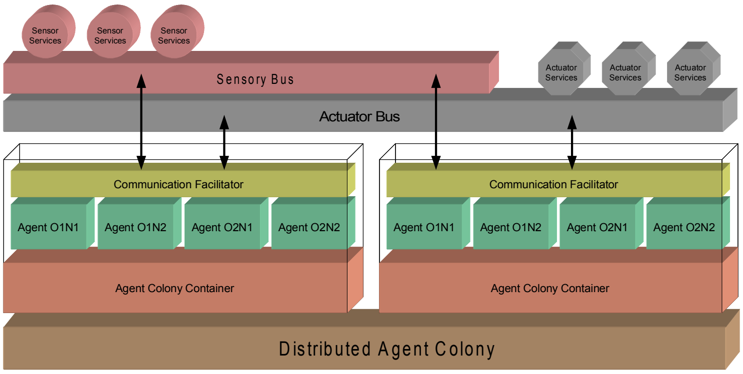


Figure 1: Agent interaction using sensory and actuator messages.

Published messages are not naturally retained within the network for later consultation. Therefore, the AAANTS framework has provided a service called Message Queue Server (MQS) to retain the history of published messages. This essentially acts as a repository of all sensations and actuator messages that have taken place with in a specific period in time. Agents can communicate with the MQS to query recent patterns of data. Each agent undergoes an adaptation stage with the intention of improving their behaviour to evolving environment conditions. The information stored in the MQS performs an analogous function to that of pheromones used in insect colonies. Pheromones are chemicals deposited by individual insects in order to exchange information among individuals and are evaporated temporally. Similarly, the sensory and actuator information captured by MQS are dissipated temporally.

Knowledge Representation

Agents as discussed earlier, are autonomous entities that respond to environment sensations while maintaining coherent knowledge structures relevant for its behaviour. Agents in the AAANTS model perform the inference related activities individually by matching information gathered from the surrounding with the frame-based knowledge structures in possession [15]. Further, these frame-based knowledge structures are modified using Reinforcement Learning techniques based on varying environment state that reinforce a certain behaviour. Actuator channels too can be used as input to agents since behaviour of some agents can act as sensations to others. For example, activating the behaviour of "opening of a door" can act as a sensation to trigger activity on other services such as lighting, air conditioners, electric appliances etc.

Agent behaviour naturally does not solely depend on another for activation, since other environment conditions needs to be consulted. For example, a group of agents may have adapted to a relationship of a human entering a room in summer with that of activating the air conditioner. Such basic behaviour makes agents naively adapt repetitive patterns without considering other complementary factors in the environment. A better solution would be to gather other complementary variables from the environment and adapt to changing situations in a real-time manner. With reference to the above example, it would be more appropriate to activate the air conditioner taking into consideration the environment variables such as temperature, humidity, time, and other predictive behaviour. In addition, the user in weekends might spend only few minutes in the specified environment for some mundane activities and might not need the air conditioner to be turned on.

Reason for Non-Intentional Behaviour

We describe the cooperative behaviour of agents in the AAANTS model as "non-intentional", since there does not exist any intentional direct communication among agents using an accepted agent communication language. Agent interaction is facilitated by message exchanges disseminated in the network where the interested agents are responsible for intercepting and processing the published messages to exhibit further behaviour. This methodology enables information sharing among a group and the ability to influence behaviour on others without explicit knowledge about the participants: thereby making the interactions, non-intentional. We find this methodology having resemblance to the interaction mechanisms found in insect colonies with the use of chemicals such as pheromones.

AAANTS Coordination Model

As discussed earlier, we have defined the AAANTS system as a multi-agent system where a community of agents achieves goals collectively. So the agents will time to time submit individually decided actions to overcome needs of the community. When several urgent needs occur at once, there must be a way to resolve conflicts. One scheme for this might use some sort of central market place, in which the urgencies of different goals compete and the highest bidder takes control [10]. This strategy may fail since extent of achievement in the selected goal may not be defined. Another way is to use an arrangement called cross-exclusion, which appears in many portions of the brain [10]. In such a system, each member of a group of agents is wired to send "inhibitory" signals to all other agents of that group which makes them competitors. When any agent of such a group is aroused, its signals tend to inhibit others. This leads to an avalanche effect, as each competitor grows weaker; its ability to inhibit its challengers also weakens. The result is that even if the initial difference between competitors is small, the most active agent will quickly lock out all the others.

So cross-exclusion is one of the methods that can be used to regulate levels of activities in an agent society. But cross-exclusion can make some selected goal to totally dominate the agent functionality through inhibition.

The AAANTS model is composed of a collection of agents, each responsible for a defined type of activity. For example, with reference to the *(Figure 2)*, the movement of a Robot with four wheels and two motors on either side is controlled by four basic behaviours such as forward, turn left, turn right and stop. These four movements could be sequenced in various permutations to depict wide range of synchronised and intelligent activities.

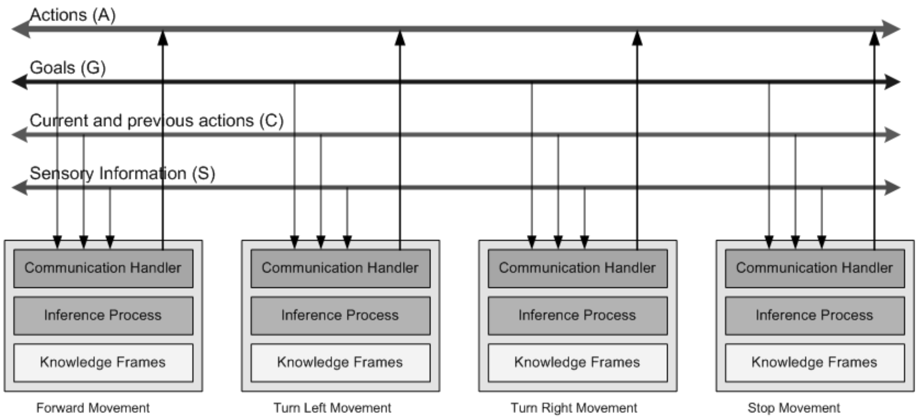


Figure 2: AAANTS Coordination Mechanism

We have discussed the AAANTS communication model as non-intentional due to absence of direct intended communication among agents. Therefore, the synchronisation related information should be kept at each agent that participates in an emergent behaviour. This information is stored in frame based knowledge structures located at each agent. The information related to sensations, goals, current and previous actions are continually published through the communication channel. When an agent intercepts these signals through the communication handler, these are matched against the knowledge structures by the inference process. The inference process should select the most appropriate behaviour.

The inference process would take into account the current goal, on-going and previous activities and environment sensations. The same behaviour for example "Move Forward" can be depicted under different goals such as object tracking, move an object from source to destination or in order to reach the power source for recharging.

