A MULTIAGENT BASED SIMULATION FRAMEWORK FOR MAMMALIAN BEHAVIOUR

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

EMMANUEL AGIRIGA

November 2016

Contents

Illustrations

List of Figures

List of Tables

Abstract

The primary aim of mammalian behaviour simulation is to allow "behaviourologists" to extend their current knowledge without needing to resort to expensive and intrusive real life experimentation. A useful mechanism for realising mammalian behaviour simulation is provided by the idea of Multi-Agent Based Simulation (MABS) where each "player" in a simulation is represented by an agent with a particular set of features or capabilities. This thesis proposed the Mammalian Behaviour MABS (MBMABS) framework.

The fundamental idea presented in this thesis is that each mammal featured in the simulation can be modelled as an agent that has a set of desires and a set of behaviours. The desires may be static, in that they do not change for the duration of a simulation, or dynamic in that they change with time during a simulation (influenced by some internal or external event).

In the work presented behaviours are modelled using the concept of a behaviour graph comprised of vertices representing states and edges indicating possible state changes. State changes occur as a result of an agent completing some self-appointed task or as a result of some external event. Each state has one or more predefined potential follow on states. Where there is more than one follow on state selection is made according to a weighted random selection process. The weightings are derived dynamically according to individual agent's desires. A particular novel element of the proposed approach is that it features a degree of randomness, agents will not behave in the same manner on each occasion that a simulation is run.

The operation of the MBMABS framework is illustrated in this thesis using a collection of mouse behaviour case studies, in which real mice are represented as individual agents. The reported evaluation of the case studies demonstrated that the proposed framework readily supports rodent behaviour simulations. The reported evaluation also indicated that the proposed simulation framework readily allows users to observe the behaviour of the simulated entities.

More specifically the evaluation of the simulations was conducted by: (i) comparing the operation of the proposed MBMABS with video data, (ii) visual observation and (iii) reference to domain experts. The MBMABS experiments conducted using video data successfully indicated that there was a similarity in the behaviour of mouse agents operating within the framework and real life mice (as recorded using video data). Mouse behaviour such as thigmotaxis and nest site selection was observed in both the simulation and video. The evaluation also indicated that the MBMABS framework readily

supported the addition of states and desires. However, is was also noted that: (i) as the number of states increased the behaviour graph became more complex and difficult to visualise and (ii) as the number of agents interacting with the behaviour graph increased, the performance of the proposed framework was also affected in the sense that it required more resources to operate optimally.

Acknowledgements

I would like to thank my Supervisors, Prof. Frans Coenen and Prof. Jane Hurst first and foremost. Without their support and direction, I would not be anywhere close to submitting this thesis. They have supported me throughout, and I really don't have enough words of thanks. I would like to thank my family for their patience, even in the face of bereavement, they stood by me in tears, urging me on. I especially appreciate those who I was unable to visit, those who I lost while doing this research, I am grateful to you because your strength helped me through. I would like to thank Muhammed Tufail, who provided me with significant assistance. I want to wish Muhammed and his entire family the best of luck. Thanks also to Fatimah, Latifah, Joe, Matt and Ashley; PhD colleagues past and present for their support. And of course, thanks to all who I have not remembered. Thanks to Helen for all the administrative support I received and to the department for the opportunity I was given to be here. Thanks to my examiners for their very important feedback with respect to my work.

Chapter 1

Introduction

1.1 Overview

This thesis is concerned with an investigation into the nature of Multi-Agent Based Simulation (MABS) frameworks specifically to support the simulation of animal behaviour. MAS simulation is concerned with the use of Multi-Agent System technology to support computer simulation [2, 3]. Although MABS may be applied in the context of a number of application domains, they are particularly suited to domains that feature a number of self-deterministic entities (such as animals or humans) where it seems natural to model these individual entities as agents (by definition agents are self-deterministic). Conceptually agents in a MABS framework exist within some multi-agent platform where they interact with one another using the mechanisms supported by the chosen platform.

There are a number of challenges associated with MABS:

1. The mechanism for representing the desired behaviour of agents and storing this in a manner that supports effective operation of a MABS.
2. The mechanisms to support the operation of a MABS so that agents behave in a manner that is as realistic as possible (in the case of behaviour simulation this will involve a degree of randomness).
3. The need for a scalable solution that can handle potentially large numbers of agents.

