

THE UNIVERSITY OF HULL

Investigation of motivation in agents using  
the simulation of 5-side football

being a Thesis submitted for the Degree of  
Master of Science  
in the University of Hull

by

Humberto António Nunes, BSc. (Hons.)

November 2001

# Abstract

Artificial Intelligence is a challenging field to research into due to the nature and complexity of its agenda. For such reason, the progress in the Artificial Intelligence field has been slow but steady. With the advancement in technology, predominately the processing power of computer hardware, Artificial Intelligence has been progressing in a faster pace in the last 10 years. One of the concepts in AI involves simulation of agents in a software perspective.

This project investigates motivation of multiple synthetic agents through software simulation. This research involves analysis and investigation of the relationship between motivation, priority and urgencies, and the use of a specific domain.

5-aside football is used as the environment for the synthetic agents to interact in.

Other concepts related to agent architecture are investigated in this research. Communication between agents is also researched into, and coaching from an external agent is used to investigate the potential of such usage in 5-aside football scenario.

Finally, agents having capabilities such as motivation, priority, urgencies, communication and coach are evaluated in terms of performance and significance of these concepts within the agent architecture. Evaluation is judged on the score of each football game, having a team of agents as footballers competing with another team with different combination of agent functionality.

This research aims to find an agent type with the combination of integrated properties and architecture which has the best performance with high speed of execution for this research application.

## Acknowledgements

I would like to thank Dr. Darryl N. Davis for providing the research and programming material, assistance in report writing, feedback for the research work and the entire supervision of the project. Thanks to Dr. Chandrasekhar Kambhampati for coordinating the entire project and Mr. Derek P. Wills assistance for the research course. I would also like to thank Dr. Sabarinath Rajasekharan for guidance in report writing, Mr. Granville J. Hawkins for assistance in report writing and Miss Suzanne Lewis for the assistance in topics related to background research. Additionally, I would like to thank my parents for the financial and moral support, and all the Postgraduate students of year 2000 in the Department of Computer Science in University of Hull for their support.

# Table of Contents

1	Introduction.....	1
2	Artificial Intelligence, Agents and Football Simulation.....	3
2.1	Introduction.....	3
2.2	Classical AI and Nouvelle AI.....	3
2.3	Cognitive Science.....	4
2.4	Computational Agent.....	4
2.4.1	Theory of Agent.....	4
2.4.2	Weak and Strong agents.....	5
2.4.3	Concepts related to agent architecture.....	6
	Blackboard control architecture.....	6
	Belief-Desire-Intention Model.....	7
	Will's architecture.....	8
	Abstract three level layer architecture.....	9
2.4.4	Summary of the agent's architecture.....	11
2.5	Motivation and Goal.....	12
2.6	Research of football simulation.....	14
2.7	Research Questions.....	16
3	Design and analysis of the agent for the 5-side football simulation.....	17
3.1	Requirements for a motivational agent.....	17
3.2	Representation of the agent architecture.....	18
3.3	Features and capabilities for an agent in the 5-side football simulation.....	19
3.4	Roles of the agents in the CAM Architecture.....	22
3.5	Conditions for agent's task execution.....	26
	For the footballer.....	28
	For the ball.....	29
3.6	Details on agent's components and means of execution.....	30
3.6.1	Rest.....	31
3.6.2	Move around.....	32
3.6.3	Move to ball.....	32
3.6.4	Block an opponent.....	32
3.6.5	Get ball.....	32
3.6.6	Catching ball.....	32
3.6.7	Shooting to goal.....	32
3.6.8	Passing the ball.....	33
3.6.9	Dribbling.....	33
3.6.10	Motivators.....	33
	Block opponent.....	34
	Dribbling.....	35
	Passing the ball.....	36
	Without motivators.....	38
3.6.11	Priority.....	38
3.6.12	Urgency.....	39
3.6.13	Communication.....	40
	Mark the opponent to block.....	41
	Pass the ball to the team-mate.....	41
	Getting the ball.....	41
3.6.14	Coach.....	42
3.7	Design of the simulation software of 5-side football.....	43

4	Implementation and testing for the simulation of 5-aside football.....	48
4.1	Introduction.....	48
4.2	Implementation of the 5-aside football simulation.....	48
4.3	Interface of the 5-aside football simulation program.....	49
4.4	Testing the program of 5-aside football.....	51
4.5	Enhancements made during tests in the simulation program.....	52
4.6	Testing different combination of agents' functionality.....	53
4.7	Problems encountered in the tests of the simulation program.....	55
5	Evaluation of the results from the simulation of 5-aside football games.....	58
5.1	Introduction.....	58
5.2	Results gathered from the 5-aside football games.....	58
5.3	Evaluation of the results from data gathered in the simulation programs.....	60
6	Critical appraisal of the research in motivation of agents in 5-aside football simulation.....	61
6.1	Introduction.....	61
6.2	Course Timetable.....	61
6.3	Critical appraisal of this project.....	61
7	Conclusion.....	63
8	Scope for Further Work.....	66
	Bibliography.....	68
	Appendix A.....	73
	Appendix B.....	78
	Appendix C.....	79

# List of Figures

Figure 1. Overlapping concepts in AI and Cognitive Science.....	4
Figure 2. The Blackboard Control Architecture .....	6
Figure 3. Extended version of Blackboard Control Architecture for multiple agents’ scenario .....	7
Figure 4. BDI Agent Architecture .....	8
Figure 5. Will’s full architecture.....	9
Figure 6. Abstract Three Level Architecture for a single Agent with Explicit Motivational Structures.....	10
Figure 7. Abstract Three Level Architecture for Multiple Agents with Explicit Motivational Structures.....	11
Figure 8. Control states .....	13
Figure 9. Abstract Three Level Architecture for a single Agent with Explicit Motivational Structures.....	18
Figure 10. Abstract Three Level Architecture for Multiple Agents with Explicit Motivational Structures.....	19
Figure 11. State Transition Diagram for the footballer of the system .....	23
Figure 12. CAM Architecture for an outfielder .....	24
Figure 13. CAM Architecture for a goalkeeper .....	25
Figure 14. Footballer aiming for the goal .....	33
Figure 15. Examples of a footballer between two opponents.....	34
Figure 16. Example of getting the ideal position <i>id</i> between two opponents .....	34
Figure 17. Calculating distances between agents .....	35
Figure 18. Length <i>a</i> ’ between object A and C .....	35
Figure 19. Footballer dribbling away from opponent.....	36
Figure 20. Footballer swirling the ball .....	36
Figure 21. Footballer between team-mate and opponent.....	37
Figure 22. Footballer not between team-mate and opponent.....	37
Figure 23. Different areas of the football field .....	40
Figure 24. Communication for agents using blackboard architecture .....	41
Figure 25. Advises from the coach .....	43
Figure 26. Football field for the 5-aside football.....	44
Figure 27. Scoreboard of the 5-aside football game.....	44
Figure 28. Footballer’s status.....	45
Figure 29. Ball bouncing from the edge of the football field .....	46
Figure 30. Ball bouncing off from the footballer.....	47
Figure 31. 5-aside football game window .....	50
Figure 32. Score board in the simulation game .....	50
Figure 33. Agent board in the simulation game.....	51
Figure 34. Agents and the ball getting stuck in the same position .....	51

## List of Tables

Table 1. Components in different architectures for identical functionality.....	12
Table 2. Features and Capabilities in a CAM architecture .....	20
Table 3. Footballer's behaviour .....	28
Table 4. Ball's behaviour.....	28
Table 5. Urgency value for footballers in each football field area .....	40
Table 6. Energy threshold for the agent to rest depending to its urgency .....	40
Table 7. Results from different speeds in the football game (ball not bouncing off the footballer).....	53
Table 8. Results from different speeds in the football game (ball bouncing off the footballer).....	53
Table 9. Different types of agents against each other in the 5-aside football simulation .....	55
Table 10. Results from the matches in different types of agents in the football team.	59
Table 11. Tasks made during the research.....	61

## List of Pseudo Codes

Pseudo code 1. Rules of the agent's tasks .....	29
Pseudo code. 5-aside football simulation (from Appendix C) .....	83

# 1 Introduction

Artificial Intelligence (AI) is one of the youngest fields in computer science [McCarthy 2000]. It is also one of the most challenging fields due to the complexity of its nature. Many of its practitioners have different points of view regarding the nature of AI, for example Nilsson [1993], Brooks [1990], Maes [1990] and Franklin [1997a].

Artificial Intelligence has many aims, for example:

- to simulate intelligence for analytical studies and thereby improve the understanding of intelligence and behaviour [Davis 2000]
- to perform tasks in a sensible manner [Maes 1990]
- to enhance the performance in computing using synthetic agents [Sloman 1994; Bates 1991]
- to create machine intelligence by using concepts taken from Mathematics, Cognitive Science, Philosophy, Psychology, Biology and Sociology. [Holland 1994; Franklin 1997b]. For this reason, it has captured the attention of many researchers in these fields.

This has resulted in many useful concepts and systems, for example GermWatcher in medical field for checking patients' infections [Coiera 1997], CHEMREG for checking the health and safety of chemical products [Wilson 1998], and Quake 2 in gaming for entertainment purposes [Lent 1999].

One of the challenging areas in AI relates to Cognitive Science and the concepts of motivation and emotion. This project investigates motivation using 5-a-side football as a dynamic environment for the synthetic agents to interact in. The nature of 5-a-side football provides multiple agents scenario for each team to participate in the game. Murphy [2000] suggests that researches in applying AI to games or competitions attract both students and scientists. It is enjoyable to those who like designing or playing such games, and makes for a motivated career in robotics and AI.

Although 5-a-side football simulation is used as a research vehicle, this project is mainly concerned with the investigation of motivation among multiple agents in competitive and co-operative manner, depending on which team these agents belong to. Therefore, the football simulation used will be relatively simple. Through the investigation of agent architectures, an analysis and evaluation will be performed on their suitability for modelling motivation in multiple agent scenarios.

Chapter 2 illustrates the concepts regarding to the background research of this project, and research questions raised requiring investigation. In addition to motivational aspects of multiple agents, communication and co-ordination of distributed agents (using Abstract Manager) are also investigated.

Chapter 3 investigates the concepts for designing in the agent architecture, and the software and programming of simulation of 5-a-side football.

Chapter 4 illustrates the structure for implementing the 5-a-side football program. Tests made during the programming phase and problems encountered are annotated in this chapter.

Evaluation of results taken from the simulation software is discussed in chapter 5.

Critical appraisal of this research is discussed in chapter 6.

Finally, conclusions and scope for further work of this research are in chapter 7 and 8 respectively.

## 2 Artificial Intelligence, Agents and Football Simulation

### 2.1 Introduction

This chapter focuses on background research of the investigation of motivation agents in 5-aside football scenario.

Section 2 describes the concept of Classical AI and Nouvelle AI, then explains which methodology is appropriate for this research. Section 3 illustrates the relationship between Cognitive Science and Artificial Intelligence. Section 4 explains the meaning of ‘agent’ from an Artificial Intelligence perspective, and illustrates agent architectures developed by AI practitioners that are relevant to this project. Section 5 describes the concept of motivation and other related concepts. Details of the football simulation will be discussed in section 6. Section 7 indicates research questions to be investigated and aims to be achieved in this research project.

### 2.2 Classical AI and Nouvelle AI

There are many types of research activity associated with Artificial Intelligence. Two types relevant to research in Artificial Intelligence relate to the categories of representation and manipulation of data in an agent or a system, which are Classical AI and Nouvelle AI.

Classical AI [Franklin 1997b; Maes 1990] involves information processing by symbolic computation, that is, rule-based symbol manipulation. These data can be either represented using mathematical values or symbols. The system searches the rules to match a satisfied condition. Any condition that does not match any rules in the system is considered to be false. Therefore this method may lead to system failure when the programmer overlooks conditions or the system is unable to handle unexpected conditions that may arise [Franklin 1997a]. Despite these problems in Classical AI, many applications programmed in the past have proven to be capable of performing impressive tasks [Maes 1990].

Nouvelle AI [Maes 1990; Franklin 1997a] is a decomposition of intelligence into individual behaviour generating modules, whose coexistence and co-operation enable complex behaviours to emerge [Maes 1990]. Each module generates some behaviour. Optimisation is achieved by adding new modules to the system. This approach caters for situations that are not pre-defined in the system, or unexpected cases, because data read by the system are manipulated with the execution of corresponding modules, even those newly added. This approach is able to perform certain tasks that Classical AI could not. For example, in simulated football environments, it is not possible to define all the possibilities that a footballer may encounter. On the other hand, it is appropriate for the football agents to be able to generate their next action by computational calculation. When a module is added, that is using an Object Oriented Programming approach, all other modules will work in a co-operative manner and be able to optimise the system.

For further details of concepts in Classical AI or Nouvelle AI, please refer to [Maes 1990; Franklin 1997a; Franklin 1997b].

For this research, the combination of Classical AI and Nouvelle AI is employed in the simulation software. Classical AI provides the choices for the agent to decide which rules to apply to. Nouvelle AI provides the potential for the agent to perform a task or interpret a situation arising from the environment.

## 2.3 Cognitive Science

When considering the science of Artificial Intelligence, another field that AI practitioners would consider is Cognitive Science.

Cognitive Science is a study of intelligent behaviour in computational or living beings.

There is no separate ground between Artificial Intelligence and Cognitive Science, because concepts in both studies overlap each other. Both sciences include studies or implementations of intelligence, and the use of their results in improving or helping human lives. Figure 1 illustrates this phenomenon.

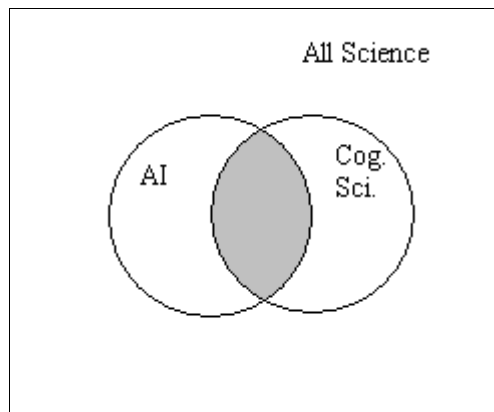


Figure 1. Overlapping concepts in AI and Cognitive Science

Computing can be used as cognitive aid for analysing theories [Sloman 2000a]. For such reason, Artificial Intelligence and Cognitive Science are linked.

Implementing a Synthetic Cognitive System (software or hardware) is much more complicated than just a system to perform tasks correctly. The Synthetic Cognitive system has to contain a detailed functional description of its counterpart Natural Cognitive System [Schetz 2000]. For example, when implementing a humanoid robot from a Cognitive Science perspective, it should contain the features of humans, such as emotional states, learning and communicational ability. As this example shows, implementing a Cognitive System is very difficult, if not impossible.

## 2.4 Computational Agent

### 2.4.1 Theory of Agent

In Artificial Intelligence and Cognitive Science, the agent is used as representation of concepts in these science fields.

An agent is described as a living or non-living entity, such as for instance an animal or a car. In Artificial Intelligence and Cognitive Science, the meaning of an agent is more specific. There are various different definitions by scientists that define what an agent is, but these definitions contain similarities that are intended to express the concept of an agent.

Maes [1995] defines agents as computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realise a set of goals or tasks for which they are designed.

In addition to Maes definition, Hayes-Roth [1993b] defines that an agent may perform action to affect conditions to the environment.

The definition from Maes and Hayes-Roth can be applied to both synthetic (robotic or computational) and biological agents. It means that an agent is capable to perform a task with its own desire, surviving in an environment, or even changing an environment for its suitability. Agents in Artificial Intelligence and Cognitive Science are also used for analytical purposes [Davis 1998].

It can be said that all agents have goals. For example, a goal for a biological agent can be to stay alive [Jenkins 1998]. If the agent does not want to stay alive, its goal might be a tendency to kill itself.

## 2.4.2 Weak and Strong agents

At the moment the technology in computing does not allow one to implement and test (in terms of monetary costs and response speed in complex systems) all the concepts and designs in the literature of Artificial Intelligence and Cognitive Science into a single system or parallel systems executing concurrently. There are still many concepts in these fields for the design and implementation that need to be investigated.

Agents in Artificial Intelligence and Cognitive Science are categorised into two types in respect to their capability and functionality. They are ‘weak agents’ and ‘strong agents’.

Weak agents possess autonomy, reactivity, pro-activity, social ability [Wooldridge 1995b] in their environment. These agents are intended to achieve their goals in the simplest way possible within a specified time, using an architecture built as simple as possible. This can be done for instance using simple mathematical equations to generate an output for an agent.

Strong agents include all the attributes in weak agents plus human capabilities such as knowledge, belief, intention and other motivational behaviours (emotions, desires, goals, and attitudes) [Davis 2001]. The intention of having strong agents is to mimic humans in performing a task and to achieve their goals. These agents are useful for scientists in analytical purposes for AI and Cognitive Science, or to simulate human behaviour.

Castelfranchi [Wooldridge 1995a] indicates that agent autonomy is crucial for AI studies. Castelfranchi has defined several different types of autonomy which an agent may possess, such as motivational autonomy, social autonomy, and goal autonomy. Furthermore, there are degrees of autonomy within the agent, as opposed to the concept of agent in full autonomy or no autonomy present in the agent. For such reason, agent’s autonomy is limited.

In addition to weak agents and strong agents, two approaches relating to functionality of the agent architecture are considered when designing and implementing a new concept for AI, to overcome the technological limitations mentioned. These are *broad and shallow*, and *deep and focussed*.

*Broad and shallow* corresponds to an architecture containing several different functions, but simple in design and may not be able to achieve high performance in terms of the quality of execution and efficiency [Sloman 1994]. This concept enables a programmer to introduce multiple capabilities and features for an agent performing simple tasks.

In contrast, *deep and focussed* architecture involves great details in design and high performance for a limited number of functions in an agent. This concept enables an

agent to have limited functionality with a great deal of technicality and outstanding performance, but within a very limited domain.

The approach used in this project in designing and implementing agents is *broad and shallow*, making the implementation easier and able to focus on studying the motivational behaviour of the agents. These agents are represented as the football players and the coach of each team. These agents are constructed using strong agents' attributes, thus enabling the investigation of motivation in multiple agent scenarios.

### 2.4.3 Concepts related to agent architecture

The architectures covered in this section are relevant to this project because it deals with agent architectures applicable to the design and implementation of a motivation agent.

#### Blackboard control architecture

This concept was developed by Lesser and Hayes-Roth, and involves an agent architecture containing features of *executor*, *agenda manager* and a *scheduler*. The *executor* performs the next action and makes changes to the memory. The *agenda manager* triggers known actions which satisfy the current events and which are placed into as *agenda of possible actions*. The *scheduler* evaluates possible actions and chooses the highest (most suitable condition) rating action to be executed next. Figure 2 illustrates the Blackboard Control Architecture.

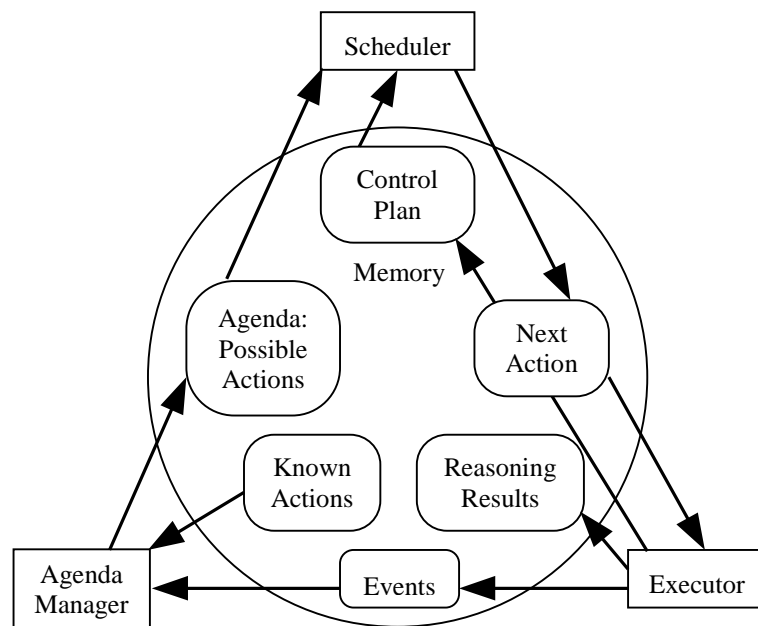


Figure 2. The Blackboard Control Architecture (redrawn from [Hayes-Roth 1993a])

The names used by the components in the *Memory* (inside the circle) represent clearly their functionality. These components inside *Memory* are all storage modules.

The memory in this agent's architecture acts in an important role for efficiency. It determines the next action, making real-time events achievable. Such phenomena are performed by the enabling of *agenda manager* and *scheduler* to compete for a real-time event; the highest rating action within these components determined in a specified time will be used.

Plans are changed if the actions taken in execution sequence are different from the data in *Control Plan* inside *Memory*.

The blackboard architecture can be expanded for multiple agents' scenario. Figure 3 illustrates the extended version of the blackboard architecture.