Implementation of the Non-intentional Model of Communication

The implementation of the AAANTS model initially has focussed on developing a framework to facilitate the design objectives. The framework mainly focuses on providing facilities to a multi-agent system such as process management, communication, agent life-cycle management, persistence management, mobility and security to a multi-agent system [14]. Implementation details of communication sub-system are of main concern within this paper.

Communication Layer

As described earlier, the AAANTS communication subsystem can be segmented into Content, Message and Communication Layers together with some system level services. The Communication Layer is implemented using UDP multicasting. Multicasting enables information publishers to disseminate a single message to multiple subscribers thereby eliminating redundant, retransmission of messages found in a unicast protocol such as TCP/IP. We have used two multicasting groups to separate messages into sensory and actuator origin as depicted in (figure 1). This separation has been intentionally performed due to the high traffic rate in the sensory channel.

Message Layer

The Message Layer is placed on top on the Communication Layer to provide proper encapsulation of sensory and actuator signals as messages. The messages are created by the publishers and intercepted by the subscribers. Each message contains a header and a data portion. The header contains information such as subject, originator ID, verification data and sequence number. The main publishers of the sensory channel are the heterogeneous sensory services that capture environmental sensations such as real-time video, audio, voice recognition, temperature and motion. The adjective "heterogeneous" is intentional in generalising the services because of variety of sensations, platforms and application programming languages (C, C++ and Java). The primary consumers of the sensory channel are the egalitarian collection of agents that relentlessly listen for messages published by the sensory services. The next channel as depicted in figure 1 is actuator channel, which mainly carries actuator signals published by the agents. The agents perform real-time processing of signals against their knowledge bases to publish the inference as messages that can activate behaviour in processes embedded in the environment, called actuator services. The agents that belong to a homogeneous group recursively become listeners to the messages in the actuator signal channel. This enables agents to give real-time sequence of inter-dependent activities that has been learned in the past.

Content Layer

The Content Layer is embedded with the Message Layer and mainly focuses in the data portion of the message. The content is based on XML that has the natural advantage of describing various types of content. Both agents and services possess XML parsers to create and extract information from the content layer. We have included further functionalities in the XML parsers used by agents to handle disparate and unpredictable patterns of content.

The Prototype

We have initially created several wrappers in C++ and Java to represent the Communication Layer that handle multicast messaging within a distributed network. These wrappers adhere to both Proxy and Observer Pattern as described in [5]. A message wrapper is used by another library that offers a façade to construct and extract messages that represents the message layer, which is further extended to handle XML content manipulation. These libraries were further amalgamated with other processes to build up the AAANTS framework.

We have developed an initial simulation *(Figure 3)* to represent an Intelligent Room, using several sensory and actuator services. We have selected some basic level sensory services such as motion sensor, sound sensor, vision camera for gesture recognition, infrared sensor for remote control commands and a voice recognition engine developed on IBM ViaVoice engine. Each of these services, though heterogeneous in functionality, merges into a single level of interaction because of XML based content layer. We have also developed some actuator services based on a robotic toolkit (Lego Mindstorms), voice synthesis engine (based on IBM ViaVoice), text message sender and X10 based electronic appliance controller.

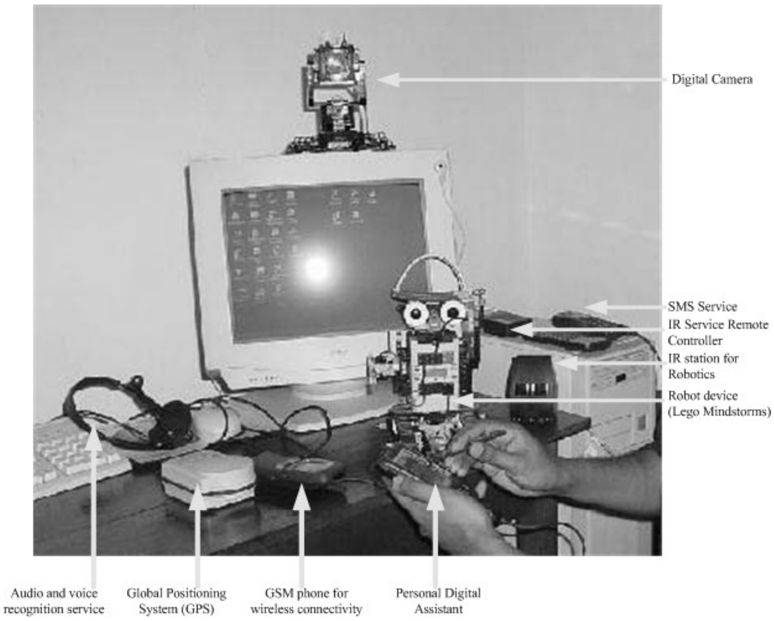


Figure 3: Testing and Simulation Environment for the Intelligent Room Project.

We have tested the current implementation on a limited functionality "Intelligent Room" project to condition the environment state depending on user behaviour. We developed devices to gather environment state information such as motion (user entering or leaving the room and movement to specific parts of the room), temperature, light intensity, noise levels, and speech recognition as primary sensory inputs. The actuations or changing of environment state is carried out by using typical appliances such as lights, air conditioners, fans, televisions, and radio/music players. We initially choreographed episodes of typical human behaviour against a predefined sequence of changing environmental states. Thereafter, collected behavioural patterns exhibited by the AAANTS implementation for the same scenarios. The data gathered showed strong correlation to the expected results.

Future Work and Conclusions

We have already performed initial testing using these services to stimulate cooperative behaviour amongst a colony of agents with in a single colony container. The test has shown favourable qualitative results in terms of cooperative behaviour within a single colony with low intensity of real-time sensory signals. We have seen some conflicting behaviour with high intensive environments in terms of activity sequence and correctness. We have also seen duplicate behaviour within a homogeneous community of agents that depict unexpected fuzzy behaviour.

We have devised several methodologies to overcome the deficiencies found in the initial testing phase. We are in the process of enhancing the reinforcement-learning techniques used for stimulating further adaptive behaviour of the agents. In addition, to overcome conflicting behaviour, it is favourable to periodically facilitate evolutionary and reproductive activities to create new flavour of agents and to eliminate individuals that perform poorly over a period of time.

Further, we are in the process of testing the implementation with the smart navigation of a robotic vehicle using computer vision techniques. We are confident that lessons learned from the AAANTS model would contribute to further clarify the understanding of emergent behaviour based on common-sense reasoning.

References

[1] Martin Beer, Mark d'Inverno, Michael Luck, Nick Jennings, Chris Preist, and Michael Schroeder, 1999, Negotiations in Multi-Agent Systems, A report as a result of a panel discussion at the Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS'98)

[2] Dimitris N. Chorafas,1998, Agent Technology Handbook, McGrow-Hill, Series on Computer Communications, ISBN-0-07-011923-6.