Note that the above challenges are related. In response to these three challenges the central idea proposed in this thesis is the concept of a "behaviour graph": a mechanism that used to represent/store agent behaviour that lends itself to usage in the context of MABS while at the same time providing for scalability. The behaviour graph is specifically intended to address the above challenges. Given a behaviour graph the vertices represent states while the edges represent potential transitions between states (vertices). Note that with respect to a given MABS the agents may either all subscribe to a single behaviour graph or they may each have individual graphs associated with them (or a mixture of the two). Whatever the case each agent is always conceptually located somewhere within its behaviour graph. The behaviour graph idea has parallels with the idea of "state diagrams" and "state charts", as used to describe the behaviour of complex systems, and the idea of Finite Sate Machines (FSM) [4, 5]. The distinction is that behaviour graphs, as conceived of in this thesis, are designed specifically to support the operation of agents in the context of MABS, especially MABS to support animal behaviour studies.

Another important distinction between agents operating using a behaviour graph, as proposed in this thesis, and MAS founded on (say) the idea of FSM or Belief Desire Intention (BDI) models, is that the operation of the agents contained in the proposed MABS should feature a degree of randomness. On each occasion that a simulation is run, given the same scenario, the agents should not necessarily behave in an identical manner; there should be a degree of unpredictability in their operation. This is an important distinction between the behaviour graph concept and other more prescriptive mechanisms. The idea of randomness is included within the behaviour graph concept so that the way agents "move around" the graph is different on each occasion, this is typically not the case when using (say) FSM or BDI models. Another important element of the behaviour graph concept is the idea of desires, these are "objectives" which agents operating within a MABS framework wish to achieve. These desires may be constant or dynamic. A constant desire is one that persists at a constant strength throughout a simulation. A dynamic desire is one whose strength changes during a simulation according to internal (controlled by individual agents) or external (not control by individual agents) influences. As such the idea of desires as espoused in this thesis has some similarity with the BDI model of operation (referred to above) used in some MAS in the context of collaborative problem solving [6]. The distinction is that the agents in a MABS do not necessarily have a specific problem to solve; they may have a set of time dependent competing objectives of varying significance (corresponding to the varying strengths of each agent's desires, desires that change with time), or no objectives at all.

The application focus for the work described in this thesis, the motivation, is animal behaviour simulation. In this context the MABS paradigm is an excellent fit, each animal can be represented as an agent, as can be the environment in which they are intended to inhabit and any other objects that might exist within that environment. The remainder of this introductory chapter is organised as follows. The motivation for the work is discussed further in Section 1.2. Section 1.3 then presents the research question formulated to direct the work described throughout the rest of this thesis. Section 1.4 presents the adopted research methodology, Section 1.5 presents the main contributions of the work, Section 1.6 presents the structure of the remainder of this thesis, and Section 1.7 the publications to date arising from the work described. This introductory chapter is concluded with a summary in Section 1.8.

1.2 Motivation

Simulation is used extensively to study real world scenarios by replicating the interactions between entities of all kinds. Examples include financial market analysis [7], government policy formulation [8], transportation and traffic simulation [9] and manufacturing process analysis [10]. Computer simulations offer the advantages that: (i) once established they are inexpensive to operate, (ii) they are non-intrusive, (iii) they can be used for "what if" style experiments without causing any permanent damage (conditions can be safely varied by applying different parameters, and results recorded), (iv) they provide a simple mechanism whereby experiments can be repeated using the same or a different set of parameters, and (v) they provide an excellent tool to enhance understanding of some domain of interest. Most current work on computer simulation has been directed at human behaviour simulation [11–13]. However, there is a growing interest in the computer simulation of animal behaviour [14, 15].

The work presented in this thesis focuses on animal behaviour simulation. More specifically rodent simulation, especially mice (specifically harvest mice) behaviour; although the work also has more wide reaching benefits in the wider context. The goal of animal behaviour simulation is to allow behaviourologists to extend their current knowledge without needing to resort to expensive and intrusive real life experimentation. The knowledge gained from such simulation studies is significant in variety of ways as follows:

1. Rodents unwittingly act as vectors for pathogens, causing many zoonotic diseases (diseases affecting humans as a result of human contact with infected animals) and livestock diseases. For example rodents are known to aid the transmission of Lassa Fever by carrying the primary host of the virus which causes the fever [16]. This Lassa Fever virus is estimated to cause about 5000 deaths annually, and it is strongly believed that rodent control may be an effective way of limiting the disease [16]. A rodent behaviour MABS, of the form envisioned in this thesis, can provide a better understanding of how best to provide this control.