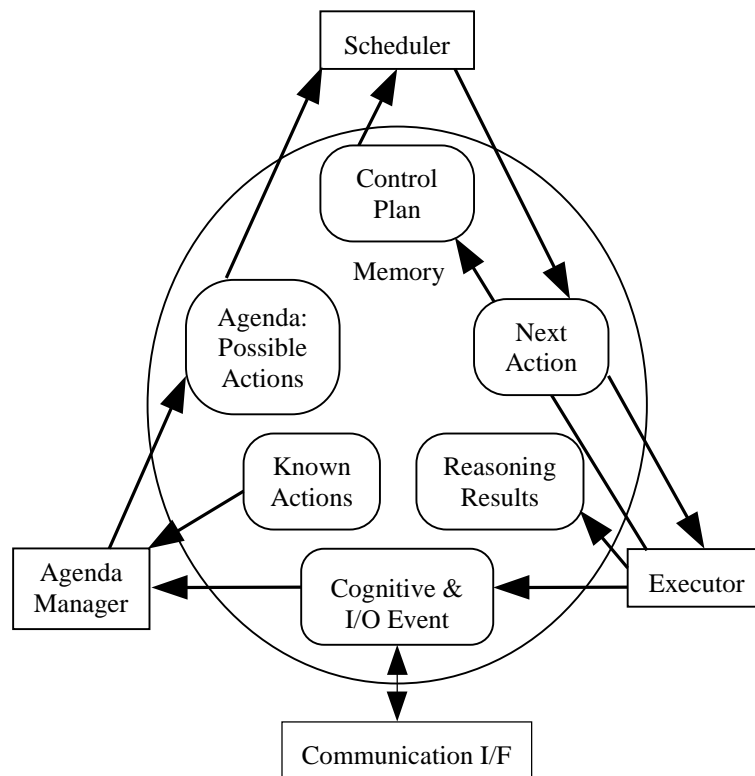


Figure 3. Extended version of Blackboard Control Architecture for multiple agents' scenario (re-drawn from [Hayes-Roth 1992])

The module *Communication* uses an infrared device to transfer data between robots (agents) in the same environment [Hayes-Roth 1993a]. Alternative data transfer protocols between agents can be used.

Hayes-Roth and colleagues used this architecture to develop the computer software OPM (Opportunistic Planning Model), which in a later stage was implemented into the software system BB1 [Hayes-Roth 1993a]. BB1 has been used for determining protein structures and designing construction site layouts. Both implementations had satisfactory results due to the efficiency of its agent's architecture.

### **Belief-Desire-Intention Model**

The Belief-Desire-Intention (BDI) model was originally developed by Bratman [1987], and used for implementing reasoning agents [Fiadeiro 1996; Singh 1998].

The concept of this model integrates a psychological view of how humans perform tasks in a sensible manner.

As expected from the nature of its name, this model contains components regarding to belief, desire and intention. Figure 4 illustrates BDI agent architecture.

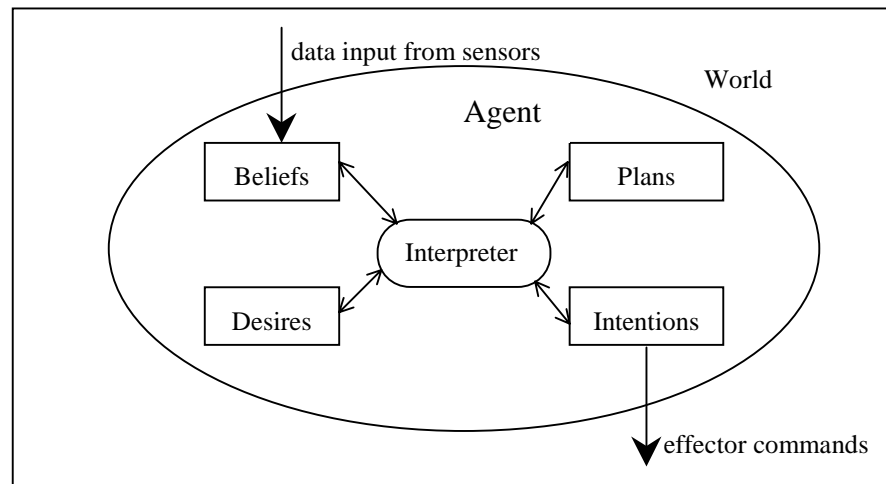


Figure 4. BDI Agent Architecture (re-drawn from [Fiadeiro 1996])

Other components in this architecture consist of *Plans* and *Interpreter*.

The *Beliefs* component reads data from the environment. The agent uses this data to update the changes on the current environment and hence this component is crucial for the agent. Previous events of the environment are also accumulated to aid the updating procedure [Singh 1998]. To sum up the concept of this component, it represents what the agent perceives and understands from its environment.

Data in the *Desires* component represents a desired state for the agent. Without this component, the agent would be goal-less and perform senseless random tasks.

The *Plans* component computes sequence or sequences of actions that enables the transition to a desire state. Actions generated in this component must not contradict themselves or the data in *Desires* component, otherwise delay for reaching a goal might occur.

The *Intentions* component is where the agent outputs its data to the environment, thus taking the next action. The actions determined from this component are directly influenced by *Desires* and *Plans*.

The *Interpreter* serves as a medium for data transfer between other components in the agent.

This agent's architecture enables the agent to perform suitable tasks to achieve a desire state as quickly as possible (if there is a possibility to achieve a goal).

### Will's architecture

*Will* is a software agent developed by Moffat and Frijda to model aspects of the human mind [Wooldridge 1995b]. The architecture of this agent consists of *perceiver*, *planner*, *reactor*, *predictor*, *executor*, *concerns* and *memory*.

The *perceiver* collects data from the outside world (real world for robots or computer environment for a synthetic agent) and passes them to memory. The *planner* has identical functionality to the *scheduler* in Blackboard Control architecture. The *reactor* is used when real-time events need to be performed by bypassing the *planner* and the

*predictor*. The *predictor* has an identical functionality to the *agenda manager* in Blackboard Control architecture. The *executor* is identical to the module of its counterpart in Blackboard Control architecture. *Concerns* form a layer round the *memory*, this module serves as a filter for relevant data to be store in *memory*. *Concerns* reflect on what the agent is aiming to do to achieve its intended goal or goals. Figure 5 illustrates Will's architecture.

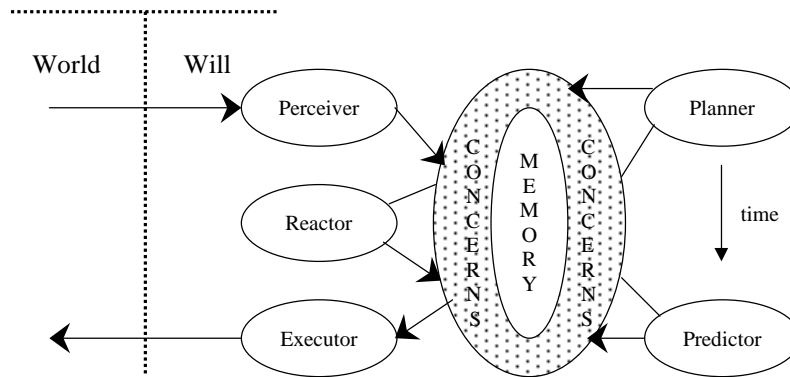


Figure 5. Will's full architecture (taken from [Wooldridge 1994])

This agent's architecture provides memory storage of relevant events which can be used to predict or plan the next action. The memory may store irrelevant events, but these data will have a low charge (value), thus being ignored by other modules [Wooldridge 1994].

Will was implemented as a prototype for playing games with the human user, which involves planning moves and reacting to those from the user. Therefore the architecture for Will is appropriate in a game domain.

### Abstract three level layer architecture

The abstract level layer architecture is a conceptual view of how the human mind works to react on the environmental changes and performing tasks for achieving an intended goal.

Davis developed this architecture using by four different layers: reflexive, reactive, deliberative, reflective [Davis 2001a; Davis 1998; Davis 1996].

The reflexive layer contains processes to allow immediate response to sensory events related to the environment. For example, when a human burns his hand by a candle fire, the hand will be forced to move away from the fire (as a reflex), even without the consent of that person. The reflexive layer (or reflex for an agent) may not be overwritten.

The reactive layer consists of capabilities and behaviours including explicit goal-oriented motivational states. An example of this agent's capability is to move towards food when hungry.

The deliberative layer is separated from the reflexive and reactive layers by an *attention filter*. The *attention filter* only passes relevant data for activation of the deliberative and reflective layers when necessary. Due to the fact that the deliberative and reflective layers take more computational processing and time, they are only used when the processes in reflexive and reactive layers cannot deliver the next suitable step for the agent.

The deliberative layer consists of cognitive behaviours such as explicit planning, decision, memory recall and scheduling.

The reflective layer (not necessary used for all agents) serves as a meta-control layer for monitoring the internal processing of an agent. This layer is not be used in this project. An alternative for this layer is to use an *Abstract Manager* (explained further detail later in the section).

Figure 6 illustrates the abstract three level layer architecture of an agent, which does not included the reflective layer.

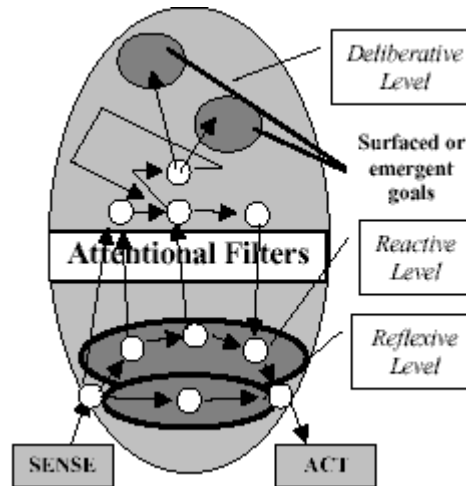


Figure 6. Abstract Three Level Architecture for a single Agent with Explicit Motivational Structures (taken from Davis)

Each of the modules is represented as circle, within which the programmer can pre-define which components are needed for an agent, depending on its environment and goals.

This project is to simulate 5-a-side football, therefore the environment consists of multiple agents that may have different tasks and goals. These agents from figure 6 may work as individual agents. Agents may be working with the collaboration of footballers in the same team. The collaboration between these agents is achieved by communication or architecture such as the one illustrated from figure 7, the architecture between multiple agents using the agents in figure 6.

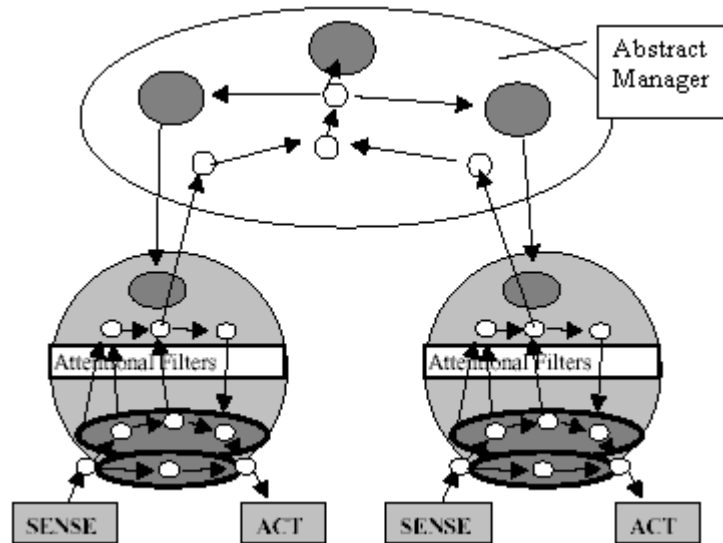


Figure 7. Abstract Three Level Architecture for Multiple Agents with Explicit Motivational Structures (taken from Davis)

There are three agents represented in figure 7, two single agents represented in the bottom which are identical to the agent in figure 6 and an agent on top.

The agent on top is called *Abstract Manager*, which can be performed as a medium of communication between agents. The *abstract manager* does also perform other tasks that require considerable computation, time or overall structuring of the environment. The *abstract manager* may not need to respond to real-time events (as opposed to other agents). Therefore it can do time-consuming tasks in the background such as planning, or re-scheduling for the other agents. Other agents can perform independently before data are received from the *abstract manager*.

The nature of the *abstract manager* is appropriate for this project, when this concept is integrated as a simulated football coach. In football, the coach instructs tactical plans to the team. The coach may plan prior to the game. During the game, the coach might need to re-plan if necessary when conditions in the game do not go well. These features of a coach can be allocated to the *Manager* in the 5-aside football simulation of this project.

Each agent (apart from the coach) will be represented as a football player in the simulation. If more agents are needed, then more agents can be linked to the *Manager*.

The architecture of agent from figure 6 is used as the model for building the single agent for this project. Such a model enables the analysis of motivational aspects of a single agent. Alternatively, the architecture of agents from figure 7 is implemented to study their behaviours in multiple agents' scenario, as opposed to individual single agents.

#### 2.4.4 Summary of the agent's architecture

The agents' architecture covered in the section 2.4.3 possesses certain degree of similarity, they are all representations of computational minds to simulate a human's partial functionality of mind (how a human thinks and reacts with the environment). Table 1 illustrates the components that identify the identical functionality on these agents' architectures.

<b>Blackboard Control Architecture</b>	<b>Belief-Desire-Intention Model</b>	<b>Will's Architecture</b>	<b>Abstract Three Level Layer Architecture</b>
Agenda Manager		Predictor	Deliberative layer
Scheduler	Plans	Planner	Part of Deliberative Layer
Executor	Intentions	Executor	Act
	Beliefs – Data from Sensors	Perceiver	
	Desire	Concerns	Part of Deliberative Layer
		Reactor	Part of Reflexive Layer

Table 1. Components in different architectures for identical functionality

The vehicle to explore the motivational behaviour for multiple synthetic agents' scenario is to design and implement agents using *Abstract Three Level Layer* architecture. Features in other agents are mapped to these agents, aiding in building the agent's architecture for this project.

## 2.5 Motivation and Goal

The aim of this project is to investigate motivation in a multiple agent scenario. Therefore it is important to explore the conceptual meaning of motivation [Sloman 1994; Norman 1994; Jenkins 1998; Beaudoin 1994; Davis 2001a; Davis 2001b]. The concepts of motivation and goal are linked to each other. For such reasons, this section is dedicated to describing these concepts.

Control states represents the internal behaviour of an agent, and may not manifest themselves as external behaviour [Davis 2001a]. These control states can be temporally brief or expansive.

Motivations and goals are influenced by the control states of the agent, as illustrated in figure 8.

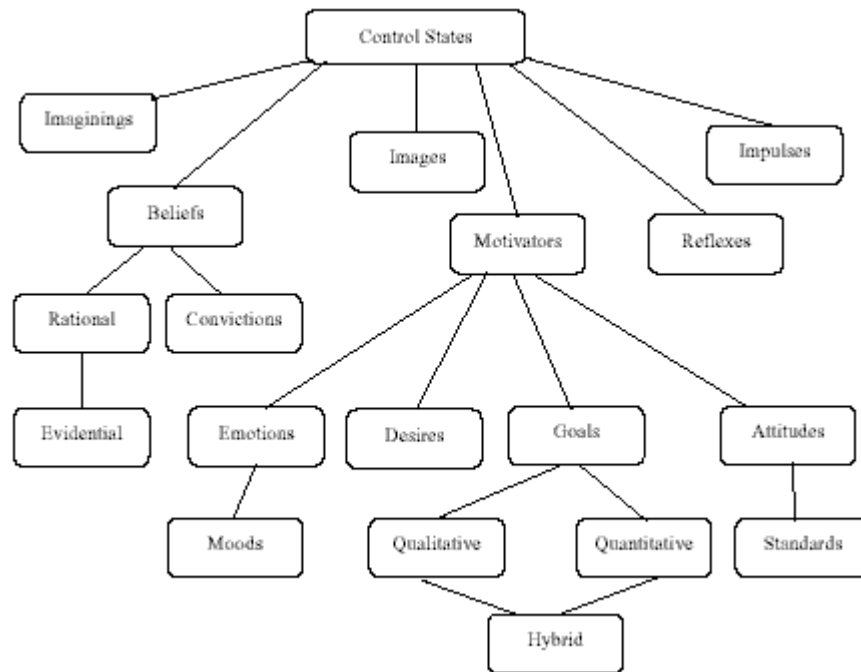


Figure 8. Control states (taken from [Davis 2001a])

Motivations, in context regarding to Artificial Intelligence, are referred to as reasons that push an agent into performing an action, and serves as the basis material for the constitution of tendencies [Ferber 1999]. Motivation for an agent only relates to actions which ultimately lead to goals. Motivators are dispositions to assess situations and respond to these situations and assessments in a certain manner [Davis 2001a].

Motivators are means for an agent to express its motivations. Sloman [1994] has defined twelve attributes in assessing motivators:

- Semantic content
- Motivational attitude
- Rationale
- Current belief
- Importance
- Urgency
- Insistence
- Intensity
- Plans
- Commitment
- Dynamic state
- Management information

*Semantic content* represents a proposition denoting a possible state or states, which may be true or false.

*Motivational attitude* of the agent represents the tendency of acting toward the *semantic content*, for instance ‘make true’ to obtain the ball for the footballer.

*Rationale* represents the reasoning towards a motivation or state.

*Current belief* of the agent represents what the agent currently believes: whether is true or false in respect to states or objects.

*Importance* is defined as whether the goal is worth pursuing; if not, this goal may be abolished. *Importance* may be intrinsic or may be based on an assessment of the consequences of certain actions.

*Urgency* is defined as the deadline for an intended goal. *Insistence* assesses the *importance* and *urgency* to determine an action or sets of actions to be performed.

*Intensity* influences the likelihood of continuing such actions to be carried out.

*Plans* are actions or sets of actions to be performed to achieve the goal.

*Commitment* represents the likelihood of performing such actions or sets of actions.

*Dynamic* state represents the situation of a motivator, for example 'being considered' or 'near completion'.

*Management information* represents the state of current relevant management of the agent.

These factors of Motivators proposed by Sloman helps in making a synthetic agent capable of expressing motivational aspects when pursuing its goals. Due to the implementation parsimony, not all of these factors are implemented; only those relevant to this project are chosen, and the scope is discussed in detail in section 3.1.

A Goal is a desire to go from one state to another or maintain a current state [Jenkins 1998]. Goals require planning and explicit differentiation between its external environment and the agent's internal environment [Davis 2001b]. An agent may have multiple goals at the same time. An agent can achieve multiple goals by performing certain sequence of tasks, or a certain task may spawn into multiple goals when performing different sequences.

The concepts of Motivation and Goals are essential to this project, since the agent possesses motivational capabilities. Therefore, the motivational factors and goal are integrated in the synthetic agents for the simulation in 5-side football.

## 2.6 Research of football simulation

The agents in this research interact in a football environment.

The topics covered in this section are related to the latest research in football simulation, mainly in software simulation aspects.

Recent researches in football simulation are related to hardware, software or their hybrid domains. In terms of hardware, they are related to robotics and functions such as sensory and visionary devices in the robot. In respect to the software domain, there are more topics that can be explored in the football context. These include efficiency in generating the best result for the next action of the agent, enhancement in reading data from sensory devices, machine learning (for prediction or generation for the next moves) or agent analysis (for studying agents behaviour).

The aim of this project is to simulate motivational aspects in multiple agents' environments through football simulation in the software domain. Therefore, other topics related to the research of football simulation are irrelevant for this research.

A yearly event called RoboCup (since 1997) attracts many enthusiasts in the football simulation field to compete with each other. The competitions are divided into five categories, they are:

- Simulation League
- Small-Size League (F-180)
- Middle-Size League (F2000)
- Sony Four Legged Robot League
- Humanoid League

At the current state, there is no competition in 5-aside football simulation in RoboCup. Apart from the Simulation League, all these competitions involve in using robot football players. Each League has different rules, which also includes a different number of robots for each team.

The Simulation League involves 11 software agents for each team in the football field.

The simulation of these football games is executed in software called RoboCup Soccer Server, which serves as the environment for these agents to interact in. This server accepts action commands from the distributed agents in every 100-milliseconds and updates the world state (environment) once every computer clock cycle [Stone 2000b]. The communications established between these agents are deliberately made to have limited range and capacity by using an unreliable communication channel. All the agents, including those from the opponent's team, can pick up the signal. Therefore, an agent must not fully rely on a team-mate's communication for generating the next action because such data is also available to the opponent's agents.

For further information about rules and regulations in RoboCup, refer to its corresponding Web site ([www.robocup.org](http://www.robocup.org)).

Journals from Tambe [2000], Tanaka-Ishii [2000] and Tomoichi [2000] illustrate their implemented software for analysing data retrieved from RoboCup, in simulation and robotic tournaments. Each of these examples of software are used for analysing agents in multiple behavioural level, and not for a single agent. This helps scientists in understanding and improving the performance of the football teams in such an environment.

One particular team has demonstrated an outstanding achievement in researches related to football simulation and robotics. Veloso and colleagues formed a football team called CMUnited for both simulation and robotics tournaments [Stone 2000a]. They were the champions in simulation league in 1998 and 1999, and 1997 in Small Size League.

The architecture in these agents used by Veloso and colleagues is called Layered architecture [Veloso 2000]. Each layer represents a task that the robot or simulated agent is capable of performing, such as a layer for controlling the ball, one for dribbling, and another for passing the ball. For details related to their research and the architecture of the agent, please refer to Stone [2000a].

There are some features and capabilities from their agents related to the football player [Stone 2000a; Stone 2000b] which can be use to integrate into the agents for this project. These are:

- Football player's capabilities
  - Kicking
  - Dribbling
  - Passing the ball
  - Ball interception (block opponent's passing)
  - Goal tending (shooting the ball to the goal)
  - Goal defending
- Football player's feature
  - Vision of detecting the ball in front of the player
  - Communication between agents

The capabilities mentioned above are identical to human's perspective in playing football.

The aim of this project is to use football simulation as a vehicle to generate the behaviour of the co-operation and competition between agents. Therefore rules and regulations of RoboCup will not be use in designing and implementing the simulation of this project. Such a decision is made to avoid diverting from the aims and objectives of the project. By focussing on great details on the football simulation may divert and degrade the analysis of motivation in a multiple agent scenario.

5-aside football is used as a vehicle for the agents to interact with such an environment. The rules of 5-aside football are simpler than 11-league football. The goalkeeper may go outside but cannot hold the ball beyond the penalty area. The goalkeeper cannot go across the half-side of the opponent's field. The ball and footballers cannot go beyond the field, and therefore the ball bounces back once hit to the side of the football field.