[3] Faramarz Farhoodi and Peter Fingar, 1997, Developing Enterprise Systems with Intelligent Agent Technology, Featured in Distributed Object Computing, "DOC" Magazine By Object Management Group (OMG).

[4] Jacques Ferber, 1999,Multi-Agent Systems, An Introduction to Distributed

Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.

[5] Erich gamma, Richard Helm, Ralph Johnson, John Vlissides, 1995, Design Patterns- Elements of Reusable Object-Oriented Software, ISBN: 81-7808-135-0.

[6] Michael R. Geneserath, Narinder P. Singh, Mustafa A. Syed, 1994, A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation, Stanford University.

[7] Bert Holldobler, Edward O. Wilson, 1990,The Ants,ISBN: 0-674-04075-9

[8] Timothy Lacey, Scott A. DeLoach, 2000, Automatic Verification of Multi-agent Conversations, Eleventh Annual Midwest Artificial Intelligence and Cognitive Science Conference, University of Arkansas, Fayetteville.

[9] Timothy Lacey, Scott A. DeLoach, 2000, Verification of Agent Behavioral Models, The 2000 International Conference on Artificial Intelligence (IC-AI'2000) Monte Carlo Resort, Las Vegas, Nevada.

[10] M. .Minsky, 1986, The society of mind, Simon and Schuster, New York, New York. ISBN 0-671-65713-5.

[11] Hyacinth Nwana & Divine Ndumu, 1996, An Introduction to Agent Technology, Intelligent Systems Research Applied Research and Technology, BT Labs.

[12] Van Parunak, John Sauter, and Steve Clark, 1997, Toward the Specification and Design of Industrial Synthetic Ecosystems, Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).

[13] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2002, AAANTS – Distributed Mobile Component Architecture for an Ant Colony based Synthetic Ecosystem MATA'02, Fourth International Workshop on Mobile Agents for Telecommunication Applications, Universitat Pompeu Fabra.

Barcelona, Spain.

[14] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2003, AAANTS-A Distributed Agent Framework for Adaptive and Collective Intelligence, Mobile Agents for Telecommunication Applications (MATA'2003), IEEE/IFIP, Marrakech, Morocco.

[15] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2003, Enhanced Frame Based Representation for an Intelligent Environment, (KIMAS'2003)

International Conference on Integration of Knowledge Intensive Multi-Agent Systems, IEEE Boston Section, Cambridge MA.

[16] Narinder P. Singh and Mark A. Gisi, 1995, Coordinating Distributed Objects with Declarative Interfaces, Computer Science Department, Stanford University, Stanford, California, Software Technology Lab, Hewlett-Packard, Laboratories, Palo Alto.

[17] Peter Stone, 1998, Layered Learning in Multi-Agent Systems, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[18] Katia Sycara, Matthias Klusch, Seth Widoff, Jianguo Lu, 1999, Dynamic Service Matching Among Open Information Environments, The Robotics Institute, Carnegie Mellon University, U.S.A. Computer Science Department, University of Toronto, Canada.

Paper 5: AAANTS – Distributed Mobile Component Architecture for an Ant Colony Based Synthetic Ecosystem

R.A. Chaminda Ranasinghe
Department of Computer Science
University of Colombo, P.O.Box 1490
Colombo, Sri Lanka
+ 94 77 347220
chaminda@interblocks.com

Ajith P. Madurapperuma
Department of Computer Science
University of Colombo, P.O.Box 1490
Colombo, Sri Lanka
+ 94 1 581245
ajith@cmb.ac.lk

Keywords: Mobile agents, Multi-agent systems, distributed agent architectures, ubiquitous services, ant colony based agent platforms and synthetic ecosystems.

Abstract

AAANTS (Adaptive Autonomous Agent colony interactions with Network Transparent Services) model is conceptualised by extracting hybrid features from a very successful community life style found in the animal kingdom – the Ants. Ant colonies have evolved with means of performing tasks collectively, which are far beyond the capacities of their constituent components. The model conceptualises and implements a mobile colony of agents, that actively and intelligently interact with distributed networked services deployed on existing middleware architectures, such as CORBA and JMS. Therefore, this paper discusses the details of a distributed and mobile agent colony architecture that uses industry standard middleware to deploy environmental services.

Introduction

Our work conceptualises a multi-agent system, which is called a Colony, as agents work in harmony and synergy to achieve community wide goals. At present, an introductory level definition can be given to an agent as an entity with perceptions, goals, cognition, actions, and domain knowledge, situated in an environment [STON98]. AAANTS model uses the community life style of insects as a metaphor with further inspiration from "The Society of Mind" theory [MINS86]. The colony of agents is conceptualised by extracting features of very successful community life style of the ants.

Ants together with many other insect species, occupy a central place in artificial life due to their individual simplicity combined with relatively complex group behaviour [PARU97]. The evolution of the Ant has provided it with the capacity of functioning collectively, which enables it to perform tasks far beyond its constituent components. They do so without being hard-wired together in any specific architectural pattern, without central control. The consensus is that comprehension of emergent complexity in insect colonies such as Ants will serve as a good foundation for the study of emergent [BABA01] [GARC01], collective behaviour in more advanced social organisms, as well as leading to new practical methods in distributed computation.

AAANTS model is mainly defined by a distributed colony of agent components and a collection of services. A service can be any type of a component bundle consisting of hardware and software that is networked with a defined access interface using industry standard middleware, such as Java Messaging Service (JMS). The agents in AAANTS system are given sensory and actuator capabilities by these distributed heterogeneous services. The distributed agent colony too is defined as a collection of components. We understand components as an independently deliverable package of software operations that can be used to build applications or larger components [KNAP98]. Therefore, the colony can be envisaged as a distributed environment where myriad of components work in synergy while being mobile to specific colony locations during their life cycle.

In the AAANTS model we have used three industry standard middleware namely CORBA, JMS and JINI. CORBA is used for managing agent colony components by offering services for mobility, persistence, knowledge storage and name services. JINI is used for managing distributed collection of heterogeneous services that can range from high-level networked applications to embedded utility programs. JMS is used to merge the communication gap between agent components and the JINI based services.

The AAANTS model

Artificial software agents can be designed from different paradigms such as collaborative, reactive, hybrid, mobile and smart agents [NWAN96]. The proposed AAANTS model can be classified as a hybrid of cognitive and reactive models, which are currently popular within the agent research framework. The uniqueness of the AAANTS model can be summarised by considering the following design objectives.