2. Rodents can sometimes cause expensive damage by feeding on food crops (and garden plants). It is estimated that rodents destroy or spoil a significant proportion of the world's food supplies. Costs worldwide run to many billions of pounds, while the extent of food loss is of particular current concern with respect to global food security [17]. A recent innovation in pest control is the use of strategic scent signals (semiochemicals) to manipulate rodent behaviour. Of particular note in this context is the LoLa project [18] whose objective is to develop new tools and strategies for rodent pest control to reduce the considerable damage that rodent pests cause[1]. To develop ideas concerning the use of scent controls requires knowledge of how rodents behave within a wide range of complex habitats, knowledge that is

[1]The lead on the LoLa project is Prof. Jane Hurst who has provided significant input to this thesis with respect to the evaluation of the reported outcomes; she was the "domain expert" for the knowledge incorporated into the behaviour graph concept

currently only partially available. Mouse simulations of the form facilitated by the research presented in this thesis provide a step towards providing this knowledge.

3. It has also been suggested that simulation allows behaviourologists to acquire knowledge concerning: (i) the survival instincts of animals [19]; (ii) how animals learn, for example simulation may provide an insight into how rodents adapt their learning in the context of certain activities [20]; and (iii) animal conservation and control.

1.3 Research Question

From the foregoing the research presented in this thesis is directed at an investigation of the provision of a Multi-Agent Based Simulation (MABS) framework that can be used to model the behaviour of animals, with a focus on mouse behaviour, although the techniques proposed are applicable to a much wider range of application. Important aspects of the framework (as already noted) are that it supports the idea of randomness, the behaviour of the agents should not be precisely predictable each time a simulation is run; and be scalable. The research question addressed by the thesis is thus formulated as follows:

How can multi-agent based technology best be used to simulate animal, especially rodent, behaviour in as realistic a manner as possible?

Note the term "realistic" in this context implies a degree of randomness (this is expanded upon in the following section, Section 1.4). The resolution of this research question necessitated the consideration of a number of research issues which were formulated in consultation with domain experts (rodent behaviourists) and are as follows:

1. Given that each agent (entity) within a MABS will posses a particular set of features and traits, a suitable mechanism whereby these features and traits can be represented, in a well structured manner, was required. The central idea here, as noted above, is the usage of a behaviour graph, although the nature of this graph was unclear at commencement of the programme of research.

2. As noted in the introduction to this chapter an important element of the proposed behaviour graph structure is the concept of desires. The idea here is that desires will affect the operation of the behaviour graph when invoking "state changes", although how this would operate was a matter for the research.

3. Following on from (2) above it was also unclear how desires would be encapsulated and how they would change with time as a simulation progressed.

4. Individual agents will need to be able to make autonomous decisions based on their surroundings and desires; some appropriate mechanism for doing this would therefore also need to be incorporated.

5. The agents will exist in an environment, possibly an agent in its own right, which will also have certain features associated with it; appropriate techniques would be required to represent such environments, and the interface with the activities of other kinds of agents.

6. The desired animal behaviour MABS, unlike the kind of problem solving usually conducted with respect to more standard forms of MAS, needed to feature a degree of randomness; the MABS agents should sometimes behave in an unexpected manner because this is what animals do in real life. Some mechanism for achieving this would thus also be required so that such randomness could be built into the MABS.

7. As also already noted, any solution to the above issues must be scalable; scalability was thus also identified, in its own right, as a research issue requiring investigation in the context of this thesis.

1.4 Research Methodology

To achieve the desired research goals the adopted research methodology was to commence by considering MABS for animal behaviour study in an abstract context, the idea being to produce a generic solution with respect to the above identified research issues that could then be refined with respect to (harvest) mouse behaviour simulation. The behaviour graph concept, together with its operation in the context of changing desires was thus modelled in the abstract first and tested using large number of agents (up to 1000 agents). The aim here was to confirm scalability. More specifically the intention was to conduct large-scale stress testing using many agents.

Broadly it was anticipated that the generic animal behaviour MABS should allow for the modelling of the following primary activities:

- Movement,
- Exploration,
- Nest Site Discovery,
- Safe Travel Route Identification and
- Nest Site Defence

Once the MABS mechanisms underpinning the generic idea of an animal behaviour MABS has been established, the next stage was to consider a sequence of specific harvest mouse behaviour case study categories in consultation with a leading rodent behaviourist (Professor Jane Hurst of the Mammalian Behaviour and Evolution group, Institute of Integrative Biology at the University of Liverpool who had agreed to provide support for the project). The case study categories to be considered would be of increasing complexity starting with a “mouse in a box without obstructions case study category”. The case study categories would then be progressively modified to include more complicated landscapes that feature more complex obstructions. The case study categories were all to be founded on the sort of experiments conducted by rodent behaviourists interested in observing the way that mice behave when placed in an enclosed environment (specifically a 1.22m X 1.22m box). This was so that the simulated behaviour could be compared with the known behaviour from real life experiments (the process for this will be presented later in this section). Three categories of case study were considered as follows:

1. **Single mouse in a box without obstructions**: In this case study category the mouse agents were expected to exhibit a common mouse characteristic known as *thigmotaxis*, an affinity to walls [21], explore their environment, and to find a nest site. Note that the box used for this category of case study has no obstructions within it.
2. **Single mouse in a box with obstructions**: In this case study category, mice were expected to explore their surroundings, the ultimate goal is to find and maintain an “optimum” nest location. The box used for this category of case study has obstructions, distinguishing it from case study category 1. In total 4 such scenarios were considered :
   (a) O-Box,
   (b) H-Box,
   (c) Tunnel and,
   (d) Maze
3. **Single Mouse Responding to Danger**: The third category of case study considered was the most complex in that it included a broader range of behaviours. More specifically this third category of case study included mouse agents responding to danger and defending their nest sites from intruders. The mouse responding to danger category of case study features box with and without obstructions.

Only male mice were considered with respect to the work presented in this thesis. This is because male mice behaviour is more complex than female mice, some of the reasons for this include that; (i) male mice are more fiercely territorial than female mice [22], (ii) their territories are larger than that of females, and sometimes they have males and females whom they specifically allow into their territories [22].

Evaluation of the effectiveness with which each case study was realised within the MABS context was conducted in terms of corroboration and consistency [23] using three mechanisms: (i) reference to domain experts, (ii) usage of video data and (iii) visualisation. A brief description of each mechanism is provided below:

1. **Reference to domain experts**: The domain experts used were animal behaviourists from the mammalian behaviour group of the University of Liverpool, specifically, the lead on the LoLa project, Prof. Jane Hurst.

2. **Usage of video data**: The video data that was obtained recorded the behaviour of mice in box environments. The video data was analysed using image processing software [24]. Similar video data was obtained with respect to the simulations. Consequently, the real life behaviour of mice could be compared to the simulated behaviour of mouse agents through the simulation data obtained. This was achieved by dividing the environment of interest into a set of grid squares and recording the number of times each grid square was visited in the video and in the simulation. If the number of visits per grid square were roughly similar it would be possible to argue that the behaviour was consistent (correct). Simulation visualisation was achieved by creating an interface whereby the progress of individual case studies was animated (in real time) so that the progress could be observed and the "realisticness" of the simulation judged with reference to domain experts.

3. **Visualisation**: The idea here was that if the visualisation was deemed by domain experts to be an accurate reflection of the real behaviour of mice, the simulation (and associated mechanisms) could be argued to be effective.

Throughout the conducted evaluation the overriding criteria for success, as noted in Section 1.3, was the realism of the generated simulations. To be genuinely useful, simulations conducted using the proposed MABS framework had to be as realistic as possible. Realism is a subjective quantity but was considered with respect to the above three listed mechanisms.

1.5 Contribution

The work presented in this thesis makes a number of significant technical and application based contributions. These are summarised in this section starting with the technical contributions as follows:

1. The concept of the behaviour graph used to represent animal behaviour in such a way as to facilitate MABS. The concept of desires, used as a mechanism to support the operation of the behaviour graph by directing agent activity, was also of interest.

2. A mechanism to support the usage of behaviour graphs in a manner that featured a degree of randomness in simulations, thereby creating more realistic mammalian behaviour MABS.

3. The Mammalian Behaviour MABS framework (MBMABS).

4. A mechanism for visualising (animating) simulations in real time so that the progress of simulations could be observed.

5. A novel mechanism for evaluating simulations using real life video data.

The application based contributions were then as follows:

1. A framework (the MBMABS framework) that can be used effectively by animal behaviourologists to conduct simulations in a cost effective manner.

2. A tool that, although directed specifically at harvest mice behaviour, could be extended with respect to other animal types.

Overall the most significant end product of the work presented in this thesis was the MBMABS framework, a MABS framework that can be used to simulate animal behaviour.

1.6 Thesis Structure

The structure of the remainder of thesis is as follows. In Chapter 2 a literature review is presented, whilst the following Chapter 3 considers the application domain. The behaviour graph concept, the central feature of the proposed MBMABS framework, is described and evaluated (in an abstract manner) in Chapter 4. In Chapter 5 the other components realised to support the operation of the MABS framework are discussed. Chapter 6 then considers the realisation of the MABS system using the three case study (scenario) categories presented in Section 1.4. In Chapter 7 an evaluation of the proposed MABS system is presented in the context of a male harvest mouse application domain. Finally in Chapter 8, a summary, some conclusions relating to the research question and issues identified above, and some suggested directions for future work are presented.