Details about how the footballers behave in the 5-aside football game are stated in chapter 3. For further details and rules for the 5-aside football, please refer to FIFA [2001].

## 2.7 Research Questions

This research aims to study and analyse the motivational aspects of multiple agents' scenario through 5-aside football. The points to be clarified throughout this research are:

- How does a Multiple Agent System work in 5-aside football environment?
- Does multiple agents work better than individual agent in the same team?
- How does motivation affect the performance of the agents?
- Which investigated architecture can score the most in the football game?
- What kind of trade-offs would be necessary to avoid the technological limitations (acceptable response time for adequate results) for the agents in such environment?

## 3 Design and analysis of the agent for the 5-aside football simulation

### 3.1 Requirements for a motivational agent

The agent in this research contains motivational capabilities for decision making and evaluation of its actions.

As mentioned in chapter 2.5, Sloman [1987; 1994] has defined twelve attributes in assessing motivators in an agent, they are:

- Semantic content
- Motivational attitude
- Rationale
- Current belief
- Importance
- Urgency
- Insistence
- Intensity
- Plans
- Commitment
- Dynamic state
- Management information

Explicit representation (in form of data or rules) of some of these attributes is considered in this research. Only a few of these attributes would be implemented due to the essence of such an attribute to be contained in the agent. The following attributes are considered here:

Semantic contents - rules within the agent for decision making.

Current belief - the agent perceives knowledge from the environment, and reacts to it according to its belief. For example, the footballer goes towards the ball when he senses its position.

Urgency - an evaluation where the agent judges how fast it needs to react to certain events, depending on its role towards such an event. For instance, the footballer in defend mode detects a ball approaching to its football goal, it is very urgent to react so that this footballer can either trap the ball or change the ball's direction.

Dynamic state - the agents in this simulation are integrated into a dynamic environment, that is, states in the environment change rapidly. The agent must be able to react according to the changes of the environment.

Management information - this is achieved by using manager for this research (illustrated in section 3.2) to enhance the co-ordination of the agents in the same team. Information is accumulated within the management agent, and this information can be then transferred to the relevant agent.

## 3.2 Representation of the agent architecture

The capabilities of the motivational agent requires to be integrated to the agent architecture.

The architecture used in the simulation of these motivational agents is one instance of CAM (Computational Agent with Motivation) Architecture, which is illustrated in figure 9. This architecture is identical to the abstract three level layers architecture illustrated in section 2.4.3.

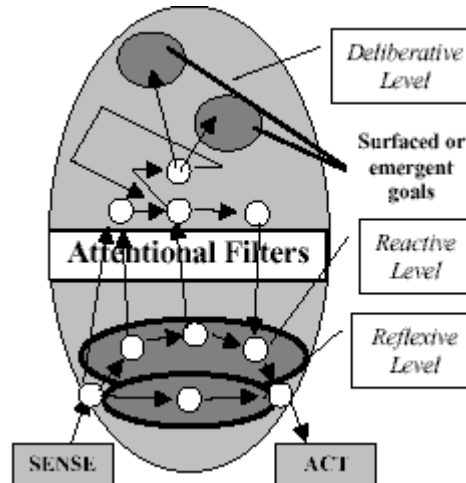


Figure 9. Abstract Three Level Architecture for a single Agent with Explicit Motivational Structures (taken from Davis)

This architecture is divided into three different layers.

- Reflexive Level (level 0)
- Reactive Level (level 1)
- Deliberative Level (level 2)

Functions in each layer differ in their complexity, in terms of processing-time or computational quantity required to process them, and are explained in section 3.4.

Communication is established between agents of the same team by data sharing. Data sharing in this context is either information shared between corresponding agents, or suggestions/instructions to be transferred to the corresponding agents regarding some actions.

Although an agent may receive instructions from another agent, it will evaluate such information and decide whether to follow these instructions.

An alternative architecture to establish co-operation between agents is to enable a team containing the Abstract Manager, as illustrated in figure 10.

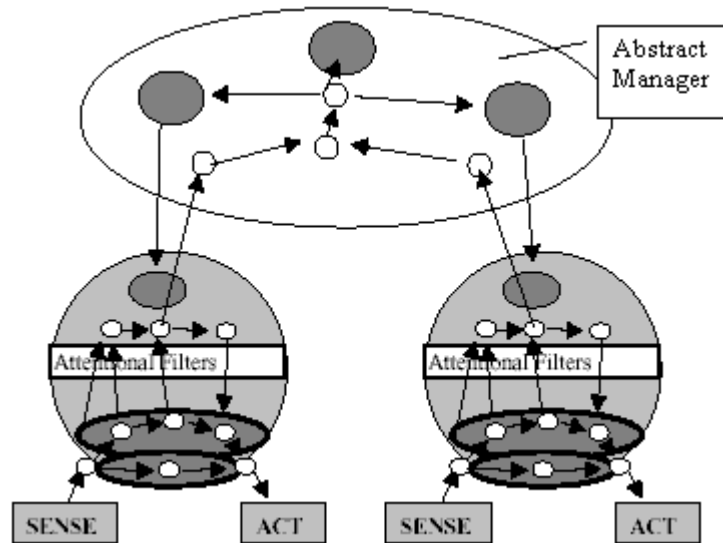


Figure 10. Abstract Three Level Architecture for Multiple Agents with Explicit Motivational Structures (taken from Davis)

The Manager is used for aiding agents' co-operation aspects, in terms of communication and planning for the whole team.

This architecture enables the co-operation of the team to work more efficiently because time-consuming decisions and manipulations are been handled by the Manager instead of individual agents. Such a concept favours other agents to respond to real-time events or perform manipulation related to other events instead of processing time-consuming yet less urgent decisions.

### 3.3 Features and capabilities for an agent in the 5-side football simulation

The agent adapts and acts upon its environment. In this research, the scenario chosen for the agents to interact in is 5-side football.

The agent is represented as a footballer with the role of either:

- Outfielder
  - Striker
  - Defender
- Goalkeeper

According to the 5-side football rules [Fifa 2001], each team must have 5 footballers in the field during the game. Situations such as all outfielders being strikers or all being defenders are allowed, as long as there is one and only one footballer taking the goalkeeper role. This project is mainly focus on motivational aspects for multiple agents. For the simplicity of the simulation, each team contains two agents taking the role of striker, another two agents taking the role of defender, and one agent as the goalkeeper.

The footballers contain the following features and capabilities:

- Capabilities
  - Running

- Kicking
- Dribbling
- Passing the ball
- Block opponent's passing
- Shooting the ball to the opponent's football goal
- Ball catching (for the goalkeeper)
- Trapping the ball (for the outfielder)
- Features
  - Sensing the ball in the football field
  - Sensing other footballers
  - Communication between agents of the same team
  - Energy Level
  - Goal Management
  - Motivation generation and management (using motivators from section 3.1)

Table 2 illustrates the features and capabilities of these footballers mapped into the CAM Architecture from figure 9.

<b>Reflexive Layer</b>	Detecting the ball Trapping/Catching the ball Energy Level
<b>Reactive Layer</b>	Kicking Shooting the ball to the goal Blocking opponent's passing Dribbling 1 (push the ball to a particular position) Dribbling 2 (avoiding an opponent to get the ball) Passing the ball (may require communication)
<b>Deliberative Layer</b>	Motivation management Urgency management Priority management

Table 2. Features and Capabilities in a CAM architecture

An incremental approach is used for building these agents. The simulation prototype contains the football field and agents having simple roles, that is, possessing a Reflexive Layer only. When these functions are tested and approved, a Reactive Layer will be built for these agents, and finally the Deliberative Layer will be added.

Functions can be manipulated by the manager (illustrated in figure 10) to enhance the performance of the agents and the overall team.

- Allocating the ball
- Passing the ball
- Overall planning for relocating agents position in the field

This research investigates whether the overall performance of the football team differs with a manager.

Each agent contains energy level and each task made consumes its energy.

The energy level for each footballer is measured in units of a thousand, that is, the energy value is 1000 when full. If the energy level is below the threshold, in order to recover energy, the footballer tends not to run. Energy use for a footballer can be measured in the following:

- Walking 0 (not taking any energy)
- Running -1 to -3 depending on the speed
- Shooting or passing the ball -1 to -3 depending the displacement of the ball
- Resting +10

Speed of walking is slow in comparison to running for the footballer. The footballer only walks when there is no particular task to do. Resting enables the footballer to regain energy. The footballer may rest instead of performing a task, if it is not urgent.

The footballer contains a value to represent its energy value. The threshold for the energy value is predefined in the system.

The footballers and the ball are considered to be agents in the environment, having their own attributes regarding the functionality and behaviour of these agents. These attributes are:

- Footballer's attributes
  - Energy Level
  - Motivators
  - Its position from x and y axes
  - Its velocity of displacement with regard to x and y axes (negative or positive value depending on direction)
  - Value representing which team the player belongs to
  - Boolean value representing whether the player has the ball or not
  - Footballer's role
    - Goalkeeper
    - Outfielder
      - Attack mode (striker)
      - Defend mode
- Ball's attributes
  - Its position from x and y axes

- Its velocity of displacement regarding to x and y axes (negative or positive value depending on direction)
- Value representing whether the ball has been possessed by a footballer. In case the ball is possess by a footballer, this value represents which team this player belongs to

Depending on the footballers' role, it has different tasks to fulfil:

- Goalkeeper
  - Avoid the ball to get into its team's football goal
  - Catch the ball
  - When having the ball, pass to the nearest team-mate
  - Do not pass the midfield line
- Outfielder
  - Block opponent's player to avoid getting a pass ball
  - Get opponent's ball
  - Get the unattended ball
  - Trap the ball
  - Dribble the ball until further action
  - Pass the ball to a team-mate
  - Shoot the ball to the opponent's football goal

These roles make use of the agent's capabilities for its performance.

### 3.4 Roles of the agents in the CAM Architecture

Figure 11 illustrates a state transition diagram for the agent's roles mentioned in section 3.3.



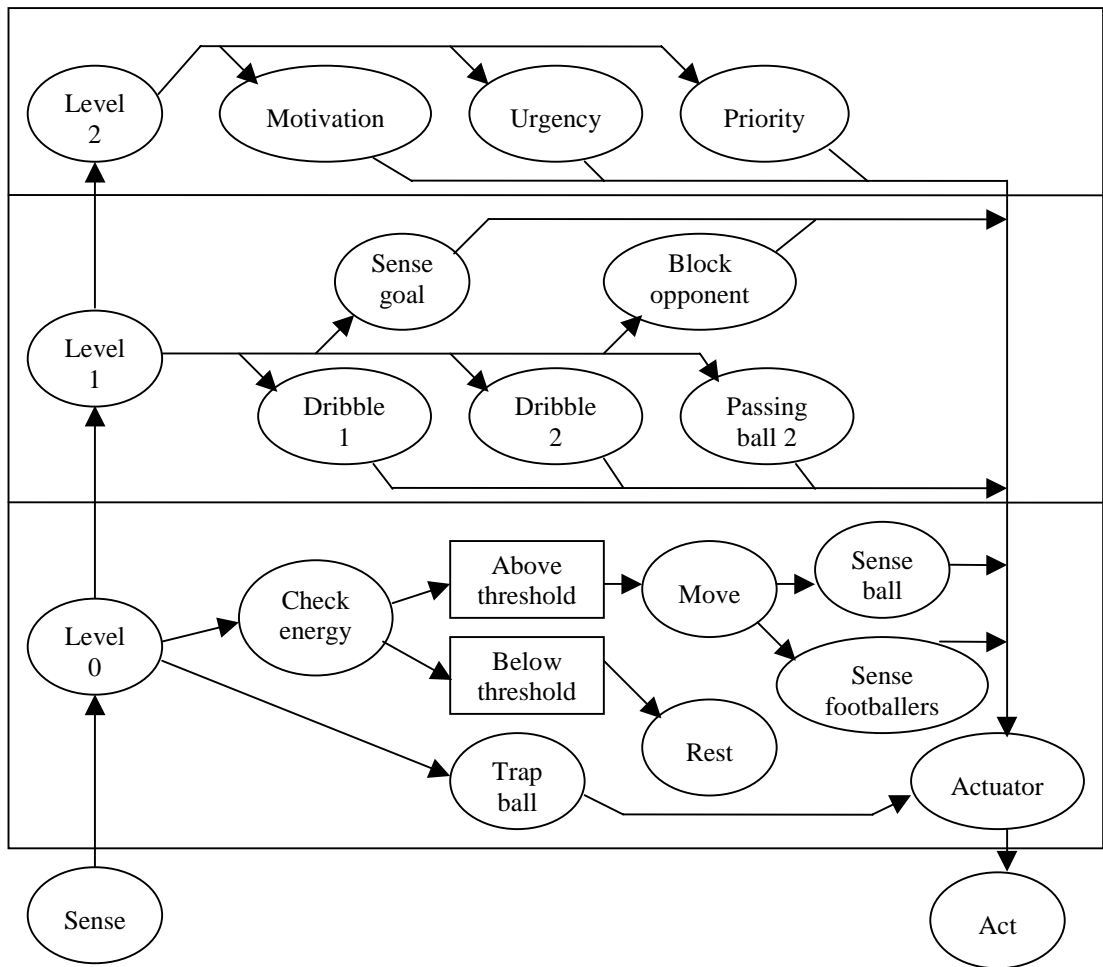


Figure 12. CAM Architecture for an outfielder

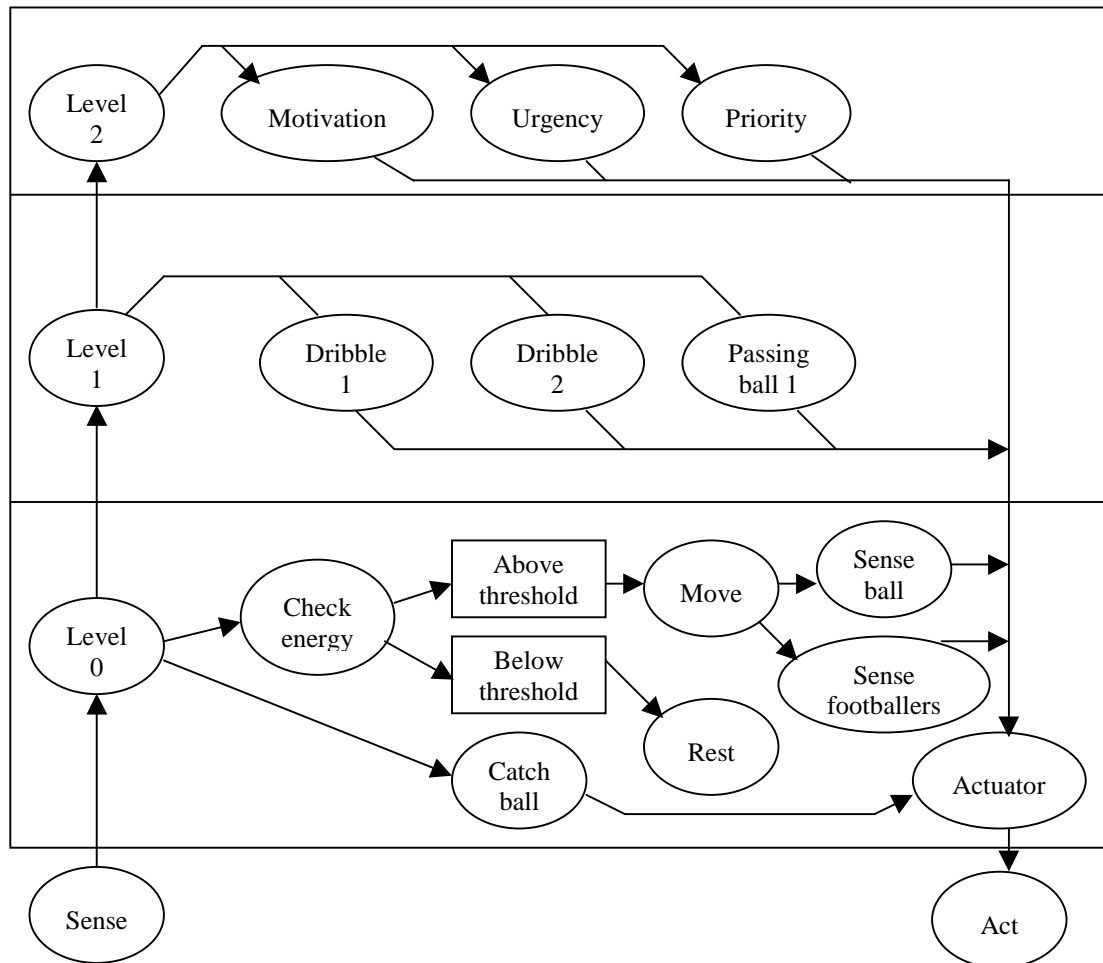


Figure 13. CAM Architecture for a goalkeeper

Functions in each layer correspond to the processing time and reaction time necessary to perform such tasks. For example, functions in level 0 are reflexive actions, meaning these actions can be performed immediately without the need to do much calculation or rules checking. On the other hand, a function in level 2 requires more computation before reaching a result for its next action.

All functions in figures 12 and 13 should be connected to *abstract manager* with a bi-directional arrow. For purposes of clarity in these figures, such connection between *abstract manager* is not shown.

The functions in the reflexive layer (level 0) are simple actions.

The functions in the reactive layer (level 1) require more processing than those in the lower layer and need to perform the following tasks:

**Sense goal:**

1. Sensing the opponent's goal
2. Calculate the best angle to score a goal
  - Is the goalkeeper able to catch the ball if the footballer uses this angle?
  - If the footballer uses this angle, is the path clear to do so?
3. Select the best angle from step 2, and set speed to shot the ball to the goal

**Dribble 1:**

1. Have ball
2. Sense nothing
3. Dribble around with the ball

**Block opponent:**

1. Sensing the ball
2. Sensing the distance and the path between the ball and the nearest opponent
3. Go to the nearest position to block that path

**Dribble 2:**

1. Sensing the nearest opponent
2. Calculate the best direction to move away from this opponent
  - Is moving back to its own field the only alternative
  - If not, do not move back (try to move forward at all times to get closer to the opponent's goal)
3. Move to the best direction from step 2

**Passing a ball to a team-mate 1 (goalkeeper only):**

1. Sense the nearest team-mate
2. Choosing a team-mate to pass the ball
  - Is that team-mate the closest to pass the ball?
  - Is the path clear to pass it to that team-mate?
    - If not, repeat checking for the next nearest team-mate
3. Pass the ball to the best team-mate chosen from step 2

**Passing a ball to a team-mate 2 (outfielder only):**

1. An opponent is approaching (a threshold is used to determine whether is close)
2. Sense all team-mates and opponents within the range
3. Choose the best team-mate to pass the ball
  - Is the path clear to pass to this team-mate?
  - Is that team-mate the closest to pass the ball?
4. Pass the ball to the best team-mate chosen from step 3

The actions taken must follow the sequences before completing the task.

Motivation management, urgency management and priority management featured in the deliberative layer are explained in details in section 3.6.

### 3.5 Conditions for agent's task execution

Functions stated in section 3.4 allow an agent to generate sensible behaviour.

For this agent's behaviour, pre-requisites have to be satisfied when a new action is performed. Tables 3 and 4 illustrate these conditions for a footballer and the ball respectively. Each rule is numbered for references used in section 3.6.

<b>Behaviour</b>	<b>Layer Level</b>	<b>Pre-requisites</b>	<b>Post-condition</b>
1. Not doing anything (Rest from State Transition Diagram)	0	Do not have the ball Energy level below threshold	Stop
2. Moving (Move from State Transition Diagram)	0	Do not have the ball Energy level above threshold	Player moving
3. Moving to the ball (Sense Ball from State Transition Diagram)	0	Do not have the ball Player moving Ball allocated  Ball not possess by anyone Or Ball not possess by a team-mate and not in defend mode	Player moving (towards the ball)
4. Dribbling 1 (Dribble 1 from Footballer role)	1	Have the ball Nothing sense within the range	Player moving (with the ball)
5. Trapping the ball (Sense Ball from State Transition Diagram)	0	Do not have the ball Ball not possess by anybody Touch ball Outfielder mode	Have the ball Halt the ball
6. Catching the ball (Sense Ball from State Transition Diagram)	0	Do not have the ball Ball not possess by a team-mate Touch ball Goalkeeper mode	Have the ball Halt the ball
7. Kicking the ball to the goal (Sense Opponent's Goal from State Transition Diagram)	1	Have the ball Close to opponent's goal	Shot the ball Do not have the ball Set ball's threshold and speed
8. Blocking opponent's pass (Sense Opponent from State Transition Diagram)	1	Player moving Do not have the ball Ball allocated Ball not possess by a team-mate Defend mode	Move to the location that blocks opponent getting the ball from a team-mate

9. Dribbling 2 (Dribble 1 from Footballer role)	1	Have the ball Opponent is too close (below the threshold)	Move away from opponent Displacing the ball
10. Passing the ball to a team-mate 1 (Passing ball 1 from Footballer role)	1	Have the ball Sense team-mate Path clear to pass the ball to a team-mate Goalkeeper mode	Pass the ball Do not have the ball Set ball's threshold and speed
11. Passing the ball to a team-mate 2 (Passing ball 1 from Footballer role)	1	Have the ball Sense team-mate Opponent is getting close Path clear to pass the ball to a team-mate Outfielder mode	Pass the ball Do not have the ball Set ball's threshold and speed
12. Planning for overall strategy	Motivational Manager	Communication with all the footballers of the team	Produce strategy and inform corresponding players

Table 3. Footballer's behaviour

Behaviour	Pre-requisites	Post-condition
13. Stop	Below moving threshold	Speed equals to 0

Table 4. Ball's behaviour

Layer levels from the footballers correspond to those outlined from Table 2. Behaviours are only activated when the corresponding layer is in use.