- Artificial colonies of agents produce intelligent behaviour by using sensory and actuator services found in the environment.
- Definition of the environment with self-describing, distributed, heterogeneous services responsible for providing sensory and actuator facilities. In addition, these services are deployed on industry standard middleware such as JINI.
- A Shared information bus that facilitates communication within the agent colony, which is analogous to the pheromone trails found in a natural Ant colony.
- Competition and coordination among agents with ontological similarities to produce appropriate behaviour for a given sensation.
- Behavioural adaptations within homogeneous agent groups using past experiences.
- Periodic evolution within the agent colony through natural selection and totipotency.
- Mobility of agent components among several distributed locations of the same colony.

Therefore, AAANTS colony can be described as a synthetic ecosystem that collectively achieves goals on behalf of the human users by utilising the networked services. These services as mentioned earlier are heterogeneous, distributed, network transparent and self-describing in nature. It is these services that provide the agent community with sensory and actuator capabilities of their environment.

In a natural Ant colony, there are homogeneous groups of Ants with common features. Similarly, agents in an AAANTS colony can be segmented into groups that are homogeneous in nature. The grouping of agents is done on a functional basis such as financial market operations, personal assistance, security, information brokering, messaging, planning, data mining, etc. Agents in the same group share a common ontology that enables them to understand each other.

As we already know a single implementation of a AAANTS colony can be distributed on several networked locations. Also within a colony, there are groups of homogeneous agent component collections, which can be distributed on several locations. Though agents are distributed, interactions among them are possible with the use of messaging middleware. But still there are situations where agent components within a single homogeneous group need to be at one location in order to depict special behaviour. Consequently, we have introduced mobility characteristics to the model to facilitate agent component convergence.

AAANTS Architecture

Agent Architectures are a popular method adopted in the industry to conceptualise agent systems. They offer the traditional advantages of modularisation in software engineering and enables complex artefacts to be designed out of simpler components. The role of the architecture is to define a separation of concerns that identify the main functions, which ultimately give rise to the agent's behaviour and define interdependencies among them [LUCK98]. Therefore, architectures describe the high-level configuration of a system's constituent components and the connections that coordinate the activities among those components [KNAP98].

There is already quite a promising collection of architectures conceptualised and implemented by the academic and commercial institutions based on BDI (Belief-Desire-Intension), Blackboard, Subsumption, etc [FERB99]. We have the option of either to reuse an existing architecture or to design a new model altogether. But while considering the design objectives of the AAANTS model, it became quite apparent to us that existing architectures could not fully support the requirements of the AAANTS model.

Consequently, we have formulated a new architecture specially designed to achieve design objectives of the AAANTS model.

A conceptual view of the overall architecture of the AAANTS model is depicted in *figure 1*. The architecture has adopted a layered approach that has segmented the overall architecture into three functional concerns: the environment, adaptation, and agent colony. Each of the layers is composed of an egalitarian collection of components that interface in a dynamic manner.

The communication needs of an agent colony and distributed services are dissimilar. An obvious hurdle is to overcome interfacing among adjacent layers of the architecture while maintaining simplicity and dynamism. As a solution to this we have resort to the use of communication middleware that support heterogeneous distributed components to interface with one another in simple and a dynamic fashion.

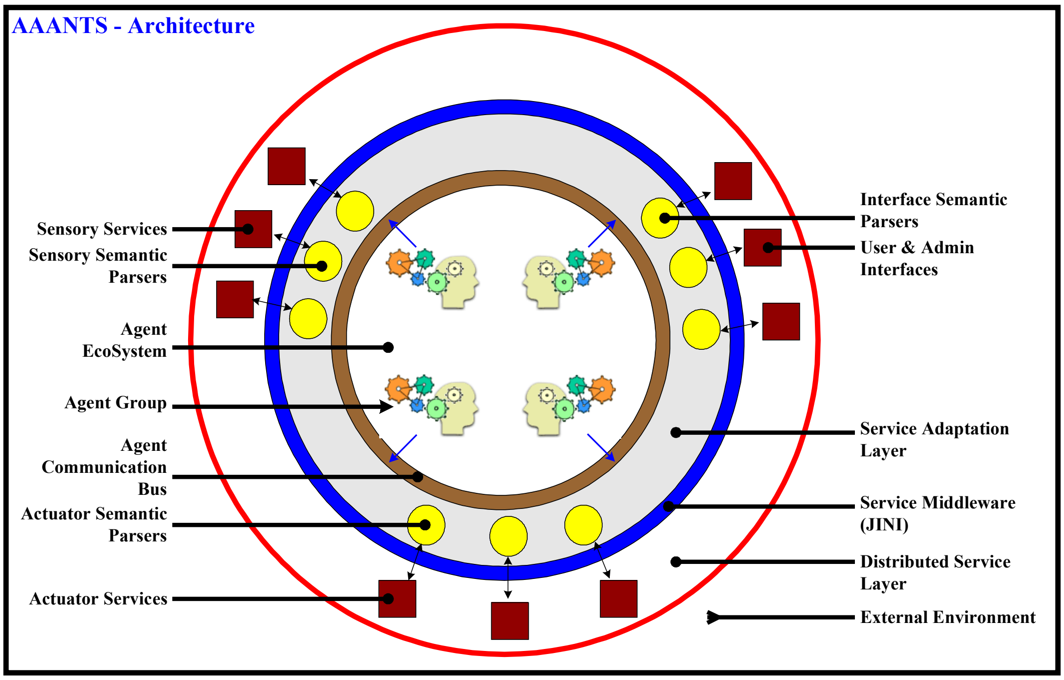


Figure 1: Layered architecture of AAANTS model [RANA01]

Implementation

AAANTS model is justified by using an implementation that consist of three modules: System Definition Components (SDC), System Execution and Control Components (SECC), and System Monitoring and Visualisation Components (SMVC). Figure 2 depicts these abstract level modules and a summary of interactions.

SDC is a broad term representing a collection of application tools that can be used to define an initial agent colony. The SDC components are used initially by an administrator to define and describe the characteristics of a specific AAANTS implementation. The definition involves describing about the layout, distribution model, ontologies, and initial knowledgebase. This definition is also persisted in a "Repository" implemented as a CORBA service.

SECC represents the run-time environment of the implementation consisting of agents, communication channels, and services. SECC is a highly dynamic and active environment that is analogous to an active ant colony. An important component of the run-time environment is the use of a distributed container to support total life cycles of agent components. Containers connect to each other and facilitator services with the use of CORBA middleware. The containers use the CORBA channel to exchange serialised mobile agent components. There after the services are implemented with the use of JINI middleware that offer benefits such as self-describing attributes, embedded support, attribute based name services, and event-based discovery and notifications. The agents interact with these services with the use of a gateway service that map JINI and communication bus middleware. The communication bus middleware is implemented using Java Messaging Service (JMS) that support subject based asynchronous messaging. The gateway performs an important function since agents and distributed services use two different types of communication middleware.