1.7 Publications

Two publications have arisen out of the work presented in this thesis. These are itemised below. The significance of the publications is that they were used to inform a number of the chapters presented later in this thesis (as indicated).

1. **E. Agiriga, F. Coenen, J. Hurst, R. Beynon, D. Kowalski, 2011, Towards Large-Scale Multi-Agent Based Rodent Simulation: The "Mice In A Box" Scenario, AI-2011 Thirty-first SGAI International Conference on Artificial Intelligence. Cambridge, England 13-15 December 2011.** This paper described some initial research regarding the usage of MABS to support rodent simulation, including discussion regarding: (i) the components of a framework of the form proposed in this thesis, such as the environment and the entities that operate within the environment; (ii) an investigation into the key

features of these entities (two kinds were specified, dumb entities that have no reasoning ability, and intelligent entities that possess reasoning ability); and (iii) how these entities might interact with each other and their environment. The work presented in this paper contributed to the work presented in Chapters 4 and 5.

2. **E. Agiriga, F. Coenen, J. Hurst and D. Kowalski, 2013, A Multiagent Based Framework for the Simulation of Mammalian Behaviour, AI-2013 Thirty-third SGAI International Conference on Artificial Intelligence. Cambridge, England 10-12 December 2013.** This paper was the first to describe the Mammalian Behaviour Multi-Agent Based Simulation (MBMABS) framework. The primary idea behind this framework was the behaviour graph (the phrase "lattice" is used in the paper) comprising vertices representing states and edges representing possible state changes. The paper proposed that state changes occur as a result of an agent completing some self-appointed task or as a result of some external event, and are directed by individual agent desires. The ideas proposed in this paper were adapted with respect to the work presented in Chapters 4, 5 and 6.

The above two published papers are included in Appendices A and B at the end of this thesis.

1.8 Summary

In this chapter the research domain of interest has been introduced, together with the supporting motivation, for the work presented later in this thesis. The central element of the research is the behaviour graph concept. The central motivation for the work is to provide a mechanism whereby animal (rodent) behaviourologists can gain a better understanding of animal behaviour, specifically male harvest mouse behaviour. However, it is argued that the techniques present have a more general applicability. This chapter has also: (i) presented the adopted research methodology including the evaluation strategy, (ii) listed the main contributions of the work and (iii) provided an overview of the structure of the remainder of the thesis. In the next chapter, Chapter 2, a review of previous work related to that presented in this thesis is given together with comparisons of related concepts such as Belief Desire Intention (BDI) models and Finite State Machines (FSM).

Chapter 2

Literature Review

2.1 Introduction

In this chapter a review of previous work, related to the research described in this thesis, is presented. The review is directed at computer simulation in general and the Multiagent Based Simulation (MABS) technique in particular. A number of key MABS modelling concepts and MABS frameworks for creating MABS are presented. The objective of this review is to underscore that MABS is suitable for modelling complex behaviour, such as animal behaviour; and to examine some of the MABS concepts applied to animal behaviour simulation.

The chapter is structured as follows. Section 2.2 presents an overview of computer simulation techniques in the context of the animal behaviour focus of this thesis. Section 2.3 discusses three alternative computer simulation technologies to MABS (Equation Based Simulation, Monte Carlo methods and Expert Systems Based Simulation) whereby the desired mammalian behaviour simulation might be undertaken and gives reasons why these techniques were considered inappropriate. Section 2.4 then discusses MABS in general while the following section, Section 2.5, considers critically a number of techiques for supporting the concept of MABS, namely: (i) individual modelling, (ii) Finite Sate Machines (FSM) and (iii) Belief Desire Intention (BDI) models. Section 2.6 then provides a review of MABS Frameworks, the focus of this thesis. Section 2.7 then presents a summary of the chapter.

2.2 Computer Simulation

The use of computer simulation can be traced back to the 1940s when advancements in technology allowed John von Neumann to use computer simulation to investigate neutron diffusion with respect to the design of hydrogen bombs [25]. In this section the concept of computer simulation is discussed, including its history and its application.

The advancement of computing technology has meant that computer simulation has increasingly become a central element of scientific experiment [26]. Computer simulation is concerned with the use of computer science techniques and technology to realise a model of some system [27, 28]. The models are typically implemented in such a way that they can be reconfigured and run using differing parameters [28, 29]. The idea being to support "what-if" style experimentation [30]. This affords researchers the opportunity to better understand natural phenomena [28]. Computer simulation is widely utilised today as a scientific tool for studying the behaviour of real world systems [31].

The remainder of this section is organised as follows. The rational for computer simulations is first considered, in general terms, in Subsection 2.2.1. Subsection 2.2.2 then considers computer simulation in the context of a range of application domains. This is followed in Subsection 2.2.3 with some general idea concerning the criteria for evaluating computer simulations.