Rules are laid out to the system to represent the behaviours stated in Tables 3 and 4 as in pseudo code 1.

### For the footballer

```

1. IF (!have ball) THEN
    1. IF (Energy <= EnLevel) THEN stopped /* Rest */
    2. ELSEIF (Energy > EnLevel) THEN move /* Move Around */
2. IF (!have ball) AND (move) AND (Sense Ball) THEN
    /* Agent has enough energy to move and sense a ball */
    1. IF ( Ball(Possess) != Sameteam) THEN
        1. IF ( Self(mode) = Outfielder) THEN block the opponent's
            pass /* Move to Opponent */
        2. ELSE ( Ball(Possess) = No-one) THEN go to the ball /* Move to
            ball */
3. IF (have ball) AND (Sense Nothing) THEN move with the ball to the
    opponent's field /*Dribble 1*/
4. IF (!have ball) AND (Touch ball) THEN /* just touch the ball */
    1. IF ( Self(mode) = outfielder ) AND ( Ball(Possess) = no-one )
    THEN
        /* can only possess the ball if no one is possessing it */
        Ball(Possess) = its team, have ball = true, ball speed = 0

```

```

2. IF ( Self(mode) = goalkeeper ) AND ( Ball(Possess) != team-
mate) THEN
    /* can possess the ball disregarding if it's possess by an
    opponent, but no need to get the ball from a team-mate's
    possession */
    Ball(Possess) = its team, have ball = true, ball speed = 0
5. IF (have ball) AND (Sense Opponent's Goal) THEN /*Shoot to goal*/
    SHOOT, have ball = false, set ball speed and threshold
6. IF (have ball) AND ( Self(mode) = goalkeeper) THEN
    IF (Sense team-mate) THEN Pass to team-mate, have ball = false,
    set ball speed and threshold /* Pass to team-mate */
7. IF (have ball) AND ( Sense Agent(other team) ) AND ( Self(mode) =
outfielder) THEN
    /* an opponent is approaching while possessing the ball, this
    does not apply to goalkeeper */
    1. IF ( Agent(other team) < Danger range) THEN Move away
    /*Dribble 2*/
    2. ELSEIF ( Agent(other team) > Danger range) THEN Pass to team-
    mate, have ball = false, set ball speed and threshold /* Pass
    to team-mate */
8. IF (have ball) AND (Energy <= EnLevel) THEN
    /* possess the ball but too tired to move around */
    Pass to team-mate, have ball = false, set ball speed and threshold
    /* Pass to team-mate */

```

### For the ball

```

9. IF ( ball energy < threshold) THEN Stopped /* Rest */

```

#### Pseudo code 1. Rules of the agent's tasks

An agent use decision making to execute a task by checking which rule it satisfies. In case of multiple rules chosen, with care selection, it would execute non-contradictory rules only. Otherwise, it needs to choose the preferable rule according to the motivational factors and priorities within the agent.

As shown in rule 1, when the energy value is too low, the footballer will move less to restore energy, similar to resting in real-life footballer. If the footballer has enough energy, then it will move randomly. Rules 2 to 7 would only be true if the energy was above the threshold.

Rule 2 enables the footballer to detect the ball. Once the ball's position is located, if the footballer is in 'outfielder mode' and the ball is controlled by an opponent's player, then this football agent will block another opponent's player to avoid getting a passed ball. Otherwise, it will head straight to the ball's location to get it.

Rule 3 allows the footballer to freely move around the field with the ball if it has not sensed any other footballers around or is near the opponent's goal.

Rule 4 distinguishes whether it is a goalkeeper catching the ball or an outfielder trapping the ball. In the case of the goalkeeper, it holds the ball and avoids the ball being possessed by the opponent's player. On the other hand, an outfielder can only have the ball if it is not possess by anybody.

Rule 5 enables the footballer to shoot the ball to the opponent's goal if it has the ball and is nearby the goal.

Rule 6 enables the goalkeeper to pass the ball immediately to a team-mate.

When passing the ball to a team-mate, it is preferable to pass it to the team-mate with a safe path (making it impossible for an opponent's player to have the ball in the middle of the pass).

Rule 7 makes a decision for the footballer as to whether to dribble or pass the ball to a team-mate when an opponent's player is approaching.

Rule 8 enables the football to pass the ball immediately to a team-mate when having the ball and not having enough energy to run on the football field.

The footballer has a limited range for sensing the players' and the ball's positions, so that the agent would not waste its computational power and time to process all its data.

The threshold for the ball represents the momentum left for the ball to travel. As shown in rule 9, if this value is 0, the ball has to stop.

### 3.6 Details on agent's components and means of execution

As covered from section 3.5, each agent has to comply with certain conditions to be able to perform the following tasks. This section will illustrate the details on these conditions are evaluated to consideration whether such task can be performed.

In addition to agent's roles, components in the deliberative layer of the agents are explained in detail.

The following functions are agent's behaviours which was covered in section 3.5 :

- Outfielder
  - Agent's displacement without the ball
    - Rest (Rules 1 and 13 from Tables 3 and 4, Rest from State Transition Diagram, Rules 1.1 and 9 from Pseudo code 1)
    - Move around (Rule 2 from Table 3, Move from State Transition Diagram, Rule 1.2 from Pseudo code 1)
    - Move to ball (Rule 3 from Table 3, Move then Sense Ball from State Transition Diagram, Rule 2.2 from Pseudo code 1)
    - Blocking opponent's pass (Rule 8 from Table 3, Move then Sense Ball then Block Opponent from State Transition Diagram, Rule 2.1 from Pseudo code 1)
  - Action regarding to the footballer controlling ball
    - Trap ball (Rule 5 from Table 3, Move then Sense Ball then Ball Obtained from State Transition Diagram, Rule 4.1 from Pseudo code 1)
    - Shoot the ball to the goal (Rule 7 from Table 3, Carrying the ball then Sense Opponent's Goal then Shoot to the Goal from State Transition Diagram, Rule 5 from Pseudo code 1)
    - Pass the ball to a team-mate (Rule 11 from Table 3, Carrying the ball then Sense Opponent then Pass to a teammate from State Transition Diagram, Rule 6 and 7.2 from Pseudo code 1)

- Dribble (1) with the ball (Rule 4 from Table 3, Move then Ball obtained then Nothing then Dribble from State Transition Diagram, Rule 3 from Pseudo code 1)
- Dribble (2) away from the opponent (Rule 9 from Table 3, Carrying the ball then Sense opponent then Dribble from State Transition Diagram, Rule 7.1 from Pseudo code 1)
- Goalkeeper
  - Agent's displacement without the ball
    - Rest (Rules 1 and 13 from Tables 3 and 4, Rest from State Transition Diagram, Rules 1.1 and 9 from Pseudo code 1)
    - Move around (Rule 2 from Table 3, Move from State Transition Diagram, Rule 1.2 from Pseudo code 1)
    - Move to ball (Rule 3 from Table 3, Move then Sense Ball from State Transition Diagram, Rule 2.2 from Pseudo code 1)
  - Action regarding to the footballer controlling ball
    - Catch the ball (Rule 6 from Table 3, Move then Sense ball then Ball obtained from State Transition Diagram, Rule 4.2 from Pseudo code 1)
    - Pass the ball (Rule 10 from Table 3, Carrying the ball then Sense Opponent then Pass to a teammate from State Transition Diagram, Rule 6 from Pseudo code 1)
    - Dribble (1) with the ball (Rule 4 from Table 3, Move then Ball obtained then Dribble from State Transition Diagram, Rule 3 from Pseudo code 1)
    - Dribble (2) away from the opponent (Rule 9 from Table 3, Carrying the ball then Sense Opponent then Dribble from State Transition Diagram, Rule 7.1 from Pseudo code 1)

The outfielders and the goalkeeper behaviours are similar, only minor addition is required to distinguish agents in different roles.

The following sub-sections explain how each of these functions operate.

### 3.6.1 Rest

An agent contains a value for its energy level. During the football game, energy would be consumed depending on the activities involved. The agent may reach a certain state of being too tired, meaning the energy level too low to do certain tasks. When this state arises, the agent needs to stop running to regain its energy.

If the agent is resting, it will regain 10 units in this simulation, and the agent would not move in that particular simulation cycle.

However, the agent would still be able to control the ball if necessary even when its energy is low. This capability to control the ball even when the energy is low enables the footballer to get away from an opponent or kick the ball if necessary, allowing the agent to deal with such extreme circumstances.

### 3.6.2 Move around

The footballer is freely able to move around when there is no particular task to perform. This action does not consume any energy from the agent because it is walking around as opposed to running (which takes energy) to a particular position of the field.

The goalkeeper is freely to move around inside its penalty area, and the defenders are freely to move around their side of the field. The reason for not having defenders and goalkeepers moving around the whole football field is to have these footballers in the most appropriate location of the field to be prepared for the opponents to go to these territories.

The footballer's turning is in multiple of 45 degrees. Due to the high resolution used in this simulation, the footballer can turn and travel to any direction on the football field.

### 3.6.3 Move to ball

The footballer will move to the ball if it is not possessed by one of its team-mates. However, the goalkeeper cannot go to the opponent's half of the field, even if the ball is nearby.

### 3.6.4 Block an opponent

When the footballer does not have the ball, it may prevent an opponent getting the ball. The footballer would determine which is the ideal position to block such an opponent, and it requires motivation to be able to perform such a task. This capability of determining the ideal position to go will be explained in details in section 3.6.10.

### 3.6.5 Get ball

When the footballer touches the ball, it will possess it provided that the ball is not possessed by another footballer. The footballer has the reflex to trap the ball, even without the consent of the footballer to get the ball.

### 3.6.6 Catching ball

Catching the ball is permissible by the goalkeeper only. This behaviour is also a reflex for the goalkeeper. The only difference between catching a ball and getting a ball for an outfielder is that the goalkeeper can hold the ball while moving around in its penalty area. If the case of the goalkeeper having the ball outside its penalty area, it can travel with the ball only by dribbling, and it is permissible for an opponent to get the ball.

### 3.6.7 Shooting to goal

When the footballer senses the opponent's goal and has the ball, it will shoot the ball into that goal in an attempt to score.

When shooting the ball, the footballer will use its energy to add to the speed of the ball, making the ball travel faster than dribbling or passing.

To avoid the possibility to predicting where the footballer is shooting the ball, it uses a random generator to determine at which position the footballer is aiming.

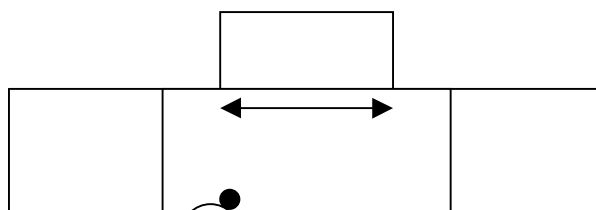


Figure 14. Footballer aiming for the goal (not to scale)

Figure 14 illustrates the situation where the footballer can shoot the ball at the goal. The footballer can be aiming to shoot anywhere between the arrow, excluding the goal posts.

Once the footballer steps into the penalty area, it will try to aim the position at which to shoot the ball. The randomiser for choosing the target makes the opponent's goalkeeper unable to predict where the ball is going to before the ball has been shot.

### 3.6.8 Passing the ball

The footballer passes the ball to a team-mate when it is possible and sensible to do so. With motivators, the footballer is able to determine whether is a safe pass or the most suitable team-mate to pass to, which will be explained in details in section 3.6.10.

### 3.6.9 Dribbling

The footballer dribbles the ball only in the two following circumstances:

1. Have the ball, no other preferable task to do, dribble to the opponent's field.
2. Have the ball, opponent nearby, dribble the ball away from the opponent.

In the second circumstance, the footballer can determine which is the best direction to dribble away from the opponent by using motivators, which will be explained in details in section 3.6.10.

### 3.6.10 Motivators

Motivations are reasons that push an agent into performing an action, and serves as the basis material for the constitution of tendencies [Ferber 1999]. Motivation for an agent relates to an action which lead to goals, as described in section 2.5.

Motivators enhance the agent's performance to achieve its goal. Agent's goal may change depending on the current state or other agents' actions in the environment. Therefore, the method which an agent performs a task to achieve an intended goal may change at any stage in the environment. For this simulation, the outfielders' goal is to score and goalkeepers to protect opponents from scoring.

Certain tasks for the agents contains motivators, which are:

1. Block opponent
2. Dribble
3. Passing the ball to a team-mate

Other motivators are also featured in the agents' components, as covered in section 3.1, which are : *Semantic Contents*, *Current Belief*, *Urgency* (explained in details in section 3.6.12), *Dynamic State* and *Management Information* (explained in details in sections 3.2 and 3.6.14). These Motivators may be executed concurrently.

*Blocking opponent*, *Dribble* and *Passing the ball to a team-mate* are motivators which an agent chooses only one of them to be executed. The agent needs to distinguish which

of these three tasks to choose from in a particular situation, not only because they cannot be executed simultaneously (three of them perform tasks differently), the agent needs to perform one of these three tasks to lead to its goal, depending on its role.

Motivators mentioned throughout this project are related to these three tasks. Other five motivators such as *Semantic Contents*, *Current Belief*, *Urgency*, *Dynamic State* and *Management Information* are mentioned and pursued research separately throughout this project to study their merits for the agent's behaviour.

The following subsections describe how *block opponent*, *dribble* and *passing the ball to a team-mate* are functioned for each of these tasks.

### Block opponent

When an opponent has the ball, the footballer might try to block another opponent to avoid a pass being made. When such situation arises, the footballer must be able to determine which is the ideal position to go to in order to block the opponent.

In the case where the footballer is not between the ball and the opponent that it wants to block, it will go either to the ball or opponent, whichever is the closest.

In case the footballer is between (horizontally or vertically) the ball and the opponent to be blocked, such as cases illustrated in figure 15, the footballer has to determine the closest point to go to.

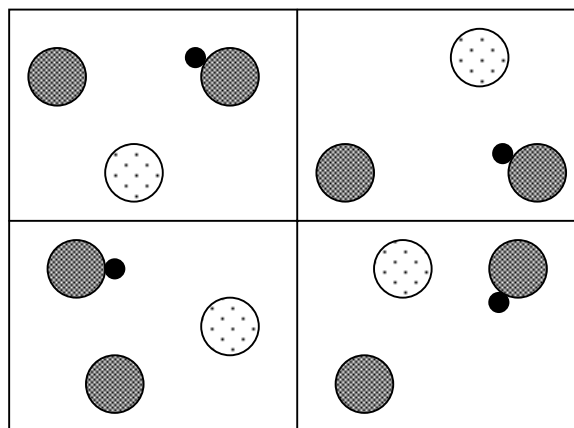


Figure 15. Examples of a footballer between two opponents (one with the ball)

The ideal position for the footballer to go to in this situation is a perpendicular point between the opponents. Figure 16 shows an example on how to determine the ideal position *id*, where position *id* is the perpendicular between the two opponents.

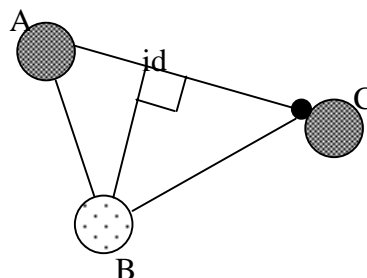


Figure 16. Example of getting the ideal position *id* between two opponents

To find the perpendicular point, calculate the distance between the three agents, as shown in figure 17 for lengths *a*, *b* and *c*.

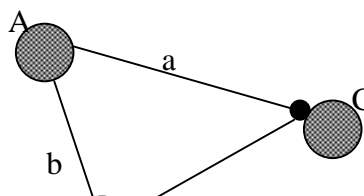


Figure 17. Calculating distances between agents

Using these distance values,  $a'$  can be determined. The value  $a'$  is a length between object A to the perpendicular point  $id$ .

The following formula enables to calculate the ratio  $a'$  from the example:

$$a' = \frac{a^2 + b^2 - c^2}{2a}$$

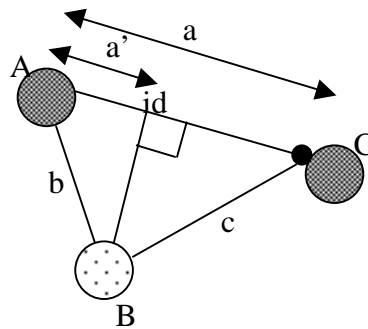


Figure 18. Length  $a'$  between object A and C

Figure 18 illustrate the length  $a'$ . By having the value  $a'$ , position  $id$  is derived from the ratio of  $a'$  and  $a$ .

$$ratio = \frac{a'}{a}$$

This means that the position  $id$  is the point between the line of object A and C with length  $a'$ .

The line between object A and C is the path where the opponent can pass the ball to the other opponent. By calculating the position  $id$ , the footballer goes to that position, which is the closest point to block the opponents' pass.

The footballer chooses the closest opponent to block. Alternatively, by allowing communication between team-mates, the footballer can choose the closest opponent that is not to be blocked by another team-mate.

### **Dribbling**

When the footballer has the ball, it will dribble the ball.

If the footballer does not sense any opponent nearby, it will dribble the ball to the opponent's half of the field.

In circumstances that the footballer senses an opponent, it needs to attempt immediately to dribble away from this opponent.

First, the footballer considers the closest opponent it can sense.

In the case where the opponent is not close, the footballer carries on dribbling the ball to the opponent's half of the field.

On the other hand, if the opponent is close, the footballer goes in the opposite direction to the opponent, as an example from figure 19.

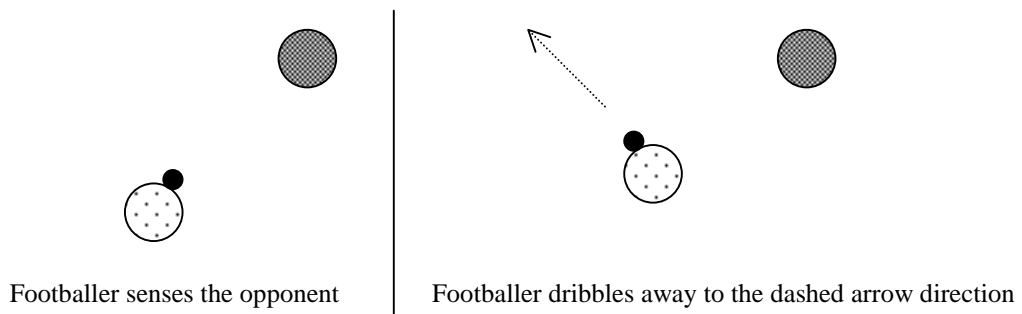


Figure 19. Footballer dribbling away from opponent

The footballer may dribble away from the opponent in the opposite horizontal or vertical direction, depending on the distance between them.

Additionally, if the opponent is too close, the footballer will swirl the ball to run away. This technique overwrites the footballer to run in the opposite direction from the opponent, because this method will enable the footballer to run away without dribbling the ball back to its field. Figure 20 shows the footballer swirling the ball.

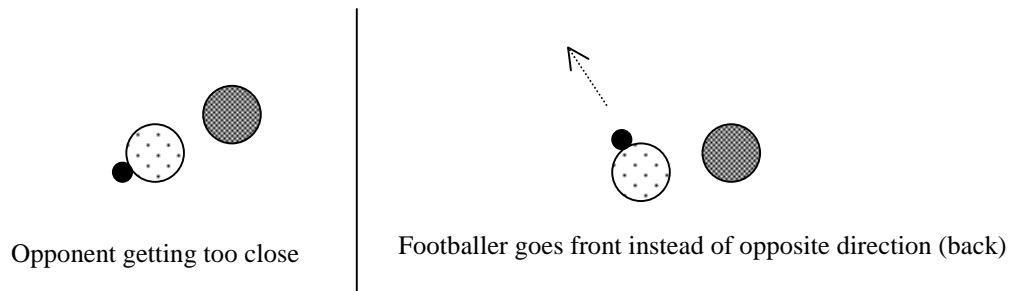


Figure 20. Footballer swirling the ball

With these techniques, the footballer will be motivated to dribble to the opponent's half of the field and get away from the opponent.

### Passing the ball

When a goalkeeper has the ball or an outfielder who has the ball is approached by opponents, the footballer needs to pass the ball to a team-mate if possible.

The footballer needs to be able to determine whether it is safe to pass to the corresponding team-mate.

First the footballer considers the closest team-mate.

If the footballer is between the team-mate and the closest opponent, such as examples illustrated in figure 21, it is safe to pass to the team-mate. The reason it is safe for such pass is because it will take less time and distance for the team-mate to get the pass than for the opponent to get the ball.

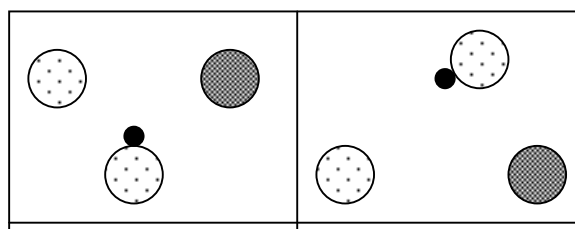


Figure 21. Footballer between team-mate and opponent

When the footballer is not between the closest opponent and the team-mate, such as examples illustrated from figure 22, it is a safe pass if the distance between the footballer and the team-mate is closer than that between the footballer and the opponent.