The next important part of SECC is the agent-based components. These agents are nurtured by the containers that provide life-cycle based functions, thread of execution and mobility. Agents interact with each other in the same colony with the use of subject-based messages that use XML for content definition.

SMVC is a set of tools used by the administrators and the end users to interact with an active agent system. SMVC is used for monitoring activities with a single colony implementation. The tools provide the visualisations of interactions among agents and services. It too provides a simulation environment that can be used to determine behaviour of the agents on different data sets.

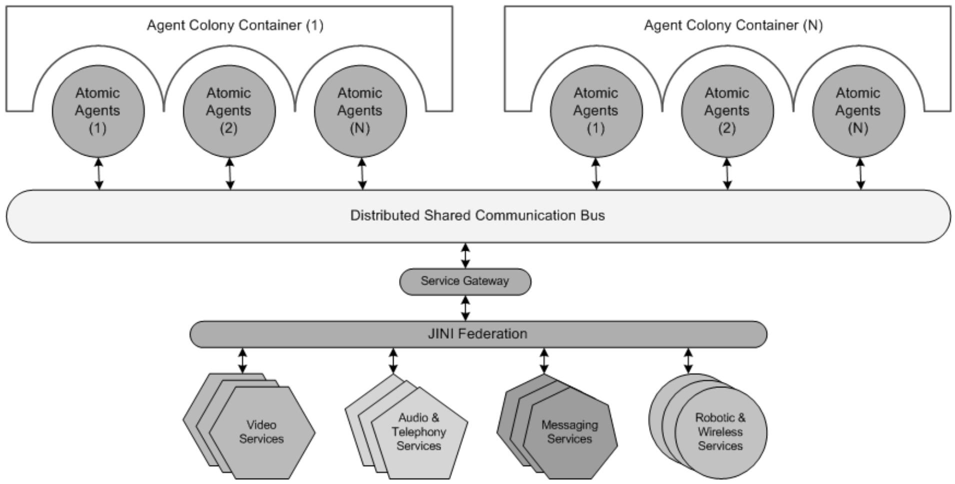


Figure 2: High-level components and their interactions in the AAANTS implementation

Conclusions and Implications

In this paper we have presented the architectural design of the AAANTS model that is analogous to a natural Ant colony. AAANTS is a general-purpose agent model since it has the capability to interact with heterogeneous services and has shown remarkable improvements over other functional monolithic agent models in terms of adaptability and knowledge/component reusability.

The implementation has proven that the use of the Adaptation Layer for interfacing has helped to overcome the messaging conflicts among agents and services. It was apparent from the implementation that this layer excludes the need for brokering and matchmaking services present in traditional deliberative architectures, since it enables subject based self-describing messages.

AAANTS model has helped us to observe emergent behaviour similar to that of a natural Ant colony. These agents sense the environment and communicate with others using primitive message constructs to offer emergent adaptive behaviour as a community.

References

[BABA01] Ozalp Babaoglu, Hein Meling, Alberto Montresor, 2001,Anthill: A Framework for the development agent-based peer-to-peer systems, University of Bologna, Italy, Norwegian University of Science and Technology, Norway.

[FERB99] Jacques Ferber, 1999, Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.

[GARC01] *Carlos Gershenson García, 2001,* Artificial Societies of Intelligent agents.

[KNAP98] Michael Knapik, Jay Jhonson, 1998, Developing Intelligent Agents for Distributed Systems, McGraw-Hill, ISBN 0-07-035011-6.

[LUCK98] M.Luck, M. d'Inverno, M.Fisher, 1997, Foundations of Muit-Agent Systems: Techniques, Tools and Theory.

[MINS86] M.Minsky, 1986, The society of mind, Simon and Schuster, New York, New York. ISBN 0-671-65713-5.

[NWAN96] Hyacinth Nwana & Divine Ndumu, 1996, An Introduction to Agent Technology, Intelligent Systems Research Applied Research and Technology, BT Labs.

[PARU97] Van Parunak, John Sauter, and Steve Clark, 1997, Toward the Specification and Design of Industrial Synthetic Ecosystems, Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).

[RANA01] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2001, Distributed Network Service Utilisation with Adaptive Ecology of Mobile Agents, INET 2001 the

11th Annual Internet Society Conference – Poster Session.

[STON98] Peter Stone, 1998, Layered Learning in Multi-Agent Systems, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Paper 6: AAANTS – An Intelligent Synthetic Ecosystem

R.A. Chaminda Ranasinghe
Department of Computer Science
University of Colombo, P.O.Box 1490
Colombo, Sri Lanka
+94-77-347220
chaminda@interblocks.com

Keywords: Multi-agent systems, distributed agent architectures, ubiquitous services, ant colony based agent platforms, synthetic ecosystems.

Abstract

There are many examples of successful emergent intelligence found in natural ecosystems. Among them, the community behaviour of the insects and specially the ants has been admired by humans for centuries. Ants occupy an important position in community life due to individual simplicity combined with relatively complex group behaviour. Ant colonies have evolved means of performing tasks collectively, which are far beyond the capacities of their constituent components. We have designed and implemented an agent model called AAANTS (Adaptive Autonomous Agent colony interactions with Network Transparent Services), that consist of a collection of software components which can be considered as an artificial colony or society. AAANTS conceptualises and implements a colony of agents that actively and intelligently interact with distributed networked services that act as an augmented neural extension to the assisted users.

In this paper we discuss the conceptualisation of the AAANTS model with reference to community life styles of an ant colony. Further we focus on the architecture, task distribution and adaptability of a myriad of components with an active environment of distributed heterogeneous services.

Introduction

AAANTS model consists of a collection of software components which can be considered as an artificial colony or society, since they work in harmony and synergy to achieve community wide goals. The model can also be classified as a complex system [GARC01] [BABA01] since there is large number of components with complex interactions among them.

The colony of agents is conceptualised by extracting features of a very successful community life style found in the animal kingdom – the Ants. In this context a colony can be envisaged as a distributed environment where a myriad of components work in synergy while being mobile to specific colony locations during their life cycle [RANA02]. Synthetic ecosystems such as insect species, occupy a central place in artificial life due to their individual simplicity combined with their relatively complex group behaviour [PARU97]. Ant colonies have evolved means of performing collective tasks, which are far beyond the capacities of their constituent components. They do so without being hard-wired together in any specific architectural pattern, without central control. The consensus is that comprehension of emergent complexity in insect colonies such as ants will serve as a good foundation for the study of emergent [BABA01], collective behaviour in more advanced social organisms, as well as leading to new practical methods in distributed computation.