2.2.1 Rational for Computer Simulation

The rational for animal behaviour computer simulation, particularly rodent behaviour simulation, was presented in Chapter 1. This sub-section considers this rational in a wider, more general, context in terms of the advantages that can be gained; including the advantages with respect to animal behaviour computer simulation considered earlier. Broadly the advantages for computer simulation are very widespread and application dependant. However they can be itemised in a general manner as follows:

1. **Scope for measuring behaviour**: Some behaviour may be difficult to measure or even be noted in a real life experiment due to the speed with which they occur, or the time when they occur [32]. Computer simulations provide more scope for measuring behaviour because within computer simulation time can be controlled, therefore behaviour can be slowed down or sped up so as to be measured or investigated.

2. **Cost Saving**: Computer simulation allow models of system to be examined before they are actually created, thus reducing the risk and associated cost of creating new systems [33].

3. **Suited to What-If Style Experimentation**: Computer simulation models can be readily reconfigured with respect to varying parameters so that the effect the changing of parameters has on the system being simulated can be investigated [34]. As a result of their reconfiguration capability they readily lend themselves to "what-if" style experimentation.

4. **Safe**: Simulation is a safe way of analysing critical problems without incurring any risk with respect to participants [35].

5. **Illustration of Emergent Phenomena**: They provide for the illustration of emergent phenomena which could not otherwise be envisaged [36]. For example a set of algorithms were used to simulate the potential effects of global climate change in the United States with respect to the simulation approach described [37].

6. **Visualisation**. There is a lot of scope offered within computer simulation for visualisation of simulations, especially with the constant improvement in computer graphics technologies [25]; thus creating a functional way of presenting scientific concepts. One of the most significant examples of a functional visualisation is presented in [38] with respect to live spacial distributions.

7. **Non Intrusive**: They are non intrusive, an important issue when dealing with people [36] and/or animals [39].

2.2.2 Application of Computer Simulation

As highlighted above, the steady advancement in computer science has led to increasing use of computer simulation in a wide variety of application domains. To give a flavour of the utility of computer simulation this section itemises a number of common application domains where computer simulation has been applied.

- **Agriculture**: In the context of agriculture, computer simulation has been used for simulating crop growth and cropping systems [40]. One example is CropSyst, a multi-crop modelling tool which simulates soil productivity, decomposition and other soil management parameters in terms of productivity and the environment [40].

- **Finance**: In the domain of finance, computer simulation has been applied to risk management [41]. One example is INFRISK [42], a risk management modelling tool which simulates infrastructure project finance transactions involving the private sector in terms of their exposure to a variety of market credit and performance risks.

- **Healthcare**: In the context of healthcare, computer simulation has been used to simulate clinical trials [43]. One example was the use of Simul8, which is a generic simulation tool to simulate a randomised clinical trial for adjuvent breast cancer in terms of the processes involved in adjuvent breast cancer clinical trials [42].

- **Ecology**: In the context of ecology computer simulation has been used to simulate population viability [44]. One example is VORTEX, a population viability modelling tool which simulates population viability in terms of birth and death processes, as well as the transmission of genes between parents and their offspring [45].

There are many more example application domains illustrating the wide applicability of computer simulation. However, the work presented in this thesis is focused on behavioural simulation, specifically animal behaviour simulation.

2.2.3 Evaluation Of Computer Simulations

The evaluation of computer simulation environments is usually achieved by considering two central requirements: (i) corroboration and (ii) internal consistency [23]. Both are briefly described below:

- **Corroboration**: The term corroboration refers to the requirement that when making comparisons between simulated scenarios and real world scenarios the simulation outcomes are similar [23, 46].
- **Internal consistency**: Internal consistency is concerned with checking that the constituent parts of a simulation environment function in line with the acknowledged underlying concepts and theories used to describe the domain being simulated [23].

The MBMABS presented later in this thesis was evaluated using both corroboration and internal consistency testing. Corroboration testing was conducted by comparing simulated behaviour with real life behaviour (using a variety of mechanisms). Internal consistency testing was conducted through a process of demonstration and reference to domain experts[1]. Of course any computer system needs to also be validated and verified using established software engineering practices.

2.3 Technologies for Computer Simulation

Many technologies have been used to build computer simulation systems. From the literature four categories of computer simulation technology can be identified: (i) Equation based, (ii) Monte Carlo, (iii) Expert Systems based and (iv) MABS. The focus of the work presented n this thesis is MABS (for reasons that will become clear later in this section). Because of the significance with respect to this thesis the Multi-Agent System (MAS) based approach to simulation, Multi-Agent Based Simulation (MABS), is discussed in detail later in this chapter. The first three of the above are considered in some further detail in the following three subsections, Subsections 2.3.1, 2.3.2 and 2.3.3.