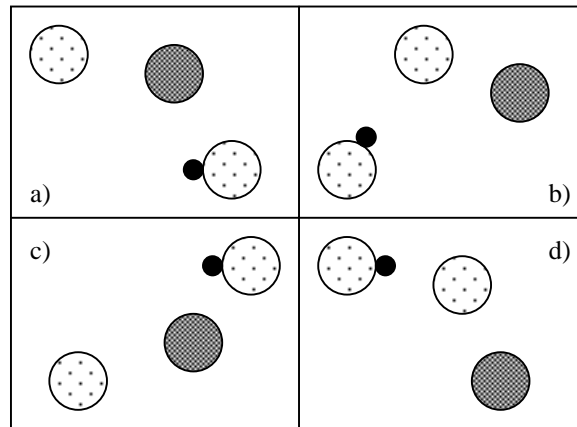


Figure 22. Footballer not between team-mate and opponent

In the cases in figure 22.a) and 22.c), they are not safe passes because the opponent can block or retrieve the ball if the footballer makes a pass to the team-mate. Cases in figure 22.b) and 22.d) are safe passes.

When the footballer passes the ball to the team-mate, it needs to turn to the closest direction facing the team-mate, in order to pass the ball as effectively and as fast as possible.

When the closest team-mate cannot get a safe pass, the footballer finds the next closest team-mate it could sense, and so forth. In case the footballer cannot determine any team-mate that it could pass to, it will refuse such an action.

*Passing the ball* only occurs if a safe pass exists. One of the features in *blocking opponent* enables the agent to get the ball only if the opponent's ball passing action is not safe. Non-safe passing the ball does not occur in the agents using the design in this project, making the designs of *passing the ball* and *block opponent* contradictory.

However, *block opponent* enables the agent to get either closer to the ball or one of the opponents having the ball. This enables the agent to be closer to the ball, increasing the opportunity for the agent to get the ball.

If agents of the current design are played against different types and designs of agents (as scope for further work), the agents with the design of this project might be more advantageous against others. This phenomenon may occur if *block opponent* and/or *passing the ball* is/are better in terms of performance than the opposing agents, thus play better and leading to victory in 5-side football simulation.

## Without motivators

Without the motivators to perform these 3 tasks mentioned in the previous subsections, the agent is capable of performing actions such as *dribbling*, *block opponent* or *passing the ball*.

On the other hand, the footballer would not be able to determine which is the best method and opponent to dribble away from or block, and could not determine which is the most suitable team-mate to pass the ball to.

Motivators enable these agents to choose which action is the most suitable to perform to, and perform such action in the most appropriate manner.

### 3.6.11 Priority

Several tasks may need to be performed by the agent at a particular time. Some of these tasks may be contradicting to each other or may not be perform simultaneously. These tasks are categorised in different priority weights, thus enabling the agent to choose which one to perform to.

Agent performs the task with the highest priority, meaning the corresponding agent has chosen the most suitable task to perform at a particular time in the environment.

Due to the constraints of time in this research course, the values used as weights for the agent's priorities are pre-defined by the programmer.

Regarding to this simulation, priorities are categorised in the following order:

#### Footballer moving

Move to ball	0.8
Block	0.6
Coach advice	0.4
Wander around	0.2

#### Footballer having the ball

Shoot the ball	0.9
Dribbling the ball	0.7
Passing the ball (footballer between team-mate and opponent)	0.6
Dribbling away from the closest opponent	0.5
Passing the ball (team-mate closer than the opponent)	0.4

#### Information that causes to lower the priority using communication

Not getting the ball passed by the team-mate	0.5
Not go to the ball	0.1

With the list of values for the priority of each of the task the agent can perform, the agent would give preference to the task with the highest priority.

The value for the footballer's priority varies between 0 and 1, 0 being the lowest priority and 1 as the highest priority.

The first set of priorities corresponds to tasks for the footballer when it has not got the ball. The footballer would give the highest priority to get the ball. Wandering around has the lowest priority; it only does this task when all other conditions are not satisfied. Coach's advice changes the priority of the footballer, but if there are other conditions that are satisfied with higher priorities, the footballer will ignore the coach's advice.

The second set of priorities corresponds to tasks the footballer can do when having the ball. The footballer has the highest priority when it senses the opponent's goal, making it to shoot the ball at that goal. Dribbling and passing the ball are split into two different types of task. The condition for dribbling with the ball is satisfied when the closest opponent is far from the footballer, whereas dribbling away from the opponent is when they are close to each other. Passing the ball if footballer is between team-mate and opponent is a safer pass, which corresponds to the situations shown in figure 21. Passing the ball if team-mate is closer than the opponent, such as examples from figure 22, is not as favourable as the examples from figure 21.

With the usage of communication, priorities for certain tasks would be decreased, such as those from the third set of priorities. If one of the team-mates is trying to get the ball, the footballer would lower its priority of getting the ball. This enables the footballer to perform another task with a higher priority. Additionally, if one of the team-mates is intending to pass the ball to another team-mate, the footballer would not intercept the ball and let the corresponding team-mate to have the ball.

Values of these priorities may use another unit or ratio, but the order for the tasks would be the same as those chosen by the programmer.

### 3.6.12 Urgency

Urgency is one of the attributes of motivators used by Sloman[1987;1994]. Urgency is pursued research separately from other attributes of motivators to study the merits of using such concept for agents' behaviour in 5-aside football simulation scenario.

Urgency defines as how fast the agent needs to react towards certain events in the current situation.

For example, if it is urgent for the footballer to get the ball, it would speed up to get the ball.

For this simulation, the agent evaluates its urgency towards the current situation. The urgency of such an agent enables it to control its speed of running or kicking the ball more appropriately, and managing its energy more efficiently.

For example, the defender is tired and standing in its field. A striker from the opponent's team is charging fast with the ball towards the defender's side of the football field. In such case, despite the defender being tired, it is urgent to get the ball to avoid the opponent to have the opportunity to score. Therefore, the defender will accelerate towards the opponent to get the ball despite the defender' exhaustion.

The footballers are evaluating their urgency depending on the current position of the ball in the field. Figure 23 labels different areas of the football field, and table 5 illustrates the urgency values for different footballers' roles in each team and areas of the field. Value 0 corresponds to not urgent for the agent, 1 for urgent and 2 for very urgent to get the ball. Area a) is the penalty area for team 1, b) is the half-field for team 1, c) is the half-field for team 2, and d) is the penalty area for team 2.

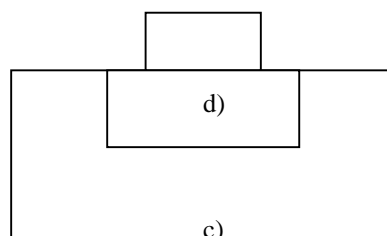


Figure 23. Different areas of the football field

Area	Team 1			Team 2		
	Goalkeeper	Defender	Striker	Goalkeeper	Defender	Striker
a)	2	2	2	0	0	2
b)	1	1	1	0	0	1
c)	0	0	1	1	1	1
d)	0	0	2	2	2	2

Table 5. Urgency value for footballers in each football field area

Table 6 illustrates the energy level at which the agent would rest (not perform any other task) depending on its urgency value.

Urgency value	Energy threshold
0	500
1	250
2	100

Table 6. Energy threshold for the agent to rest depending to its urgency

These values on table 6 only apply to outfielders, the goalkeeper will perform its task such as saving the goal. A goalkeeper rests in when it is inside its penalty area and the ball not nearby.

In the case where the agent does not have the capability to have urgency, it would rest only if its energy level went below 100.

The footballer will react depending on its urgency, which enables the agent to react more efficiently and appropriately to enhance the team's overall performance.

### 3.6.13 Communication

The footballers of the same team can communicate to each other using the blackboard architecture as shown on figure 24.

<p><u>Message board</u></p> <p>Player 1 wants to pass ball to player 4</p>
--

Figure 24. Communication for agents using blackboard architecture

When a footballer wants to communicate with its team-mates, it leaves a message to the message board. Other team-mates check the message board, and the corresponding message receiver will evaluate the message.

Whether the message receiver would follow the instruction made by the message sender is entirely dependent on the message receiver. The receiver can ignore the message if it is not intended to do such a task or a task with higher priority exists.

Communication between the agents can be established when the following circumstances arises:

1. Mark an opponent to block
2. Pass the ball to the team-mate
3. Getting the ball

The purpose of having communication between the footballers of the same team is to facilitate the decision of choosing which task to perform. Instead of having several footballers aiming to block the same opponent or getting the ball, a dedicated footballer can do that task leaving others to perform other tasks.

Communication allows the footballers to have *teamwork*, making the tasks more efficient and effective for the overall team.

### **Mark the opponent to block**

When a footballer marks a particular opponent to block, it will inform the team-mates which opponent is chosen by this footballer.

This message will enable the footballers to know which opponent is marked by a team-mate. This message helps the footballer to decide to block another opponent instead of several footballers of the same team blocking the same opponent.

### **Pass the ball to the team-mate**

When a footballer wants to pass a ball to a particular team-mate, others might try to get the ball instead of the corresponding footballer.

By letting the team-mates know which corresponding footballer is supposed to get the ball from the pass, others would not fight for the ball.

### **Getting the ball**

There could be circumstances that footballers of the same team fight for the ball. If one of the footballers of the team lets the others know it is going to get the ball, other team-mates might be able to perform another task instead of team-mates fighting for the ball amongst each other.

The message board for each team is only visible among the team's footballers. These messages are refresh every time a footballer gets a ball, because new strategies and tasks need to be made by each footballer when this situation arises.

### 3.6.14 Coach

The coach in this football simulation aids planning the strategy for the whole team.

The coach of each team can allocate the position of all footballers and the ball.

The coach, in addition to monitor the footballers of its team, will give advice to these footballers.

The team's coach can give two types of advice to the footballer as follows:

1. Closest footballer goes to the ball
2. Other team-mates, apart from the goalkeeper, follow the footballer that is going to get the ball

First, the coach has to determine which footballer is closest to the ball, then instruct the corresponding footballer to go after the ball.

As for other footballers, besides the goalkeeper, it follows the lead of this footballer for assistance. If the team-mates follow the footballer that is going for the ball, it could enable passes of the ball as a beneficial tactic for the team.

The method of instructing the other team-mates to go after the lead is as follows (footballer as the agent that is going after the ball):

1. IF defender THEN follow the lead of the footballer until it reaches the half-field  
ELSEIF striker, follow the footballer
2. IF more opponents are on the left side THEN go to the right side  
ELSEIF more opponents are on the right side THEN go to the left side  
ELSE go straight ahead

The reason of such determination on which side the footballer goes to is to move to a more advantageous position to help the footballer, meaning a better position when getting a ball from the footballer's pass.

The technique of communicating between coach and footballers of the team is by blackboard architecture, as shown in figure 25.

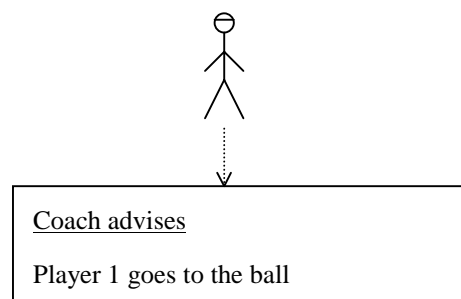


Figure 25. Advises from the coach

As the same architecture for communication between team-mates, the board from the coach advice is exclusive for the footballers in the same team. Messages will be erased when a footballer possesses the ball for the same reason as communication between footballers.

As mentioned in section 3.6.11, the footballer does not need to obey the instruction from the coach when there is a higher priority towards another task.

### 3.7 Design of the simulation software of 5-aside football

Sections 3.1 to 3.6 covered the aspects in designing an agent in 5-aside football scenario.

This chapter covers aspects regarding to software design, graphical representation and software testing for this research.

The simulation of 5-aside football is implemented in Pop-11 using Sim\_Agent Toolkit.

Pop-11 contains features such as logic programming language, that is, the syntactical structure of the language is efficient, in terms of the amount of code and execution time, for programming Artificial Intelligence software. This programming language also contains features in Object Oriented Programming. This enables the software to be implemented into modules, so that it eases the upgrading procedures and amendments for the code.

Some of the designs mentioned in the previous sections in this chapter for the agents are integrated in these logic based system format, which facilitates when adding or removing rules when needed.

Sim\_Agent Toolkit is a software library for Pop-11. This toolkit provides features to create agents in a simulated environment.

The combination of Pop-11 and Sim\_Agent Toolkit enables to contain the following necessary features to investigate this research:

- Rule-based system
- Sensing agents and objects in the environment
- Integrating motivational abilities to all agents
- Evaluate results generated by agents' behaviour

When these features are implemented, evaluation and testing are made for the investigation of the agents' behaviour, the overall outcome and the co-operation between agents.

The graphical representation of the football field uses the same format as in RoboCup Small-Size League for robots in 5-aside football. Figure 26 illustrates the layout of the football field for the simulation of these agents (not to scale):

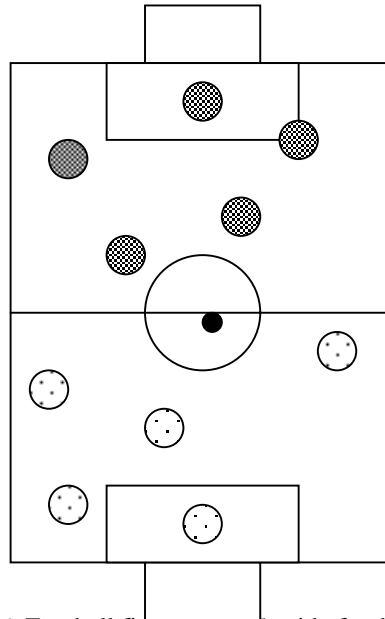


Figure 26. Football field for the 5-aside football

The agents are labelled in different colours depending on which team it belongs to, and the ball is the small black circle in figure 26.

The displacement of the footballers and the ball on the graphical interface are described as follows in terms of pixels of the interface:

- Agent walking – plus 1
- Agent running – plus 2 to 3 depending on speed
- Ball after shot or passed – plus 1 to 5 depending on speed

A separate window illustrated in figure 27 shows the current result in the football match and the percentage of the ball being controlled by footballers of each team.

Team 1 - Team 2	
Score	1 - 0
Having	49% - 51%
	the ball

Figure 27. Scoreboard of the 5-aside football game

In terms of a football game, the team with the highest score wins. However, if the game is draw, the team with the highest percentage of possession of the ball is considered to be better, because they have more opportunity to get the ball. This is the standard use in this project to evaluate which team is playing better.

When a team score a goal, the game would be reset by having the ball in the middle and all footballers on their side of the field, and the strikers of the team which has just lost a point will be closer to the ball.

Another window monitors the actions taken from the footballers in the game, as shown in figure 28.

Number of cycles: 290																
Agent	Team	Mode	Energy	Urg	Getball	Block	Shoot	Pass	Mvaway	Tkad	Ptug	PtCch	PtComm	PotDrv	Motv	Time
1	1	0	1000	0	3	0	0	0	1	0	YES	YES	YES	YES	NO	300ms
2	1	1	997	0	13	4	0	0	2	0	YES	YES	YES	YES	NO	300ms
3	1	1	913	0	5	10	0	0	5	5	YES	YES	YES	YES	NO	300ms
4	1	2	857	1	8	0	0	1	1	2	YES	YES	YES	YES	NO	300ms

Figure 28. Footballer's status

Please note that PotDrv in figure 28 and 33 corresponds to *priority* as opposed to *drive*. The concept of drive has been pursued research on this project, but further research needs to be done and the concept of *drive* was removed to be use in this project.

Each column corresponds to the counter value for each of the footballers. For example, agent 2 of team 1 has the following in cycle 290:

- In mode 1, the footballer is a defender
- Energy level is 997
- 0 for urgency, not urgent
- 13 times of going after the ball
- 4 times of blocking an opponent
- 0 time shooting the ball to the goal
- 0 time passing the ball to a team-mate
- 2 times dribbling away from an opponent
- Took 0 time advice from the coach
- The agent possess urgency, can take advises from the coach, can communicate with team-mates, a potential for priorities
- The agent does not have motivation
- It takes 300 milliseconds for the agent to perform a task between the previous and the current cycle

These values from each agent are used for the evaluation of the agent's behaviour and performance for the simulation of 5-aside football.

In 5-aside football, the footballers and the ball cannot go beyond the field. For such rule in the game, there is no throw-in rule applied in 11-league football game in this simulation.

In case of the ball hitting the border of the field, the ball will bounce back to the field in the opposite direction, such as the case in figure 29.

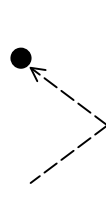


Figure 29. Ball bouncing from the edge of the football field

When a fast travelling ball is going towards the footballer, the footballer may not be able to trap the ball. Figure 30 shows the direction where the ball is initially from and goes to after bouncing off from the footballer.

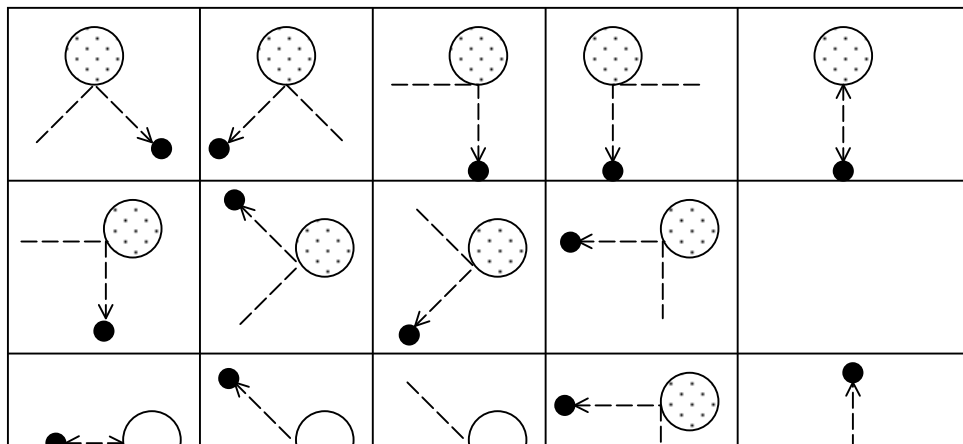


Figure 30. Ball bouncing off from the footballer

Figure 30 covers all possible combinations where the ball bounces off the footballer in different incoming directions. This phenomenon makes the footballer unable to trap the ball unless it is travelling slowly.

## 4 Implementation and testing for the simulation of 5-aside football

### 4.1 Introduction

Chapter 3 has covered the design of the agents, the programming language for the simulation and graphical representation of the agents in the 5-aside football scenario.

This chapter covers the implementation of these agents in the simulation, which includes the testing of the program and aspects regarding to the execution of the program. The technique on how to test different types of agents will be covered in section 4.6. Problems encountered during implementation are covered in section 4.7.

### 4.2 Implementation of the 5-aside football simulation

The simulation program is designed to be executed in Sun Workstation Ultra 5 system, equipment provided in the NEAT (Neural, Emergent and Agent Technologies) Lab from this department, using Solaris Windows operating system.

The specification of the Sun Workstation Ultra 5 are as follows:

CPU	: UltraSPARC-IIi 360MHz
System clock frequency	: 90 MHz
Memory size	: 128 Megabytes

The program is written in the Pop-11 programming language, with the addition of using Sim\_Agent Toolkit for the agent's functions and attributes used in this simulation.

The pseudo code in Appendix C illustrates the program written for the motivational agents in the 5-aside footballer scenario.

The declarations, methods and functions in the pseudo code on Appendix C are identical to those mentioned in the Design Chapter 3.

Each simulation cycle involves executing the necessary functions of the programs and rules for the agent.

The program is stored in the shared drive from the Computer Science Department system in this University. For such reason, the program can be executed on any Sun Workstation or PC (Personal Computer) using Exceed connected in the department's network.

When the program is executed in a Sun Workstation, its CPU will be used for display and other resources for the computation used in the program. On the other hand, if the program is executed in the PC using Exceed, the computation and processing of the program is made by the main server of the department (Scarlet), and the PC's CPU is solely used for interface display of the program.

During implementation of the 5-aside football simulation, the program was first implemented and tested with simple methods. After approval from the test, more methods were added until all methods are built and tested for the final version of the simulation program.

The programs implemented until all methods were completed are stored in the directory 'Programs' in the CD-ROM provided. Files are numbered after the name 'soccer',

number 1 being the first program built and the highest number to be the last program. The file 'soccer.p' is the final version made for this simulation.

To execute the 5-aside football simulation, first switch on Pop-11 in the Solaris system or Exceed, then load the program 'soccer.p'.

There are different ways of executing the program. If the program is executed for the first time, it can be executed by typing 'play(*n*)', where *n* is equivalent to the number of cycles to execute the program. The program can be resumed after executing 'play(*n*)' by typing 'go(*n*)' for the *n* number of times to continue the execution.

Please note that the word *drive* used in the syntax and representation in the program corresponds to *priority*.

The program is set by default for all agents to have priorities, motivation, communication within agents of the same team, urgency and coach. In circumstances for agents with or without some of these functionality, it needs to execute the program by using 'vs'. The syntax of this type is 'vs(*n, drive1, drive2, coach1, coach2, comm1, comm2, urgency1, urgency2, motivation1, motivation2*)'

For example, in case of the agents of the two teams with these functionality:

Team 1 – no priority to be considered in the tasks, with urgency, no communication, with motivation, with coach

Team 2 – has priority weights within tasks, no urgency, no communication, with motivation, without coach

Execute 10000 cycles for the simulation

The syntax for such agents in the 5-aside football simulation is:

'vs(10000,false,true,true,false,false,false,true,false,true,true)'

This type of executing the 5-aside football simulation enables to agents in the team to have different functionality.

### 4.3 Interface of the 5-aside football simulation program

When the 5-aside football simulation program is executed from Pop-11, three windows will appear. The first window displays the football game, as shown on figure 31. The agents in red are from team 1, agents in blue from team 2, and the ball as white circle. The upper half of the field belongs to team 2, and the lower half of the field belongs to team 1. All agents are labelled with their corresponding number in the team. The second window displays the score and the percentage of footballers which have touched the ball in both teams during the game, as in figure 32. The third window displays the status of all agents during each cycle of the game, as in figure 33.