This paper discusses the conceptualisation and implementation of a model that facilitate the proper functioning of various components found in a synthetic ecosystem such as AAANTS. One of the key issues addressed is the distribution and management of myriad of components that constitute an active AAANTS implementation. Thereafter, a methodology is presented that enable the dynamic interaction among components with the help of a messaging platform.

Background and Motivation

Several metaphors and paradigms have given inspiration to the proposed AAANTS model. An important metaphor is the goal-driven community life style of ants in the animal kingdom. Ants' success story spans several millions of years even before the first humans are into being. Since each individual ant maintains the capability to solve an integrated part of the overall puzzle, the key motivation was to device a model that can depict this intelligence in a similar fashion. Further inspiration is derived by "Society of Mind" theory by Marvin Minsky [MINS86], which portrays the mind as a collection of mindless components that interact and compete to provide intelligent behaviour to environment perceptions.

The concept of synthetic colony based ecosystems is present in the work done on several other projects such as Hive [MINA99], Amalthaea [ALEX96] and Anthill [BABA01]. Among them Hive is described as an ecology of distributed agents that harness the facilities of local resources by the use of an application created out of the interaction of multiple agents across a network. The heterogeneous distributed services discussed in the AAANTS theory have gained wisdom from "Ubiquitous Computing" paradigm, which was predicted by Mark Weiser [WEIS93] where naturally embedded process capable components assist humans in their daily activities.

AAANTS Model

The proposed AAANTS model can be classified as a Complex System. A complex system is usually constituted of many elements, which interact with each other. The complexity of the system is proportional to the number of elements, the number of interactions in the system, and the complexities of the elements and of their interactions [BABA01]. In natural complex systems, every element is also a complex system; therefore we can only obtain a relative complexity depending on a reference point. Since we can use various reference points, there cannot be an absolute complexity, and each relative complexity will be different. The global behaviour of the system arises from the interactions of the elements of the system. Therefore a complex system is more than the sum of its constituents. This means that a complex system has properties that are not present in its constituents. These properties are called *emergent* [GARC01]. They emerge from the interactions of the components with in a complex system.

Currently, artificial software agents can be designed from different paradigms such as collaborative, reactive, hybrid, mobile and smart agents [NWAN96]. The proposed **AAANTS model** takes wisdom from naturally occurring Ant colonies, and is a hybrid of cognitive and reactive agent domains. The uniqueness of the AAANTS model can be summarised by considering the following characteristics.

- Presence of artificial societies/colonies of agents that produce intelligent behaviour by using sensory and actuator services embedded in the environment.
- Definition of the environment with self-describing, distributed and heterogeneous services.
- Evolution of the agent colony by natural selection of the fittest.
- Competition among a group of agents with ontological similarities to produce the best-suited behaviour for a given sensation.
- Group adaptation of behaviour by using credibility of past actions of the colony.
- A shared communication bus to facilitate interaction among the agents and services.

With reference to the above characteristics, AAANTS can be described as an artificial agent society that collectively achieves goals of the human users with the use of networked services. These services are heterogeneous, distributed, network transparent and self-describing in nature. The services being heterogeneous do not only mean running on heterogeneous platforms but also possessing heterogeneous functionality [RANA99]. Further, these services provide the agent community with sensory and actuator capabilities of their environment.

In an ant colony there are different groups of ants that have common features. For example worker ants, soldier ants and housekeeper ants though similar in structure, possess specialities to do their tasks better. Similarly, in the AAANTS model, agents can be segmented into groups that perform similar tasks. Each group possesses the capability to perform specialised functionality. The grouping of agents is done on a functional basis such as financial market operations, personal assistance, security, information brokering, messaging, planning, data mining, etc. Agents in the same group share a common ontology that enables them to understand each other. However, since each group uses different ontologies we have to device a mechanism to bridge each of their capabilities to achieve community wide goals.

AAANTS internal representation and the environment

The agents in the AAANTS system are supposed to work in a community that intercept environment sensations and convert them to actions that benefit the society as a whole. But there may be a need to generate environment sensations within the agent community so as to simulate periodic activities within the community. This is similar to the idea of sensing events inside the brain as described by [MINS86], where only a small minority of agents connected directly to the sensors of the outer world, like those that send signals from the eye or skin; otherwise most of the agents in the brain detect the events inside the brain.

The actuator agents are responsible for changing state of its immediate environment with the influence of actions. Actions and their consequences can be analysed as transformation of a global state, responses to influences, computing process, local modification, physical displacement, and as commands [FERB99]. Therefore, an action can be accepted as a modification of the environment. In the AAANTS model, the individual actions of the agents can be linearly applied in the environment or amalgamated to a composite action that may do more complex modification.

The sensory and actuator related information flow of the AAANTS model is depicted in *figure 1*. The model uses an information middleware that facilitate subject-based broadcasting, message queuing, and routing. This middleware is similar in functionality to the chemical trails of Pheromones used by natural ant colonies to exchange information and coordinate collective activities. Agent components in an active AAANTS colony would listen to interested subjects for asynchronous delivery of sensory messages published by sensory services through the Service Gateway. Similarly, actuator related information too is published by the agent colony components to be intercepted by the respective environment services.

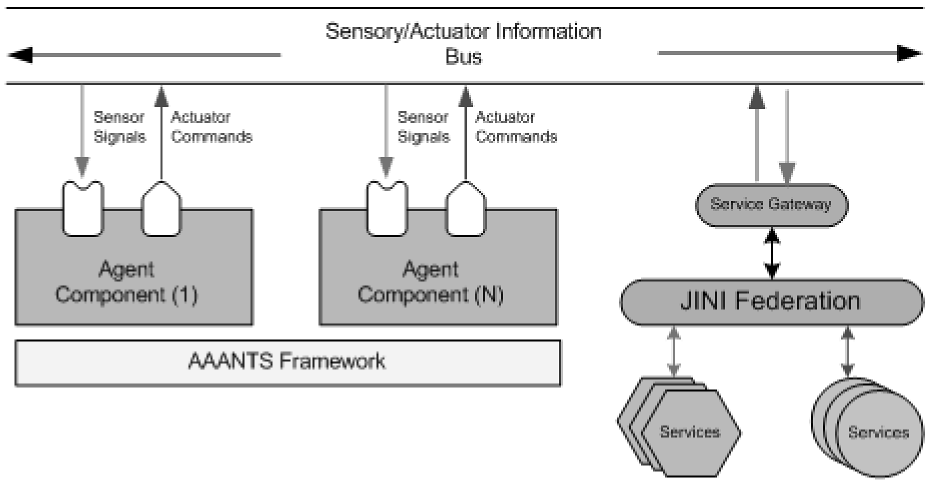


Figure 1: AAANTS interactions with the Environment [RANA01]

It is through perception that the agent acquires information about the world to allow it to prepare its action to pursue its goals. Therefore agents require an embedded perception system in order to perceive the environment. Perceptive systems can be passive or active depending on the approach taken to perceive the environment [FERB99]. In passive perceptive systems the signals follow an approach that is entirely constructional where elementary signals are pre-processed and then segmented to obtain elementary features leading to the recognition of objects, scenes, words or phrases. In active systems, by contrast, the perceptive system simultaneously receives the data coming from the sensors and the expectations and goals coming from the cognitive system. It can control it sensors in such a way as to maintain a coherent representation of its environment. Therefore, the AAANTS model should adopt an active perceptive system to comply with that of natural ants. But, it should also be noticed that there exists some amount of pre-processing of sensory information by the service specific parsers before being published to the agent components.