2.3.1 Equation Based Computer Simulation

Equation based computer simulation is the earliest form of computer simulation technology. Equation based simulations are founded solely on mathematical equations describing features (behaviours) of the environment to be simulated. More specifically, in equation based simulations the system is represented as a set of equations that define the relationships between the observable behaviour of the entities to be simulated [47]. An example can be found in [30] where equation based simulation was used to model a customer service facility; here equations were used to describe queuing time, service time and so

[1]In particular Prof. Jane Hurst from the Mammalian Behaviour and Evolution Group, University of Liverpool

on. The general disadvantage of equation base simulation is that many simulation application domains do not lend themselves to straightforward mathematical formulation, especially where the entities to be modelled adopt some sort of decision-making process. Also, in the context of the proposed MBMABS we wish to incorporate a degree of randomness which is not readily facilitated by mathematical approaches (although of course it can be contrived).

2.3.2 Monte Carlo Simulation

Monte Carlo simulation encompass a wide range of computational algorithms that operate using a process of repeatedly simulating some process using different sets of randomly generated variables founded on some form of probability distribution or a set distribution. Commonly used distributions include normal, lognormal, uniform and triangular [48]. In this manner a large number of outcomes (models) can be generated each with its own probability of occurrence. The user can then analyse the effect that changes of the parameters of interest have on the simulated domain. Monte Carlo methods are suitable for simulating scenarios that feature a high degree of uncertainty together with large numbers of possibilities. Monte Carlo simulation is well suited to domains that feature a large umber of parameters. A disadvantage is that the number of models that can be generated can run into the thousands [49]. A further disadvantage is that parameter combinations that have very low probabilities of occurring tend to be omitted [48]. Monte Carlo methods have been used extensively with respect to financial simulation [50, 51], but have also been used in other simulation contexts such as in: healthcare and ecological risk assessment [52] and agriculture [53].

The mode of operation of the envisioned animal behaviour simulation mechanism was that it would operate on a temporal loop whereby the subjects of interest move through a spatial environment. The agents would have goals and desires. On each iteration the agents would make decisions about their next move according to these goals and desires. It would be possible to contrive some mechanism whereby a Monte Carlo method is used with respect to the decision making associated with each agent on each iteration. This would provide a degree of randomness, but the decision making (as will become clearer later in this thesis) is not simply a matter of considering a set parameters each with a value distribution. Consequently Monte Carlo simulation was considered unsuitable.

2.3.3 Expert Systems Based Simulations

Expert systems can be broadly defined as computer systems designed to simulate human experts [54]. Expert systems typically operate using *if Antecedent then Consequent* style rules that describe facets about a problem domain [55]. The rules are held in what is referred to as a “knowledge base”. In the context of computer simulation the rules are designed to capture the behaviour of the subject of a simulation [55, 56]. Using wild cards the expert system can operate using incomplete data. General disadvantages associated with expert system technology [57, 58] include: (i) the challenge of deriving the knowledge base (often resource intensive); (ii) that they are not good a working in temporal and/or spatial contexts [59], a disadvantage that has lead to a research domain dedicated to the idea of spatio-temporal reasoning [60]; and (iii) that they are not good at working with inconsistent knowledge [59].

In the context of simulation an example of where Expert Systems have been used can be found in [61], which is closely related to the work done in this thesis (animal behaviour). In [61] the Model of Animal Behaviour (MOAB) expert system is described, an expert system simulation framework for individual-based animal foraging models founded on the use of an expert systems that featured random movement rules. The latter is of interest because it introduces a degree of randomness that was also a required feature of the proposed MBMABS; but only in the context of movement, not decision making. Although realistic simulations were achieved the rules required modification for each set of simulation parameters to be considered.

A disadvantage of expert system based simulation with respect to the MBMABS of interest and thus with respect to this thesis, is that expert systems are intended for deterministic reasoning. In other words, the same result will be produced given the same scenario; not what is needed in the context of the behaviour simulation of interest where a degree of randomness is required. Note that of course a degree of randomness can be contrived in the context of expert systems, as in the case of the simulation system presented in [61], but this is not an ideal solution. Note also that the MBMABS requirement for randomness is not the same as the concept of reasoning with uncertainty (also referred to as probabilistic reasoning) where some parameter settings are unknown. The latter is typically achieved by adding probabilistic weightings so that appropriate reasoning can still be conducted [56]. A frequently used mechanism for the latter is to use Bayesian probabilities to derive “Bayesian rules” [56].