Figure 31. 5-aside football game window

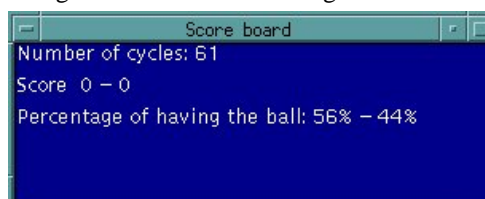


Figure 32. Score board in the simulation game

Agent board																
Number of cycles: 61																
Agent	Team	Mode	Energy	Urg	Getball	Block	Shoot	Pass	Mvaway	Tkad	Pturg	PtCch	PtComm	PotDrv	Motv	Time
1	1	0	954	1	21	0	0	1	0	1	Yes	Yes	Yes	Yes	Yes	3300
2	1	1	877	1	30	6	0	2	1	22	Yes	Yes	Yes	Yes	Yes	3400
3	1	1	877	1	30	6	0	3	0	22	Yes	Yes	Yes	Yes	Yes	3400
4	1	2	935	0	37	6	0	3	1	14	Yes	Yes	Yes	Yes	Yes	3400
5	1	2	937	0	41	6	0	2	0	12	Yes	Yes	Yes	Yes	Yes	3400
1	2	0	1000	0	0	0	0	0	0	0	Yes	Yes	Yes	Yes	Yes	3400
2	2	1	939	0	0	7	0	0	0	54	Yes	Yes	Yes	Yes	Yes	3400
3	2	1	939	0	0	7	0	0	0	54	Yes	Yes	Yes	Yes	Yes	3400
4	2	2	876	1	39	9	0	0	4	9	Yes	Yes	Yes	Yes	Yes	3400
5	2	2	874	1	36	9	0	0	6	10	Yes	Yes	Yes	Yes	Yes	3400

Figure 33. Agent board in the simulation game

The images captured in figures 31 to 33 are from Pop-11 running the 5-aside simulation program under Exceed in PC. Columns of messages such as those in figure 33 are not aligned properly, but will align if the program is executed on a Sun Workstation system.

All the interfaces illustrated in these figures are identical to the layout mentioned in the design chapter 3.

The values illustrated in figures 32 and 33 enables to analyse the status of all agents and the progress of the team during the game. In case of these figures, it represents the status of the agents and the team in cycle 61 of the 5-aside football game.

#### 4.4 Testing the program of 5-aside football

During implementation of the 5-aside football simulation, tests were carried out to ensure the simulation meets the requirements and performance in this project.

The program was first built with simple functions introduced in the agents' architecture. After testing, additional functions were incremented to agents until they reached the full architecture from the designs of the agent architecture.

During testing of the simulation, one incident had appeared several times. On these occasions, a group of agents are fighting for the ball near the edge of the football field. Due to the rules of the 5-aside football, the ball cannot go beyond the field, making the ball bouncing back to the field. An agent may be trying to kick the ball to avoid being taken by an opponent, but the ball goes back to the same agent and it kicks the ball in the same direction again. This incident makes the ball and some agents become stuck in the same positions, such as the example shown on figure 34. The ball keeps going back to the agent after kicking.

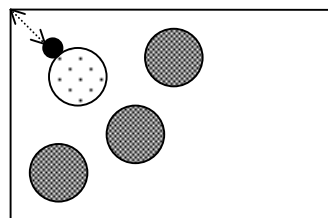


Figure 34. Agents and the ball getting stuck in the same position

For such incident, a method in the program is added to check the position of the ball. If the ball remains the same position for 50 cycles during the game, the ball will be place in the centre of the football field. The reason of positioning the ball back to the centre of the field is for the agents to carry on the game, without been unfair to place the ball in either side of the football field.

The 5-aside simulation program is tested running on three different types of execution, they are:

1. Exceed on a PC
2. Sun Workstation with Windows Solaris
3. Sun Workstation without Solaris or any other Operating System

The speed of execution of the program varies between these different types of system.

Executing the program in Exceed is dependent on how many users are logged to the Scarlet System in this department, the share server.

The speed of the program executing under Windows Solaris is dependent on how many processes are executed in the Sun Workstation used. Even without any programs except Pop-11 running, the system uses substantial amount of system memory executing the interface, UNIX and Solaris Windows on the background for the display and system's operation.

The speed in executing the simulation program without Solaris is the fastest among all. This is achieved by remote log-in to a Sun Workstation which is not in use. In this test, the Sun Workstation is not logged-into by any user and the system is in energy saving mode (processing power running in minimal). Using another Sun Workstation, remote log-in to this unused Sun Workstation and execute the 5-aside football simulation program. This unused system uses the system memory solely when executing the simulation program, enabling it to execute as fast as possible and sending the data back to the other Sun Workstation for displaying the program.

Programs in the 5-aside football have been executed in either of these types of system. Further details in performance of speed in these are discussed in section 4.7.

#### 4.5 Enhancements made during tests in the simulation program

Tests were made by executing 10000 cycles in the 5-aside football simulation programs. For Sun Workstation running on Solaris Windows, the program takes approximately 2 hours to execute 10000 cycles. For the same computer without Solaris Windows, it would take approximately 30 minutes to execute the same number of simulation cycles. Using Exceed on a Personal Computer, it takes approximately 1 hour to execute the same amount of simulation cycles for the program.

The results show that with the speed of agents and ball displacement designed, the football game would make no score for both teams. The tests were made with both teams having priorities, motivators, urgencies, coach and communication among team-mates.

The rate of agents and ball displacement speed were increased correspondingly to investigate which speed would be appropriate for testing the agents' performance towards different combination of agents' functionality, which are explained in details in section 4.6.

Table 7 shows the average result from football matches. The ball used in these matches does not bounce off the footballer in high speed. These results include the score and the percentage of footballers touching the ball in the game. Each average result is taken from five football games. The results of each of these games are shown on table 1 in Appendix A.

Speed * 1	Speed * 2	Speed * 3	Speed * 4	Speed * 5
1-0 48-52	1-2 48-52	1-4 50-50	1-4 40-60	2-8 50-50

Table 7. Results from different speeds in the football game (ball not bouncing off the footballer)

Table 8 shows the average result from football matches with the feature of having the ball bouncing off the footballer if its displacement is rapid. The results of each of these games are shown on table 2 in Appendix A.

Speed * 1	Speed * 2	Speed * 3	Speed * 4	Speed * 5
0-0 49-51	1-1 47-53	2-3 51-49	2-6 48-52	3-5 48-52

Table 8. Results from different speeds in the football game (ball bouncing off the footballer)

In practise, two teams of agents with the same combination of functionality may not be causing the football game to draw. This statement is true because footballers do take opportunities to get the ball or score when such situation arises. The footballers have minimum capability to play football and score, disregarding the presence of functionality such as motivators and urgencies in the agent. Therefore, even the agents in both teams with the same capabilities, one might win due to opportunities taken by the team. However, for evaluation's purpose, results of agents with the same combination of functionality are more favourable if the final score is draw, but not having the final score as 0-0 (too low for a football game score). This concept facilitates the evaluation for agents with different functionality in each team. One team with the corresponding combination of functionality is considered to be better than another team if that team has won the game or having higher percentage in possessing the ball if the final score of the game is draw. To be able to achieve such method for evaluation in this project, the simulation needs to be able to generate draw game and approximate percentage of possessing the ball when agents have exactly the same combination of functionality in both teams.

Judging from the results from tables 7 and 8, the most ideal speed for the agents would be speed times 2 with the ball bouncing off the footballer. However, without the consideration of the average result, using data from table 2 in the Appendix A show team 1 has more advantage towards team 2, that is, team 1 wins more often than team 2. Therefore this ratio and feature for the agents is not suitable for evaluation purpose.

The second result considered was speed times 3 with the ball bouncing off the footballer. The results from table 2 in Appendix A also show the tendency of one of the teams to win more frequently than the other team. In this case, team 2 has defeated team 1 4 out of 5 times. This ratio is not considered for such a reason.

The next consideration was speed times 2 without the ball bouncing off the footballer. Despite the average result from table 7 showing these teams do not have drawn games, its results from table 1 in Appendix A shows 3 out of 5 games are draw.

Using the ratio of speed in ball and agents displacement times 2 was chosen to be used throughout all the other games for testing and evaluation of different combination of agents' functionality. For such reason, the feature of having the ball bouncing off the footballer was taken out of the 5-aside football simulation program.

## 4.6 Testing different combination of agents' functionality

Agents in each team have basic capabilities that enable these footballers to play 5-aside football. Each team may or may not have some of the functionality to perform particular tasks. This project explores such concept to enable testing agents with different

combination of functionality. The following are the combination for testing the agents of the team with different functionality:

- Agents with co-operation VS Agents without co-operation
- Agents with motivational aspects VS Agents without motivational aspects
- Agents with urgency VS Agents without urgency
- Agents with priorities VS Agents without priorities
- Team with Manager VS Team without Manager

Agents are labelled in the following types:

- A0 – Agent with basic capabilities to play 5-aside football
- A1 – Agent as A0 with addition of motivators
- A2 – Agent as A1 with addition of priorities
- A3 – Agent as A2 with addition of urgency
- A4 – Agent as A3 with addition of advice taken from their coach

In addition to these functionality featured in the agents, agents may or may not contain the functionality to communicate between team-mates.

Table 9 shows all the combinations to test the different agents' functionality in the 5-aside football scenario.

A0+ vs A0-	A0+ vs A1+	A0+ vs A2+	A0+ vs A3+	A0+ vs A4+
	A0- vs A1+	A0- vs A2+	A0- vs A3+	A0- vs A4+
	A0+ vs A1-	A0+ vs A2-	A0+ vs A3-	A0+ vs A4-
	A0- vs A1-	A0- vs A2-	A0- vs A3-	A0- vs A4-
	A1+ vs A1-	A1+ vs A2+	A1+ vs A3+	A1+ vs A4+
		A1- vs A2+	A1- vs A3+	A1- vs A4+
		A1+ vs A2-	A1+ vs A3-	A1+ vs A4-
		A1- vs A2-	A1- vs A3-	A1- vs A4-
		A2+ vs A2-	A2+ vs A3+	A2+ vs A4+
			A2- vs A3+	A2- vs A4+
			A2+ vs A3-	A2+ vs A4-
			A2- vs A3-	A2- vs A4-
			A3+ vs A3-	A3+ vs A4+
				A3- vs A4+
				A3+ vs A4-
				A3- vs A4-

				A4+ vs A4-
--	--	--	--	------------

Table 9. Different types of agents against each other in the 5-aside football simulation

These combinations in table 9 covers the teams with one type of functionality against all other types of agents. Cases such as 'A2 vs A1' will not be tested after having the combination 'A1 vs A2', because these combinations are using identical types of agents.

The sign plus ('+') and minus ('-') signifies the existence of communication between agents of the same team. For example, 'A3- vs A4+' means the agents in the first team have motivation, priorities and urgencies, whereas agents in the second team have motivation, communication, priorities, urgencies, takes advice from coach and communication. In case of executing the 5-aside football simulation program with 'A3- vs A4+', the syntax in the Pop-11 program would be 'vs(*n*, true, true, false, true, false, true, true, true, true, true)', where *n* is the number of cycles to be executed in the program.

#### 4.7 Problems encountered in the tests of the simulation program

In some circumstances of the simulation of the 5-aside football game, the agents get stuck in the game. In this case, agents are not able to determine which way to go. That situation arises when the ball is allocated back to the centre of the field, leaving all agents in the program unable to sense the position of the ball.

The agents have limited sense range in the football field: in terms of the pixels of the program, this is a radius of 200 from the centre of the agent. The football field is 382 pixels wide and 785 pixels long. Therefore an agent can sense approximately one quarter of the football field size.

In cases where the agent can listen to the advice from team's coach (another agent), some of the agents are able to allocate the position of the ball, and this prevents having the agents stuck in the game.

To enable the evaluation of the 5-aside football simulation remains accurate, the games where the football agents are stuck would be discarded. Every set of the different types of agents contains 3 programs with results that do not contain agents been stuck at any occasion in the game.

One measurement integrated in the 5-aside simulation football was to benchmark the program, measuring the time taken from one simulation cycle to another of the agent performing its task.

As mentioned in section 4.4, there are three methods used to execute the simulation program.

Executing the program using Exceed from Windows NT in the PC varies its speed depending on how many users are logged to the server. This proves to be fruitless to benchmark the program, because the results in their execution are at random speed.

In a later stage during data gathering, programs were not executed using Exceed on the PC, because it is time-consuming on executing these programs.

Executing the program on a Sun Workstation with or without Windows Solaris performs a stable (less variant) speed, because its speed is only affected by the memory taken in the applications executed in the computer. Unfortunately, during the data gathering from these programs, the Department had switch off all system for a power

check (on the 25<sup>th</sup> of August 2001), and the speed of executing these programs after this incident had accelerated.

The execution time for these programs after the power check had approximately decrease to half.

Before the power check, programs executed on the Sun Workstation under Windows Solaris took approximately 2 hours, whereas a Sun Workstation without Windows Solaris took approximately 30 minutes.

If the assumption is made, where the speed of the program execution is half that of the programs before the power check, then different combination of agents' functionality had taken approximately the same amount of time to execute the simulation program. However, this assumption cannot be validated. Benchmarking the program can only be valid if the hardware and software had remained at the same performance throughout data gathering.

The time taken to execute all the programs for data evaluation in this research had taken 20 days running these programs for approximately 10 hours a day. At the particular moment when the power check was held in this department, half of the data has been gathered from executing the simulation programs.

Due to the constraints of time for the research course, these programs cannot be disposed of and the programs executed again for benchmarking purposes.

The concept of measuring the benchmark to evaluate the agents' performance is therefore abolished.

Table 3 in the Appendix A illustrates the benchmark of these programs.

The conclusion generated from benchmarking the 5-aside football simulation is that the program is executed faster when there are less application running in the system, and having the system not logged by more than one user. In case of the Sun Workstation with Solaris, it executes slower than the one without Solaris, because the operating system Solaris Windows and UNIX takes up substantial amount of computer memory.

Another problem encountered on evaluating the performance of the agents in the program is their energy level. Each agent's energy level changes depending on its performance. In the case where an agent runs and kicks the ball often, its energy level decreases more rapidly. The agent can regain energy by resting or during the reset of the game. Resetting the game happens when an agent score a goal and all footballers need to return to their side of the football field. During the reset of the game, each agent gets an additional 100 to its energy value or its energy value is set to a 1000 only if its energy value was previously above 900.

An agent uses its energy depending whether there are tasks to do. When the agent cannot detect the position of the ball, it wanders around the field, which does not take energy. Therefore, the agent's energy use is not dependent on which combination of agent's functionality it applied.

In spite of this, as the design of the agent architecture, agents with urgency are capable to managing their use of energy more appropriately. When the situation is not urgent, the agent either moves slower to conserve energy or rest. If the agent is in an urgent situation, it runs faster unless its energy level is too low.

Energy level is not directly evaluated in this research. Agent performance has indirect relationship with energy used with the conjunction of agent's urgency. Evaluation of agent's energy not in used for evaluation purposes is further explained in Chapter 7.

## 5 Evaluation of the results from the simulation of 5-aside football games

### 5.1 Introduction

In chapter 4, topics regarding the implementation and testing of the 5-aside football simulation program were discussed. During this chapter problems encountered during the testing were mentioned, justification of the reason for abolishing some of the concepts from the design were also stated.

This chapter focuses on evaluation of the results generated from the 5-aside football simulation programs. Results are compared and analysed for the evaluation and answering the research questions raised.

### 5.2 Results gathered from the 5-aside football games

The agent functionality of each team featured in the 5-aside football is as follows:

- A0 – Agent with basic capability to play 5-aside football
- A1 – Agent with basic capability plus motivators
- A2 – Agent with basic capability plus motivators and priorities
- A3 – Agent with basic capability plus motivators, priorities and urgency
- A4 – Agent with basic capability plus motivators, priorities and urgency plus advice taken from their coach

Each type of agent may or may not communicate with their team-mates. The symbolism used for agents with communicating is a plus sign '+' after the agent type, minus '-' for the agents without communication integrated.

Table 10 illustrates the results generated from the games executed in the 5-aside football simulation program. Each result is an average of score and percentage of the team having the ball during the game, for each set of agent with different type. For the values from which the average results were taken, please refer to the table 4 in the Appendix A.

A0+A0- 0-0 49-51			
A0+A1+ 0-0 46-54	A0+A1- 1-0 35-65	A0-A1+ 1-0 44-56	A0-A1- 1-0 47-53
A0+A2+ 1-0 42-58	A0+A2- 1-0 40-60	A0-A2+ 0-0 41-59	A0-A2- 1-0 42-58
A0+A3+ 0-3 22-78	A0+A3- 0-2 14-86	A0-A3+ 0-5 29-71	A0-A3- 0-4 24-76
A0+A4+ 0-5 9-91	A0+A4- 0-6 10-90	A0-A4+ 0-4 10-90	A0-A4- 0-4 11-89
A1+A1- 1-1 48-52			

A1+A2+ 0-1 60-40	A1+A2- 0-1 57-43	A1-A2+ 1-0 57-43	A1-A2- 0-0 60-40
A1+A3+ 0-6 18-82	A1+A3- 0-7 13-87	A1-A3+ 0-5 50-50	A1-A3- 0-9 13-87
A1+A4+ 0-2 8-92	A1+A4- 0-9 10-90	A1-A4+ 0-6 10-90	A1-A4- 0-4 10-90
A2+A2- 0-0 46-54			
A2+A3+ 0-5 35-65	A2+A3- 0-4 11-89	A2-A3+ 0-4 27-73	A2-A3- 0-4 34-66
A2+A4+ 0-4 10-90	A2+A4- 0-6 12-88	A2-A4+ 0-3 10-90	A2-A4- 0-4 9-91
A3+A3- 2-3 45-55			
A3+A4+ 0-1 48-52	A3+A4- 1-2 28-72	A3-A4+ 1-1 48-52	A3-A4- 2-3 48-52
A4+A4- 0-2 47-53			

Table 10. Results from the matches in different types of agents in the football team

Results from table 10 derives the following comparisons:

For agents having matches with team A0

- A1 gets the ball more frequently, but fail to lose to A0
- A2 gets the ball more frequently, but fail to lose to A0
- A3 gets the ball more frequently and wins the game with high score
- A4 gets the ball more frequently and wins the game with even higher score than A3

For agents having matches with team A1

- A2 wins the game despite getting the ball less frequent than A1
- A3 gets the ball more frequently and wins the game with high score
- A4 gets the ball more frequently and wins the game with high score

For agents having matches with team A2

- A3 gets the ball more frequently and wins the game with high score
- A4 gets the ball more frequently and wins the game with high score

For agents A3 against agents of A4

- A4 wins the game with one point ahead and has slightly more frequent having the ball

### 5.3 Evaluation of the results from data gathered in the simulation programs

With the comparisons in the previous section, plus conclusions stated in section 4.7 from the tests made in 5-aside football simulation program, each functionality for the agent architecture derives the following conclusions which are reflected from these concepts and results.

Communication does not assist the agents to get the ball more times than the opponent or win the game. However, with the absence of coaching in the opponent's team, communication enables the team to draw or lose with marginal score difference from the opponents.

Motivators enable the footballers in the team to focus on performing the most appropriate action. With motivators, it leads these footballers to get the ball more frequently as from the results shown on table 10. However, with the presence of motivators in the agents of the team, it does not necessary lead to success in winning the game.

Priorities enable the footballer to perform the task with the highest priority. It enables the footballer to choose which task to perform when there are choices of tasks to do. With the presence of priorities in the agent, it enables the team to win the game in comparison to those without considering priorities in their tasks. On the other hand, agent with priorities obtained the ball less times than those without.

Urgency enables the agent to distinguish how fast it needs to perform such task depending on the current situation. In this simulation, agents are able to preserve their energy when it is not urgent to react to the current situation, enabling themselves to have more energy when it needs to be used in urgent situations. Regarding to the results shown on table 10, the presence of urgency within the footballers of the team enables them to defeat those without urgency with very high score. Additionally, agents without urgency cannot even score one goal against a team with footballers which have urgency.

A coach aids the agents in the team for advice. With the presence of the coach help, footballers can detect the location of the ball even if it is out of their sense range. The conclusion to be derived from the results in table 10 is that the presence of a coach to help the team enables it to win the game and have the ball more frequent than the team without a coach's aid. This statement is derived from the games having teams of A3 against A4, where A4 wins the game with one score higher and having the ball 4% more than A3. Despite the possibility of agents ignoring the advice given by the coach, the frequency of agents taking this advice is high.

The agent in type A4, agents with motivators, priorities, urgencies and coach, has shown victory in these simulated football games against agents of type A0, A1, A2 and A3. With the use of communication, as mentioned above, enables the agents to get the ball more frequently.

From this conclusive results, agents of type 'A4+' produces the best performance in the simulation program built for this project, without the degradation in response time in comparison to agents of other types.

## 6 Critical appraisal of the research in motivation of agents in 5-side football simulation

### 6.1 Introduction

Chapters 3 to 5 have covered the design, implementation and testing of the 5-side football simulation. This chapter covers the knowledge and experience gained throughout the whole research project, as well as problems encountered. This chapter commences with the course timetable, and subsequently the critical appraisal of the project.

### 6.2 Course Timetable

This course for the MSc Research commenced on 29<sup>th</sup> of September 2000 and finishes on the 28<sup>th</sup> of September 2001.

Table 11 shows the tasks made by the student during this research for the motivational agents in 5-side football simulation.