Reproductive and evolutionary aspects of the AAANTS

AAANTS system is composed of groups of agent colonies that may have intra or inter-group interactions. However, among all types of interactions, intra-group interactions are more common and frequent since the agents with in a group try to achieve a shared set of goals. Since all the agents are adaptive, we can expect an improvement and efficiency in the intelligent activities demonstrated by all the agents in the system. However, since each agent learns independently from the environment, different agents may depict heterogeneous behaviour to the same kind of environment sensations. Some of the behaviour of individual agents may get obsolete as the total system proceeds in time. Therefore, we have proposed an evolutionary and reproductive mechanism similar to that found in the biological environment to improve the well-being and suitability of an agent colony over a period of time.

In the AAANTS system, each agent maintains a variable that represents its fitness in the total agent colony. Fitness is a representation of the correct behaviour in view of the end user. Agents increment or decrement their fitness variable value depending on the feedback from the responsible end-users. The agents of the same group in the colony are rewarded in the same fashion for their actions. Therefore, within the same agent group the fitness variable decides to some extent the successful actions taken by a collection of agents. Periodically the agent framework eliminates the unfit agents from the agent colony. An agent being unfit is decided by using a threshold value of the fitness variable, which again would be a variable depending on environment factors.

However, there can be situations where a collection of actions is selected. Here we come across the problem of an execution plan. We use joint and concurrent action selection methods as mentioned in [GRIF99]. A joint action is a composite action, made up of individual actions that must be performed together by a group of agents. Each agent involved in executing a composite action makes a simultaneous contribution to the overall result. Concurrent actions are those that can be performed in parallel by different agents, without the need for synchronisation.

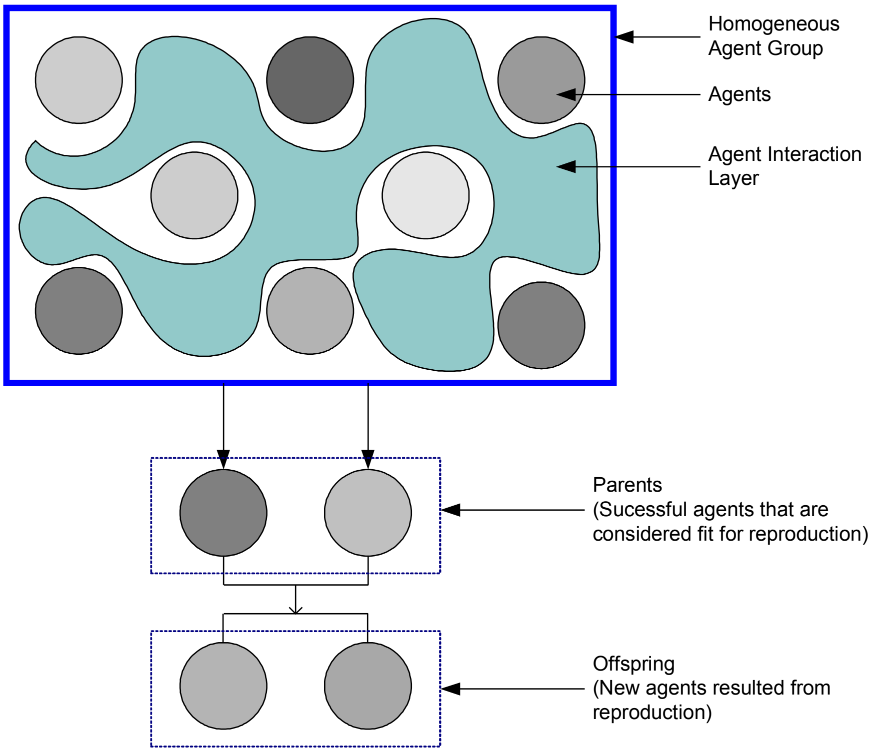


Figure 2: Evolutionary capabilities of the AAANTS model

As described in the above paragraphs, selection of the fittest agent is performed through a properly structured mechanism. During this process, unfit agents are eliminated from the agent ecology. These reduced quantities of agents should be reinstated to maintain a healthy ecology. Therefore, as found in the natural biological world, we have introduced a reproduction mechanism to introduce new member agents to the AAANTS system. The reproduction mechanism can be summarised as depicted in *figure 2*.

Agents in the AAANTS system are organised into groups. Since the agents in one group are responsible for a subset of shared activities, we can call it as a homogeneous agent group. Normally reproduction mechanism takes place within agents in a homogeneous group and among them, only the fittest are allowed. Each agent consists of reusable components that may or may not take part in reproduction. Therefore, as depicted in *figure 2*, the child agents produced may have a blend of features from their parents.

One of the main objectives behind the AAANTS system is to make it adaptive to the changing environment. There is no better example to take for this issue other than from the natural biological world – reproduction and evolution. The reproductive and evolutionary mechanisms introduced in AAANTS model can be considered as a stepping-stone for achieving adaptability found in the biological world.

Implementation

We have implemented an Agent Building Platform (ABP) for the AAANTS model. The platform is implemented on a distributed architecture using Common Object Request Broker Architecture (CORBA using VisiBroker). The platform is capable of building an agent colony to automate a defined collection of services by dynamic configuration. Interface Description Generator (IDG) of the developed software tool can describe the interfacing of agent-to-agent and agent-to-service communication.

AAANTS Building Platform (ABP) consists of distributed processes such as Domain Controller, Agent Component Manager, Controller Interface, Knowledge Repository and Service gateway. The agent components that belong to a single colony can be distributed on multiple component managers with the facility to migrate to different platforms. The total knowledge of the colony is kept at the Knowledge Repository that can be modified by the Knowledge Edit Tool (KET) implemented in the Controller Interface. Using KET, the administrators can introduce and modify knowledge structures.

The user interaction with AAANTS implementation uses heterogeneous interfaces such as stand-alone, Internet, and PDA clients. These interfaces allow the agent colony to actively communicate with user for notification, confirmation, and configuration. Users may also be responsible for maintaining some of the services such as Global Positioning Systems (GPS), Mobile phones etc. in a ubiquitous manner, that are used by agent ecologies for decision making and assistance.