<b>Date</b>	<b>Task</b>
From October to December 2000	Background Research for the project
From January to February 2001	Producing the Research Analysis Report
From March to June 2001	Design documentation and prototyping the simulation program
July 2001	Implementation and testing of the 5-side football simulation program
From beginning to middle of August 2001	Data gathering and evaluation of data retrieve from 5-side football simulation program
From middle of August to end of September 2001	Producing the Master by Research Thesis

Table 11. Tasks made during the research

### 6.3 Critical appraisal of this project

Background research was carried out at the beginning of the course.

There were problems in retrieving the published papers from the library in this University due to the lack of material available in the facility. With the use of the Internet, particular papers and certain sections of the book can be obtained electronically. The supervisor had given guidance on retrieving books, theses from researchers and other relevant papers.

During the research of the programming language Pop-11 and the Sim\_Agent toolkit, there was lack of material available, in respect to books available in library or online. The Web site devoted for Sim\_Agent and Pop-11 is available in the University of Birmingham, where this programming language and libraries software are developed by Aaron Sloman. The online materials available in this Web site are not comprehensive for beginners of such programming language, and there is a lack of explanation in details for functions provided.

The deadline of the Research Analysis report was set to be in the middle of January. Due to problems regarding the structure in the topics covered in the first analysis report, it had to be re-written and submitted in the middle of February. Due to the postponement of Research Analysis report submission, the overall project needs to be expedited to meet the final deadline of the course.

During design of the 5-aside football, the progress did go smoothly but slowly. Converting concepts in football to mathematical algorithms for programming purposes has been a very challenging task.

Implementation and testing programs in Pop-11 was complicated. Without the availability of a debugger for such programming language, the programmer must be able to trace where the problems are and how to amend them when errors arise during compilation phase.

Programs are built from using existing code for the interface and sensory functions provided by the supervisor. Functions are incremented only when all existing functions work without any logical or syntactic errors. This technique of programming lowers the possibility of having too many errors at one time and eases the difficulty in correcting these errors.

Values such as variables for sensing range for the agents, distance chosen for agents to reach in certain situation, priorities and motivators are set after several tests. The values used in the final version are chosen after these tests for the agents were performed, by the programmers' decision. These values enable the agents to react and behave as footballers are in a 5-aside football scenario, and to score in the game as much as possible for evaluation purpose.

Logical errors are seldom encountered during the testing phase of the program, due to the quality of design.

There were problems during evaluation and testing of the program, as referred in sections 4.4 and 4.7. Some problems were with regard to the nature of the topic tackled in this research, others were related to the hardware. These problems were unexpected at first, and it was a useful experience to understand how such problem could arise and how it should be confronted.

During the project, parts of the phases mentioned in Table 11 are re-visited at another period of time. With the nature of research, the student must be able to adjust and use the time sensibly to meet the deadlines and keep progressing in the right pace with good quality of work produced.

It is beneficial to upgrade personal knowledge and experience on practising Artificial Intelligence techniques, as well as research techniques implemented and to develop report writing techniques.

## 7 Conclusion

This research shows the importance and benefit of integrating motivation into multiple agent architecture.

With the use of CAM (Computational Agent with Motivation) architecture and 5-aside football simulation as the vehicle of agent environment, research of their behaviour were carried out.

During the research, concepts such as motivation, priorities, urgencies, goals, communication were used for the agent.

The coach was added to research into the effect produced by multiple agents' scenarios.

It is concluded that motivation affects the agents in the football scenario to perform tasks in focussing the right action or sequences of actions. Additionally, these motivators enable the agents to get the ball more times than those without.

Priority enable agents to distinguish which short-lived action or actions favour the current situation facing the agent. This enables the agent to choose which action to perform in a certain state when there is a variety of possibilities the agent can choose from. In the 5-aside football scenario, agents that distinguish priorities in their tasks lead to victory when confronting those without.

Urgency enables the agent to react in appropriate speed and time towards the current situation. In this research, it is proven that urgency are not only enabling agents to react fast in urgent situations, leading them to win the game, but also managing their use of energy more appropriately.

Throughout the research, it is concluded that a goal does not necessary have to be explicitly specified for the agents. In these agents' scenarios, none of the agents have an explicit goal to pursue. The ultimate goal for the overall team is to win the football game. Each agent has its role to perform: strikers to attack and score, defenders to get the ball from opponents and pass to the strikers, and goalkeepers to defend their goal to avoid the opponent scoring. By having each agent perform tasks according to their roles, the ultimate goal is implicitly achieved.

Communication was added to the multiple agents' scenario. Communication is only permitted between agents of the same team, and therefore agents from the opposite team cannot retrieve conversations made by the other team. Communication does not lead the team to win or get the ball more often in the 5-aside football as expected, but it keeps the team from losing from those without communication, and not repeating the same task as another team-mate is performing in a particular time.

Communication is achieved using Blackboard architecture, where all messages are placed on the blackboard, and all agents of the same team can retrieve these messages.

The agent having the role of team's coach is proven to be beneficial. The coach aids the agents to allocate the position of the ball in this simulation. It is noted that there are occasions when all agents of the team cannot sense the ball due to its sense range, and the coach's help will eliminate the possibility of agents not being able to carry on the football game. Although the coach can only apply simple tactics in this research, it has revealed great potential of such concept for the football team to win the game. Agents may ignore the advice given by the coach if higher priorities of tasks exist, but these

agents have been taking this advice in high rates of occurrence. Therefore multiple agents are better in performance in comparison to individual agents in a team scenario.

The method of communication between footballer and the coach agent is using blackboard architecture.

The blackboard architecture is concluded to be easy to use, yet efficient to be used as means of communication between agents.

The usage of communication and coach enables to construct a multiple agent system among agents of the same football team. This concept has shown the merits of such system when confronting to individual agents in the opponent team. Team with multiple agents system has defeated the opponents in all the games shown from the results of the simulated football games. That signifies the superiority of multiple agents against individual agents.

With the hybrid use of classic AI for the rules and nouvelle AI for the behaviour and object-oriented functions of the agents, the agents has achieved high performance, in terms of execution speed and the game in 5-aside football scenario.

Before implementation and testing of the 5-aside football simulation program, it was expected that the execution of the program would be slow. It is realise that despite the presence of all the features, functions and capabilities integrated in the agent architecture, the program runs very smoothly (as displayed on the screen of the computer) in the Sun Workstation. It is not needed to eliminate any functionality from the agent's architecture. Therefore additional concepts can be added and yet still achieve fast performance in the same computer specifications used in this research.

By having good design in the agent architecture, its functions, behaviours and rules applied to the environment, its execution is relatively fast by converting biological actions such as a real footballer into mathematical algorithms.

Benchmarking the time taken for the agents in program execution and energy level are not appropriate in this research. Depending on the situation of the football game, the speed and energy level of the agent varies. Agent having a high level of energy and low speed in execution may not signify good performance; it may represent an agent not doing much work in the simulation scenario.

Without the use of memory in the agent, it restricts the ability to research on other aspects related to motivation. Due to the constraints of time in the course, memory was not introduced in this research.

The best performance agent type 'A4+', agents with motivation, priorities, urgencies, communication and coach, shows victorious results when playing against opponents of other types. The programs executed in the football simulation game in this project shows no significance amount of difference between executing a football team 'A4+' and football team 'A0-' (football agents without communication, priority, urgency, motivation or coach). Therefore, not technological trade-off was employed to execute the agent architecture with the highest amount of functionality to be running the simulation program in fast speed. For such reason, more functions may be added to the agent architecture for further research purpose, without degrading the execution speed of the simulation program from this project.

To summarise this project, using motivation, priorities, and urgency is essential for high performance in agents' scenario. With the addition of coach and communication, agents' performances are enhanced.

Artificial Intelligence is very important in computing field to help humans in performing task, either partially or fully substituting man-labour, or studying humans' behaviour for analytical purposes.

This research shows that the current technology in computing enable such agent to be executed extremely fast and reacts in real-time for agents with complex and specific functionality.

## 8 Scope for Further Work

The research can be extended by the use of memory for the agents. Memory is achieved by saving the relevant data to the system.

With the use of memory, these concepts might be researched:

- Experience
- Belief
- Plans (one of the motivator attributes used by Sloman)
- Commitment (one of the motivator attributes used by Sloman)
- Insistence (one of the motivator attributes used by Sloman)

Experience requires prior knowledge of the agent self. The agent will evaluate its data on the memory to judge on how to react in a current situation, and learning new events when they occur.

Belief is what the agent perceives about its environment. Short-lived belief is integrated into this research, but it refreshes every simulating cycle. If memory is available to the agent, its belief can be change or interpreted differently by using its experience.

This research proves that planning is achievable without memory, but its presence would enhance and increase more features to the agents' performances. Memory changes plans for the agents, because planning with experience of the agents can spawn into vast variety of tactical planning.

Commitment to an event made by an agent requires memory to be able to recognise and persist in the event to which it has committed.

Insistence also requires memory for the agent to insist on carrying out the corresponding event. Although insistence may be achieved without memory, but with the absence of experience, it is difficult to choose the sensible event to insist upon.

All these concepts mentioned above require further background research and time to design.

The 5-aside football simulation can be upgraded with a Graphical User Interface to facilitate the execution of the program. With Graphical User Interface, it might contain buttons for switching on or off motivators, urgencies, priorities, communication and coach instead of the current program typed by the user.

Values used in variables such as sense range and distance response for the events used by the agent can be change automatically, instead of by the programmer. These programs can be executed using an Artificial Neural Network [Haykin 1994] to find the most appropriate value, in this case for the agents of the team to score the highest in the game, for each of these variables to achieve highest performance. Additionally, techniques using Genetic Programming [Koza 1993; Koza 1997] could be used to experiment with 'evolving' agents.

The concept of drive [Davis 2001b] may be further pursued research on to be able to replace or aid the concept of priorities used in this project.

Concepts used in this research such as motivation, priority and urgency may be improved by enabling the agent to self-define and self-evaluate these concepts. Such improvement may use concepts such as experience and memory, as mentioned in this Chapter.

Further concepts such as emotion may also be integrated to evaluation purposes, which requires further research.

The simulation program made for this project can be upgraded into outdoor football simulation. This is achieved by adding more football rules to satisfy the outdoor football game, and change of variables when initialising the number of footballers and the football field size. More research needs to be carried out for such an upgrade.

The 5-side football simulation program can be extended to be use with real robots. With the addition of computer vision used by the robots and cameras on top of the football field, the environment where the robots interact are mapped in the system, agents in the simulation program can send data to command these robots for the subsequent action.

# Bibliography

- [Agre 1987] Agre, P.E. and Chapman, D., Pengi: And Implementation of a Theory of Activity. In: Proceedings of the 6<sup>th</sup> National Conference on Artificial Intelligence (AAAI-1987), 1987
- [Bates 1991] Bates, J. A.B. Loyall & W.S. Reilly, Broad agents, *SIGART BULLETIN*, Vol. 2, No. 4, 1991
- [Beaudoin 1994] Beaudoin, L., *Goal Processing in Autonomous Agents*. PhD Thesis, Computer Science, University Of Birmingham, 1994
- [Binsted 1998] Binsted, K., *A Talking head architecture for entertainment and experimentation*, NEC Research Index <http://citeseer.nj.nec.com/91443.html> (Last accessed 25/3/2002), 1998
- [Boutilier 1997] Boutilier, C., Brafman, R., Geib, C., Prioritized Goal Decomposition of Markov Decision Processes: Toward a Synthesis of Classical and Decision Theoretic Planning. In: *Proceeding of the 15<sup>th</sup> International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997
- [Bratman 1987] Bratman, M., *Intentions, Plans and Practical Reason*, Harvard University Press, 1987
- [Brooks 1990] Brooks, R., Elephants Don't Play Chess. In: *Robotics and Autonomous Systems* 6, pp.3-15, 1990
- [Brooks 1998] Brooks, R.A., Breazeal, C., Irie, R., Kemp, C., Marjanovic, M., Scassellat, B., Williamson, M., Alternate Essences of Intelligence. In *AAAI-98*, 1998
- [Brustoloni 1991] Brustoloni, J., *Autonomous Agents: Characterization and Requirements*, CMU-CS-91-204, 1991
- [Canamero 1997] Cañamero, D., Modeling Motivations And Emotions as a Basis for Intelligent Behavior. In: *Proceeding of the First International Symposium on Autonomous Agents*, The ACM Press, 1997
- [Coelho 1998] Coelho, H. (Eds.), The "Semantics" of Evolution: Trajectories and Trade-offs in Design Space and Niche Space. In: *Progress in Artificial Intelligence*, Springer, pp.27-38, 1998
- [Coiera 1997] Coiera, E., Artificial Intelligence in Medicine. In: *Medical Informatics, The Internet and Telemedicine*, 1997
- [Dautenhahn 1998] Dautenhahn, K. (Eds.), "Consciousness" and Conceptual Learning In A Socially Situated Agent. In: *Human Cognition and Social Agent Technology*, 1998
- [Davis 1996] Davis, D.N., Reactive and motivational agents: towards a collective minder. In: *Intelligent Agents III - Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, J.P. Muller and M.J. Wooldridge and N.R. Jennings (Eds.), Springer-Verlag, Heidelberg, 1996
- [Davis 1998] Davis, D.N., Synthetic agents: synthetic minds? In: *IEEE Systems, Man and Cybernetics*, 1998
- [Davis 2000] Davis, D.N., Designing Agents. In : *Technical Report, NEAT-1*, Department of Computer Science, University of Hull, November 2000

- [Davis 2001a] Davis, D.N., Control States and Complete Agent Architectures. In : *International Journal of Computational Intelligence*, 17(4), 2001
- [Davis 2001b] Davis, D.N., Multiple Level Representation of Emotion in Computational Agent. In: *Emotion, Cognition & Affective Computing*, AISB2001-Agents and Cognition, 2001
- [Ferber 1999] Ferber, J., *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, ISBN 0-201-36048-9, 1999
- [Fiadeiro 1996] Fiadeiro, J.L., Schobbens, P.-Y., A Logic of BDI Agents with Procedural Knowledge. In: *Proceeding of the Second Workshop of the MODELAGE Project*, 1996
- [Fifa 2001] FIFA Publications and Handbook, <http://www.fifa.com/fifa/handbook/index.handbook.html> (Last accessed 22/1/2001)
- [Frankel 2001] Frankel, C.B., Ray, R.D., Competence, Emotion and Self-Regulatory Architecture. In: *Emotion, Cognition & Affective Computing*, AISB2001-Agents and Cognition, 2001
- [Franklin 1997a] Franklin, S.P., *Artificial Minds*, MIT Press, ISBN 0-262-56109-3, 1997
- [Franklin 1997b] Franklin, S.P., Autonomous Agents as Embodied AI, *Cybernetics and Systems*, Vol.28, No.6, pp. 499-510, 1997
- [Franklin 1996] Franklin, S.P., Graesser, A., Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: *Third International Workshop on Agent Theories, Architectures, and Languages*, Springer, 1996
- [Franklin 1998] Franklin, S.P., Kelemen, A., McCauley, L., IDA : A Cognitive Agent Architecture. In: *IEEE SMC 98*, 1998
- [Furukawa 1995] Furukawa K., Michie, D., Muggleton, S. (Eds.), Reacting, Planning, and Learning in an Autonomous Agent. In: *Machine Intelligence 14*, Oxford: the Clarendon Press, 1995.
- [Garvey 1994] Garvey, A., Lesser, V., *Design-to-time Real-Time Scheduling*, 1994
- [Graca 2001] Graça, P.R., Gaspar, G., Using Cognition and Learning to Improve Agents' Reactions. In: *Adaptive Agents and Multi-agent Systems*, AISB2001-Agents and Cognition, 2001
- [Gray 2000] Gray, B., *Soldiers, Agents and Wireless Networks: A Report on a Military Application*, 2000
- [Green 2000] Green, C., *Is AI the Right Method for Cognitive Science?*, Christopher D. Green – Publication <http://www.yorku.ca/christo/papers/pubs.htm> (Last accessed 25/3/2002), 2000
- [Hasemann 1995] Hasemann, J.M., Robot control architectures application requirements, approaches, and technologies. In: *SCIA 95*, 1995
- [Hayes-Roth 1993a] Hayes-Roth, B., Intelligent control. In : *Artificial Intelligence, Special Retrospective Issue on Most Frequently Cited AI*, Journal Articles, 59, 1993, 211-218
- [Hayes-Roth 1993b] Hayes-Roth, B., An Architecture for Adaptive Intelligent Systems. In: *Special Issue on Agents and Interactivity*, Vol.72, pp.329-365, 1993
- [Hayes-Roth 1992] Hayes-Roth, F., Ertan, L., Terry, A., Hayes-Roth, B., Distributed Intelligent Control and Management : Concept, Methods and Tools for Developing DICAM

- Applications. In : *SEKE 92: Fourth International Conference on Software Engineering and Knowledge Engineering*, 1992
- [Haykin 1994] Haykin, S., *Neural Networks, A Comprehensive Foundation*, Macmillan Publishing Company, ISBN 0-02-352761-7, 1994
- [Holland 1994] Holland, J. H., *Adaptation In Natural And Artificial Systems*, MIT Press, ISBN 0-262-58111-6, 1994
- [Jenkins 1998] Jenkins, J.M., Oatley, K., Stein, N. L. (Eds), *Human Emotions: A Reader*, Blackwell Publishers, ISBN 0-631-20748-1, 1998
- [Kitano 1997] Kitano, H., The RoboCup Synthetic Agent Challenge 97. In : *Fifteen International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997
- [Koza 1993] Koza, J. R., *Evolution of Subsumption Using Genetic Algorithm*, NEC Research Index <http://citeseer.nj.nec.com/koza93evolution.html> (Last accessed 25/3/2002), 1993
- [Koza 1997] Koza, J. R., Genetic Programming. In: *Encyclopedia of Computer Science and Technology*, 1997
- [Laventhol 1987] Laventhol, J., *Programming in POP-11*, Blackwell Scientific Pub. Ltd., ISBN 0-632-01528-4, 1987
- [Lent 1999] van Lent, Laird, J. E., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K., and Tedrake, R., Intelligent Agents in Computer Games, In: *Proceedings of the National Conference on Artificial Intelligence*, July 1999, Orlando, FL, pp. 929-930
- [Maes 1990] Maes, P. (Eds), *Designing Autonomous Agents*, MIT Press, ISBN 0-262-63135-0, 1990
- [Maes 1995] Maes, P., Artificial Life meets Entertainment: Lifelike Autonomous Agents. In: *Special Issue on New Horizons of Commercial and Industrial AI*, Vol. 38, No.11, pp. 108-114, 1995
- [McCarthy 2000] McCarthy, John, Basic Questions (created on 4/4/2000), <http://www-formal.stanford.edu/jmc/whatisai/node1.html> (Last accessed 4/1/2001)
- [Minsky 2000] Minsky, M., Future Models for Mind-Machines, In: *SYMPOSIUM on How to Design a Functioning Mind*, AISB 2000 Convention, 2000
- [Murphy 2000] Murphy, R., Using Robot Competitions to Promote Intellectual Development. In: *AI Magazine*, Volume 21, No.1, 2000
- [Nilsson 1993] Nilsson, N., Rumelhart, D., *Approaches to Artificial Intelligence*, 1993
- [Norman 1994] Norman, T.J., Position paper: Motivated Goal and Action Selection, In: *Models or Behaviours, which way forward for robotics?*, AISB workshop, 1994
- [Paiva 1999] Paiva, A. and Martinho, C. (Eds.) Training a Football Playing Robot Using Emotion Based Learning Architecture. In: *Affect in Interactions Workshop Proceedings*, Section 2: Architectures and Theories in Motion, 1999
- [Pollack 1998] Pollack, M., Plan Generation, Plan Management, and the Design of Computational Agents. In: *Proceedings of the 3<sup>rd</sup> International Conference on Multi-Agent Systems (ICMAS)*, 1998
- [Posner 1989] Posner, M., *Foundations of Cognitive Science*, MIT Press, 1989
- [Scheutz 2000] Scheutz, M., AI as a Method?, In: *Commentary on Green on AI Cognitive Science* <http://www.cs.bham.ac.uk/research/cogaff/0-INDEX00-05.html>, 2000

- [Scheutz 2001] Scheutz, M., Logan, B., Affective vs. Deliberative Agent Control. In: *Emotion, Cognition & Affective Computing*, AISB2001-Agents and Cognition, 2001
- [Schoenauer 2000] Schoenauer, M. (Eds.), Interacting Trajectories in Design Space and Niche Space: A Philosopher Speculates About Evolution. In: *The Sixth International Conference on Parallel Problem Solving from Nature*, 2000
- [Searle 1995] Searle, J., *The Rediscovery of the Mind*, MIT Press, Cambridge, 1992
- [Simon 1979] Simon, H.A., Motivational and emotional controls of cognition, *Models of Thought*, Yale University Press, 1979
- [Singh 1998] Singh, M., Rao, A. (Eds.), The Belief-Desire-Intention Model of Agency. In: *Intelligent Agents V*, 1998
- [Sloman 1987] Sloman, A., Motives Mechanisms Emotions. In: *Emotion and Cognition*, pp.217-234, 1987
- [Sloman 1995] Sloman, A., Exploring design and niche space. In: *Proceedings of 5<sup>th</sup> Scandinavian Conference on AI*, 1995
- [Sloman 1997] Sloman, A., Architectural Requirements for Autonomous Human-like Agents. In: *DFKI Saarbrücken 6<sup>th</sup>*, 1997
- [Sloman 1999] Sloman, A., *SimAgent: A Toolkit for philosophers and engineers*, 1999
- [Sloman 2000a] Sloman, A., *Are Turing Machines Relevant to AI?*, 2000
- [Sloman 2000b] Sloman, A., Introduction: Models of Models of Mind. In: *The DAM Symposium: How to Design a Functioning Mind*, 2000
- [Sloman 2001a] Sloman, A., *SimAgent: A Toolkit for philosophers and engineers*, 2001
- [Sloman 2001b] Sloman, A., Varieties of Affect and the CogAff Architecture Schema. In: *Emotion, Cognition & Affective Computing*, AISB2001-Agents and Cognition, 2001
- [Sloman 1994] Sloman, A., Beaudoin, L., Wright, I., Computational Modelling of Motive-Management Processes. In: *Conference of the International Society for Research in Emotions*, 1994
- [Stone 2000a] Stone, P., *Layered Learning in Multiagent Systems : A Winning Approach to Robotic Soccer*, MIT Press, ISBN 0262194384, 2000
- [Stone 1998] Stone, P. & Veloso, M., A Layered Approach to Learning Client Behaviours in the RoboCup Soccer Server. In : *Applied Artificial Inteligence*, 12, 1998
- [Stone 2000b] Stone, P., Veloso, M., Riley, P., CMUNITED-98 Simulator Team. In: *AI Magazine*, Volume 21, No.1, 2000
- [Tambe 2000] Tambe, M., Raines, T., Marsella, S., Agent Assistants for Team Analysis. In: *AI Magazine*, Volume 21, No.3, 2000
- [Tomoichi 2000] Tomoichi, T., LogMonitor – Analysing Good Plans to Train Player Agents. In: *AI Magazine*, Volume 21, No.3, 2000
- [Tanaka 2000] Tanaka-Ishii, K., Frank, I., Arai, K., Trying to Understand RoboCup. In: *AI Magazine*, Volume 21, No.3, 2000
- [Veloso 2000] Veloso, M., Pagello, E. & Kitano, H. (Eds.), Overview of RoboCup-99. In: *RoboCup-99 : Robot Soccer World Cup III*, pp. 1-34, Springer Verlag, Berlin, August 2000

- [Wilson 1998] Wilson, K., CHEMREG: Using Case-Based Reasoning to Support Health and Safety Compliance in Chemical Industry. In: *AI Magazine*, Volume 19, No.1, 1998
- [Wooldridge 1994] Wooldridge, M. & Jennings, N.R. (Eds), *Intelligent Agents*. Springer-Verlag, 1994
- [Wooldridge 1995a] Wooldridge, M., Jennings, N.R. (Eds.), Guarantees for Autonomy in Cognitive Agent Architecture. In: *Intelligent Agents*, Springer-Verlag, 1995
- [Wooldridge 1995b] Wooldridge, M. & Jennings, N.R., *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review, 1995
- [Wu 2000] Wu, X., Building Intelligent Learning Database Systems. In: *AI Magazine*, Volume 21, No.3, 2000
- [Wyeth 2000] Wyeth, Gordon, *RoboCup F-180 League Rules* (revised on 8/5/2000), <http://parrotfish.coral.cs.cmu.edu/robocup-small/rule> (Last accessed 10/9/2000)

## Appendix A

Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
0-0 47-53	1-3 47-53	1-7 51-49	0-4 54-46	2-9 51-49
3-0 40-60	1-1 47-53	1-5 48-52	1-7 50-50	1-8 49-51
0-0 50-50	2-2 45-55	2-4 51-49	0-4 49-51	4-5 47-53
0-0 50-50	1-1 48-52	0-3 51-49	2-0 20-80	1-7 54-46
0-0 52-48	2-1 51-49	0-2 50-50	0-5 26-74	0-10 49-51

Table 1. Results from games with different speed rates

Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
0-0 52-48	1-0 49-51	2-5 51-49	4-6 49-51	3-4 49-51
0-0 51-49	2-1 43-57	4-4 47-53	2-2 48-52	1-3 42-58
1-0 46-54	0-2 50-50	0-2 53-47	1-8 46-54	4-6 50-50
0-0 47-53	0-0 46-54	1-2 52-48	1-3 49-51	2-6 49-51
0-0 51-49	2-1 48-52	1-2 53-47	3-10 47-53	5-6 51-49

Table 2. Results from games with different speed rates and ball bouncing off an agent

These data are available in the directory 'Testdata' provided on the CD-ROM. Data from table 1 are in the subdirectory 'Speed', and data from table 2 are in the subdirectory 'Bounce'.

To get data for Table 1, execute program 'soccer23.p' in the respective speed. Similarly, data in Table 2 are taken by executing the programs in 'soccer24.p' in the respective speed.

A0+A0-1sw 600 A0+A0-2sw 700 A0+A0-3pc 500 A0+A0-4nw 200			
A0+A1+1sw 800 A0+A1+2pc 400 A0+A1+4nw 100	A0+A1-2nw 200 A0+A1-3nw 200 A0+A1-4nw 200	A0-A1+1pc 400 A0-A1+2nw 100 A0-A1+3nw 100 A0-A1+5nw 100	A0-A1-2nw 100 A0-A1-3sw 600 A0-A1-4nw 100
A0+A2+3sw 800 A0+A2+4nw 200 A0+A2+5nw 200	A0+A2-1nw 200 A0+A2-2nw 200 A0+A2-4nw 300	A0-A2+3sw 800 A0-A2+4nw 200 A0-A2+5nw 200	A0-A2-1nw 100 A0-A2-3sw 600 A0-A2-4nw 200
A0+A3+1nw 200 A0+A3+3sw 700 A0+A3+4nw 300	A0+A3-1sw 1300 A0+A3-2nw 300 A0+A3-4nw 200	A0-A3+1sw 600 A0-A3+2nw 200 A0-A3+3nw 200	A0-A3-1sw 700 A0-A3-2sw 600 A0-A3-3nw 200
A0+A4+1sw 1100 A0+A4+2nw 200 A0+A4+3nw 200	A0+A4-1sw 1000 A0+A4-2nw 200 A0+A4-3nw 200	A0-A4+1sw 600 A0-A4+2nw 200 A0-A4+3nw 200	A0-A4-1sw 600 A0-A4-2nw 200 A0-A4-3nw 200
A1+A1-1sw 700 A1+A1-4nw 100 A1+A1-5nw 200			
A1+A2+1sw 300 A1+A2+2nw 200 A1+A2+3nw 200	A1+A2-1sw 300 A1+A2-2nw 200 A1+A2-4nw 200	A1-A2+1sw 300 A1-A2+3nw 200 A1-A2+4nw 100	A1-A2-4sw 300 A1-A2-5nw 200 A1-A2-6nw 100
A1+A3+1sw 800 A1+A3+2nw 200 A1+A3+4nw 200	A1+A3-1sw 1100 A1+A3-2sw 800 A1+A3-4nw 200	A1-A3+1sw 600 A1-A3+3nw 200 A1-A3+4nw 200	A1-A3-1sw 600 A1-A3-2nw 200 A1-A3-3nw 200
A1+A4+1sw 1000 A1+A4+2nw 200 A1+A4+3nw 200	A1+A4-1sw 1200 A1+A4-2nw 200 A1+A4-3sw 800	A1-A4+1sw 800 A1-A4+2nw 200 A1-A4+3nw 200	A1-A4-1sw 500 A1-A4-2nw 200 A1-A4-3nw 200
A2+A2-1sw 600 A2+A2-4nw 100 A2+A2-5nw 100			
A2+A3+1sw 300 A2+A3+2nw 200 A2+A3+3nw 200	A2+A3-1sw 300 A2+A3-2nw 100 A2+A3-3nw 100	A2-A3+1sw 200 A2-A3+3nw 100 A2-A3+4nw 100	A2-A3-1sw 300 A2-A3-2nw 100 A2-A3-4sw 200
A2+A4+1sw 300 A2+A4+2nw 200 A2+A4+3nw 200	A2+A4-1sw 300 A2+A4-2sw 200 A2+A4-3nw 200	A2-A4+1sw 300 A2-A4+2nw 100 A2-A4+3nw 100	A2-A4-1sw 300 A2-A4-2nw 100 A2-A4-3nw 100
A3+A3-1sw 300 A3+A3-2nw 100 A3+A3-3nw 100			

A3+A4+1sw 300	A3+A4-1sw 200	A3-A4+1sw 300	A3-A4-1sw 200
A3+A4+2nw 100	A3+A4-2nw 100	A3-A4+2nw 100	A3-A4-2nw 100
A3+A4+3nw 100	A3+A4-3nw 100	A3-A4+3nw 100	A3-A4-3nw 100
A4+A4-1sw 200			
A4+A4-2nw 100			
A4+A4-3nw 100			

Table 3. Average speed of execution per cycle in the program (in milliseconds)

Please note that results from games where agents got stuck in any moment in the program are not listed in table 3. The legends for each of the file are in Appendix B.

The programs executed following A2+A2-4 are made after the power check, and the values of these results are faster in average in comparison to those before that program. Programs for agents A1 against A2 are re-done after the date of power check because the original sets of data were corrupted.

A0+A0- 0-0 52-48 0-0 43-57 0-0 55-45 0-0 46-54			
A0+A1+ 1-0 50-50 0-0 42-58 0-0 45-55	A0+A1- 2-0 33-67 0-0 25-75 0-0 47-53	A0-A1+ 1-0 50-50 1-0 38-62 0-0 46-54 1-0 42-58	A0-A1- 1-0 54-46 1-0 42-58 0-0 44-56
A0+A2+ 2-0 40-60 0-0 43-57 0-0 44-56	A0+A2- 2-1 45-55 0-0 41-59 0-0 33-67	A0-A2+ 0-0 44-56 1-0 43-57 0-0 37-63	A0-A2- 1-0 39-61 1-0 45-55 1-0 43-57
A0+A3+ 0-7 13-87 0-2 46-54 0-1 8-92	A0+A3- 0-6 14-86 0-0 8-92 0-1 19-81	A0-A3+ 0-3 29-71 0-9 16-84 0-2 41-59	A0-A3- 0-5 28-72 0-3 32-68 0-5 12-88
A0+A4+ 0-6 11-89 0-2 9-91 0-7 7-93	A0+A4- 0-4 10-90 0-9 9-91 0-4 11-89	A0-A4+ 0-2 10-90 0-1 8-92 0-8 13-87	A0-A4- 0-7 8-92 0-3 15-85 0-1 9-91
A1+A1- 0-0 55-45 0-1 51-49 2-1 38-62			
A1+A2+ 0-0 62-38 0-2 57-43 0-0 60-40	A1+A2- 0-0 63-37 1-1 42-58 0-1 65-35	A1-A2+ 3-0 58-42 0-0 63-37 1-0 49-51	A1-A2- 0-0 62-38 0-0 58-42 1-0 59-41
A1+A3+ 0-4 19-81 0-6 16-84 0-9 18-82	A1+A3- 0-6 14-86 0-10 19-81 0-6 5-95	A1-A3+ 0-6 46-54 0-5 16-84 0-3 89-11	A1-A3- 0-8 16-84 0-10 9-91 0-9 13-87
A1+A4+ 0-2 8-92 0-1 4-96 0-3 12-88	A1+A4- 0-7 11-89 0-7 5-95 0-12 15-85	A1-A4+ 0-2 8-92 0-6 9-91 0-10 13-87	A1-A4- 0-1 8-92 0-8 11-89 0-4 10-90
A2+A2- 0-0 48-52 0-0 68-32 1-0 23-77			
A2+A3+ 	A2+A3- 	A2-A3+ 	A2-A3- 

0-9 21-79 0-0 68-32 0-5 16-84	0-7 19-81 0-6 13-87 0-0 0-100	0-2 53-47 0-9 17-83 0-2 11-89	0-1 57-43 0-2 33-67 0-10 13-87
A2+A4+ 0-2 10-90 0-1 13-87 0-8 8-92	A2+A4- 0-9 17-83 0-1 7-93 0-7 11-89	A2-A4+ 0-1 11-89 0-5 11-89 0-4 9-91	A2-A4- 0-4 9-91 0-2 8-92 0-5 10-90
A3+A3- 2-3 47-53 1-2 42-58 2-3 46-54			
A3+A4+ 0-2 47-53 2-2 49-51 0-0 48-52	A3+A4- 3-2 18-82 1-1 19-81 0-2 48-52	A3-A4+ 1-1 49-51 0-0 49-51 1-3 45-55	A3-A4- 0-2 50-50 2-3 49-51 3-3 46-54
A4+A4- 1-3 47-53 0-0 48-52 0-2 47-53			

Table 4. Results from games with different types of agents architecture

Please note that results from games where agents got stuck in any moment in the program are not listed in table 4.

## Appendix B

The data gathered for the evaluation of results from different the agents' type are available in the CD-ROM provided. These files are in the directory 'Results', and each sub-directory containing matches of 5-aside football game from different types. For example, sub-directory 'a1va4' contains the results from game with agents having motivators against agents having motivators, priorities, urgency and coach. The labelling of different functionality of agents are identical to those referred from chapter 5.

Each file contains 2 letters after the file name. These legends use for each of the file correspond the following examples:

- File a0+a0-3pc corresponds to a file where the program was executed on the PC under Exceed
- File a0+a1+3st corresponds to a file where the agents during the execution of the program were stuck at some point. These files are use for evaluation in any part of this project
- File a1+a1-1sw corresponds to a file where the program was executed on the Sun Workstation computer specified in section 4.4 under Solaris Windows Operating System
- File a1+a1-4nw corresponds to a file where the program was executed on the Sun Workstation computer specified in section 4.4 without any Operating System. This has achieved by execution such file in the computer that is remotely log-on by another Sun Workstation

Please note that all these files were generated from UNIX in Solaris Windows, they are not appropriate to be read using PC software such as 'Notepad' or 'Microsoft Word'.

This thesis is contained in the CD-ROM under the directory 'Thesis'.

## Appendix C

```
Declaration of Agentworld
/* Environment of the 5-aside football simulation */
Declaration of scoreboard
/* Board for displaying the score and percentage of ball touch by the
team */
Declaration of agentboard
/* Board for displaying the status of all football agents in each
simulation cycle */
Declaration of agents, coaches, ball, football goals and blackboards
for communication
Functions for displaying agentboard, Agentworld and scoreboard
/* The following are methods in the program */
Method Moveto for an agent to move to a particular position
    If agent has nothing to do, move around (random)
        If goalkeeper, move inside it's penalty area
        ElseIf defender, move anywhere behind it's half-field
        Else move around
    Else move to the determine position
    Speed up with the agent's urgency accordingly
    Take energy out of the agent depending on its speed of
displacement
    Make sure the agent would not go off the football field
    If it does, go back to the field by inverting its horizontal or
vertical displacement value
    Display the agent on the field
End of method
Method for resetting the game
/* This method is used when a team had made a score */
    Put agents back to their side of the field
    Put the strikers of the team who just lost a point closer to the
ball
    Add 100 points to all agents' energy
    In case the agent's energy is above 900, set it to a 1000
(maximum value for energy)
    Display agents in the field
    Put the ball in the middle of the field
    Display ball in the field
End of method
Method bordercheck for checking whether the ball has touched the
border of the football field
If ball touches the horizontal sides of the field, invert its
horizontal travelling value for the ball to bounce back to
the field
    If ball touches the vertical side of the field
    If ball touches the line of the football goal, add 1 to the
designated team's score
        Else invert its vertical travelling value for the ball to
bounce back to the field
End of method
Method for the agent's resting
    Add 5 to its energy value
End of method
Method kickball
If agent is a goalkeeper and inside it's penalty area, the agent can
move by holding the ball
    Else set values for the speed of the ball to be kicked
```

```

        Determine the position for the ball to be move to
        Add speed to the ball regarding to the agent's urgency
        Take agent's energy according to the ball's speed of travel
        If position for the ball to go to goes off the football field,
            agent turns to the opposite direction horizontally or
            vertically to kick the ball
        Called method bordercheck
        Display ball in the field
    End of method
Method Moveaway for the agent to dribble the ball
Determine which direction the agent wants to dribble the ball to
If new position for the ball to dribble to goes beyond the football
    field, invert ball horizontal or vertical position
    Take energy from the agent
    Display agent
    Display ball
    Call method kickball
End of method
Method DribbleTo for the agent to move to the direction facing a team-
mate
Determine which direction the agent wants to dribble the ball to for
    facing a team-mate
If new position for the ball to dribble to goes beyond the football
    field, invert ball horizontal or vertical position
    Add speed for the ball to travel to
    Display ball
    Take energy from the agent
    Call method kickball
End of method
Method ClosestPoint to determine the closest co-ordinate to block the
opponent
    Determine which is the most ideal position for blocking the
    opponent
    Call method MoveTo
End of method
Method Shootball to determine where to shoot the ball in the goal
    Determine the direction and position where the ball goes to
    Take energy from the agent
    Call method bordercheck
    Add speed for the ball to travel to
    Display ball
End of method
Method coachdec for the coach to recommend where the footballers of
its team goes to
Find the closest footballer to the ball, tell it to go to the ball
Other footballers, except the goalkeeper and defenders behind its half
    side of the field, follow the lead of the footballer
    closest to the ball
End of method
Method RecMove to go to the recommended position from the coach
    Called method MoveTo with the position from coach's
    recommendation
End of method
Method evalurg for evaluation of agent's urgency
Determine the agent's urgency according to its and the ball's
    positions in the football field
End of method

```

```

Method bounce for bouncing the ball of a footballer if travelling too
fast
    Determine which direction the ball bounce to from the footballer
    Call bordercheck
    Display ball
End of method
Initialise the agents, ball, football field and draw all object inside
the football field
Define method play as number of cycles the program executes
Define method go as number of cycles the program continues to be
executed
Define method vs as number of cycles the program executes with
different types of agents in each team
/* The following are rules for an agent, it repeats the following for
all footballers */
Agent 0
/* For outfielders displacement */
    Call method evalurg if agent is capable to have urgency
Uses timer to measure how long it takes an agent to execute all
function in every 10 cycles
    If agent is a goalkeeper, ignore the rest of the rules in Agent
    0
    If agent is tired, rest and ignore the rest of the rules in
    Agent 0
    If agent senses the ball, add drive to move to the ball
    If agent senses an opponent, use motivators in Agent 3
    Add drive for agent to wander around
    If agent can listen to coach's advice, add drive to the coach's
    decision
End of Agent 0
Agent 1
/* For outfielders to control the ball */
    If agent is a goalkeeper, ignore the rest of the rules in Agent
    1
    If agent touches the ball and it is not possess by anyone else,
    it has trapped the ball (by reflex)
    If agent touches the ball but it is travelling too fast, it
    bounces off (called method bounce)
    If agent has the ball and senses the opponent's goal, add drive
    to shootball
    If agent has the ball and senses team-mate, use motivators in
    Agent 5
    If agent has the ball and senses opponents, use motivators in
    Agent 6 (for passing the ball to a team-mate)
    If agent has the ball and senses opponent, use motivators in
    Agent 4 (for dribbling away from an opponent)
End of Agent 1
Agent 2
/* Rules for goalkeeper */
    If agent is not a goalkeeper, ignore the rest of the rules in
    Agent 2
    If agent is tired, rest and ignore the rest of the rules in
    Agent 2
    If agent senses the ball, add drive to move to the ball
    If agent touches the ball and it is not possess by anyone else,
    it has trapped the ball (by reflex)
    If agent has the ball and senses team-mate, use motivators in
    Agent 5
    If agent has the ball and senses opponents, use motivators in
    Agent 6 (for passing the ball to a team-mate)

```

```

        If agent has the ball and senses opponent, use motivators in
            Agent 4 (for dribbling away from an opponent)
        Add drive for agent to wander around
        If agent can listen to coach's advice, add drive to the coach's
            decision
    End of Agent 2
Agent 3
/* Motivators to find the closest opponent to block */
    Find the closest opponent if the agent has motivators
If agent can communicate with team-mates and has motivators, odd out
    the opponent that another team-mate is mark to block to
    Opponent chosen, add drive for the opponent to block to
End of Agent 3
Agent 4
/* Motivators to find the closest opponent to dribble away from */
    Find the closest opponent if the agent has motivators
If chosen opponent is far away, add drive for the opponent to dribble
    to in high value
    If chosen opponent is close, add drive for the opponent to
        dribble to in low value
End of Agent 4
Agent 5
/* Motivators to find the closest team-mate for passing the ball to */
    Find closest team-mate if the agent has motivators
End of Agent 5
Agent 6
/* Motivators to find the closest opponent to avoid getting the ball
from passing */
If agent is between the closest team-mate and the opponent, ignore the
    opponent if the agent has motivators (repeat for all
    opponents)
    Find closest opponent if the agent has motivators
If distance between itself and closest opponent is shorter than itself
    and the closest team-mate, do not pass to this team-mate
    In case the agent do not pass the ball to this team-mate, called
        Agent 5 to find the next closest team-mate
If distance between itself and closest opponent is higher than itself
    and the closest team-mate, set drive to pass the ball to
    the team-mate with low value
If the agent is between the closest team-mate and the closest
    opponent, set drive to pass the ball to the team-mate with
    high value
End of Agent 6
Agent 7
/* Execute functions with the highest drive value */
If agent can communicate with team-mates and realise that a team-mate
    is going after the ball, lower its drive for moving to the
    ball if agent is capable to have drives
    If the agent is capable to have drives, choose the task to
        execute with the highest drive value. Otherwise, do all
        the following tasks if the agent satisfies the rules from
        Agent 0 to 6
        If agent moves to the ball, call method MoveTo
        If agent blocks an opponent, called method ClosestPoint
        If agent wants to wander around, called method MoveTo
        If agent wants to shoot the ball to the goal, called method
            Shootball
        If agent wants to pass the ball to a team-mate, called method
            DribbleTo, and let team-mates know which team-mate to pass
            the ball to if communication is permitted
        If agent wants to dribble the ball, called method Moveaway

```

```
        If agent wants to go to the position the coach recommends,  
            called method RecMove  
End of Agent 7  
End of program
```

Pseudo code. 5-aside football simulation

Please note the messages between /\* and \*/ are comments to clarify the pseudo codes used. The word 'drive' used in the pseudo code corresponds to 'priority' in this project.