ABP also offers a tool that can be used by the administrators to activate reproductive cycles among primitive agents. This tool would first perform a fitness test to select suitable participants for reproduction. The selected participants are randomly combined to generate genetically improved offspring. The improvement can be described in lines of knowledge, composition, ontology bridging and communicability. The tool also offers set of interfaces for end users to artificially simulate the reproduction procedure. So overtime, AAANTS produce agent-based components that are better equipped to survive in the environment consisting distributed services.

The initial prototype system of AAANTS has been tested *(Figure 3)* with a robot built from a simple toolkit with a small digital camera acting as the sensory device of the robot. The robot was programmed to use the camera as its eye and take appropriate action depending on the output of the camera. Further, we used other simulated services such as infrared sensors, GPS, SMS, email, voice synthesis and recognition to simulate heterogeneity.

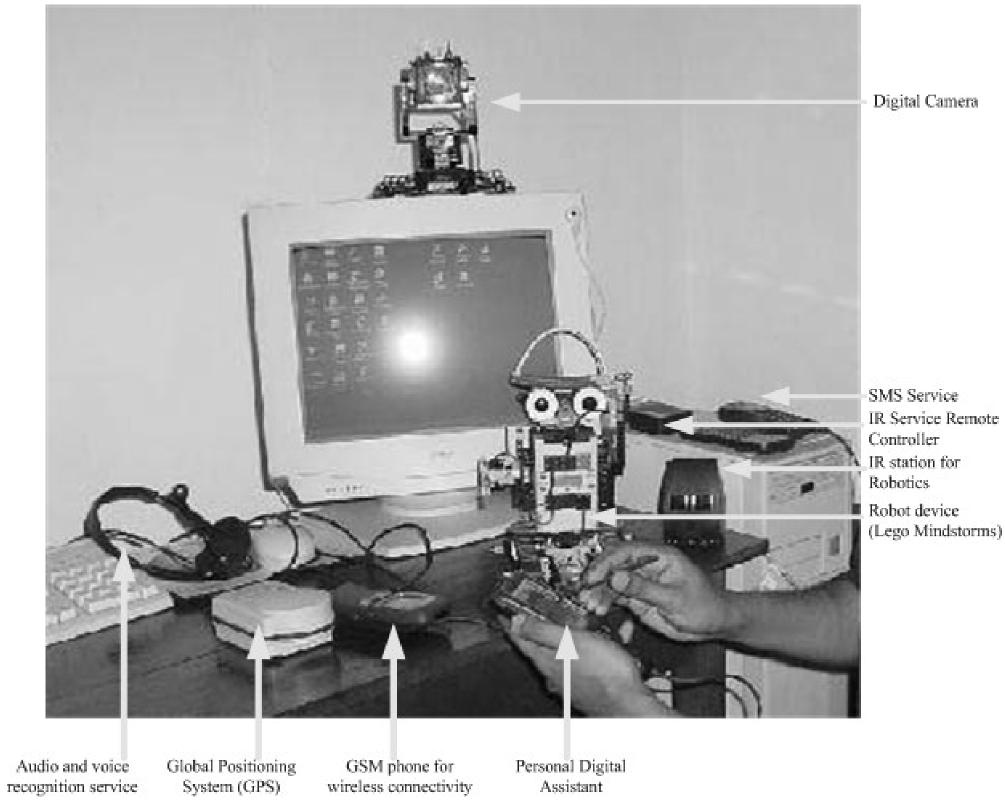


Figure 3: AAANTS simulation and testing environment

Conclusions and Implications

In this paper we have presented a design model of a system that attempts to model an agent based software system using the characteristics of an ant colony. AAANTS system is a general-purpose agent system since it has the capability to interact with heterogeneous services where it has shown remarkable improvements over other functional monolithic agents in terms of adaptability and component reusability.

We were able to simulate emergent behaviour in a synthetic ecosystem using the AAANTS model. The agents behaved in a collective manner for sensations in the environment. We tested the model with simulated and actual sensation data to ascertain predictability. The implementation depicted adaptability with proper timing and sequencing of atomic behaviour over long periods of time. We have noticed that while introducing a new service (sensation) to the environment, that a colony shows exponential adaptation when initially trained with an actual data set.

We expect to perform future improvements on the evolutionary aspects of a colony so that an implementation can exist with long lapse of administrator intervention. We hope that this would enable us to create a synthetic replica of a natural Ant colony.

References

[ALEX96] Alexandros Moukas, 1996, Amalthaea: Information Discovery and Filtering using a Multi-agent Evolving Eco System, MIT Media Laboratory.

[BABA01] Ozalp Babaoglu, Hein Meling, Alberto Montresor, 2001, Anthill: A Framework for the development agent-based peer-to-peer systems, University of Bologna, Italy, orwegian University of Science and Technology, Norway.

[FERB99] Jacques Ferber, 1999, Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, Addison-Wesley, ISBN 0-201-36048-9.

[GARC01] *Carlos Gershenson García, 2001,* Artificial Societies of Intelligent agents**.**

[MINA99] Nelson Minar, Mathew Gray, Oliver Roup, Raffi Krikorian and Pattie Maes, 1999, Hive: Distributed agents for Networking Things, MIT Media Lab.

[MINS86] M.Minsky, 1986, The society of mind, Simon and Schuster, New York, New York. ISBN 0-671-65713-5.

[NWAN96] Hyacinth Nwana & Divine Ndumu, 1996, An Introduction to Agent Technology, Intelligent Systems Research Applied Research and Technology, BT Labs.

[PARU97] Van Parunak, John Sauter, and Steve Clark, 1997, Toward the Specification and Design of Industrial Synthetic Ecosystems, Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97).

[RANA01] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2001, Distributed Network Service Utilisation with Adaptive Ecology of Mobile Agents, INET 2001 the 11th Annual Internet Society Conference – Poster Session.

[RANA02] R.A. Chaminda Ranasinghe, A.P Madurapperuma, 2002, AAANTS – Distributed Mobile Component Architecture for an Ant Colony based Synthetic Ecosystem, Accepted to be published at (MATA'02) Fourth International Workshop on Mobile Agents for Telecommunication Applications, 23rd - 24th October 2002 Universitat Pompeu Fabra. Barcelona, Spain.

[RANA99] R.A. Chaminda Ranasinghe, A.P Madurapperuma, October 1999, Distributed Service Utilisation on Heterogeneous Networks using Mobile Agents,bInternational Information Technology Conference, Colombo, Sri Lanka.

[WEIS93] Mark Weiser, 1993, Some Computer Science Problems in Ubiquitous Computing, Communications of the ACM, Hot Topics: Ubiquitous Computing, IEEE Computer.