In the context of the proposed MBMABS the decision making process that agents are required to undertake regarding their next move on each iteration can be regarded as a form reasoning; and consequently an Expert System approach might be considered appropriate. This would need to incorporate a degree of randomness and this can be contrived as noted above. It would also be possible to contrive some mechanism for including spatial information in rule antecedents. However, generating large numbers of rules to cover many possibilities, including a spatial element, and adding a probabilistic element would be both challenging and not ideal. Expert system based simulation was thus also considered unsuitable.

2.4 Multi-Agent Based Simulation

The computer simulation technology adopted with respect to the work described in this thesis was that of multi-agent based systems, thus Multiagent Based Simulation (MABS). This was because agent based technology was considered to be particularly well suited to simulations that feature multiple (autonomous) entities each with their own attributes and abilities [62, 63]. In the context MABS, entities whose operation is to be simulated are modelled as agents. As such the unique behaviours, decision making processes, and interactions with other agents can be modelled at a micro level [63]. It is then possible to study a simulation at a more general macro level [62]. The rest of this Section is organised as follows. Subsection 2.4.1 reviews the historical context underpinning MABS and its application. Agents are central to the MABS idea and thus are briefly discussed in Sub-section 2.4.2. Subsection 2.4.3 then presents some advantages of MABS.

2.4.1 History and Applications of MABS

The idea of MABS is generally considered to have been first propsed by Thomas Schielling in the early 1970s who was conducting work on simulating human behaviour. More specifically simulations designed to investigated the links between the prejudicial behaviour of individuals and their intrinsic attributes (age, income race and occupation) [64]. Today there are many examples of the application of MABS with respect to various problem domains; examples include: financial market analysis [65], government policy formulation [8], transportation and traffic simulation [66], and manufacturing process analysis [7].

Following the growth in the popularilty of MABS, a sequence of MABS workshops was established. The first MABS workshop was held in Paris in 1998, collocated with ICMAS'98, the third international conference in Multiagent Systems. In 2002 ICMAS morphed into AAMAS, Autonomous Agents and Multiagent Systems. There has been a MABS workshop at every ICMAS/AAMAS conference since 1998. A review of the MABS papers published at the MABS series of workshops held over the past decade illustrates how MABS has evolved to become the popular simulation technique that it is today [67].

2.4.2 Agents

As noted above agents are central to the operation of MABS. Agents are software entities typically described by the following characteristic features:

1. An ability to function in an autonomous manner [62, 68] .
2. Following on from (1) an ability to make decisions without external influence from a user.
3. An ability to collect information concerning the environment in which they operate;
4. An ability to communicate and/or interact with the user, the environment and other agents [62].
5. An ability to perform designated tasks in collaboration with other agents as initiated by an end user.
6. In some cases an ability to learn and adapt to their environment [68].

Of the above the autonomy characteristic is the most significant, it makes it possible for each agent in a MAS to operate in an independent manner. This is of significance with respect to computer simulation in domains comprising multiple independent entities.

A critical aspect of the operation of MABS is the process for acquiring the attributes and knowledge that the agents are expected to display [46]; and the mechanisms required so that this knowledge can be used. With respect to the MBMABS proposed in this thesis, knowledge acquisition was conducted through a process of consulting with domain experts. How this knowledge was encapsulated and utilised was a subject for the research and is presented in later chapters. In the next subsection, the motivations for multiagent based simulation is presented.

2.4.3 Advantages of MABS

The advantages of MABS will be highlighted in this section by considering the wide application of MABS to various problem domains from which it is possible to derive some of its general advantages. Example application domains include: (i) the analysis of police patrol routes [69], (ii) the analysis of schooling against performance in society [70], (iii) simulation of group learning [71], (iv) studying the spread of HIV [72], (v) analysing social values [73], (vi) studying urban housing schemes to help with setting urban regeneration policy [74], (vii) to characterise aggregations of pedestrians [75] (viii) simulation of social behaviour using agents with values and drives [76], (ix) simulation of collission forces in crowds [77], simulation of large crowds [77]. The advantages offered by MABS in the context of these applications are as follows:

Some identified advantages offered by MABS in the context of these applications are as follows:

- **Efficient Representation**: MABS help to provide a reasonable description of complex systems. In [70] MABS was used to simulate collision forces in crowds. Simulating crowds or groups is considered to be very complex, because crowds show very many divergent and complicated behaviours [71, 77]. To simulate crowd forces using MABS, in [75] each type of participant was identified in the system, and represented uniquely. Individuals in crowds were represented using agents and assigned unique identifications and features; including their locations in an environment, distinctively represented as a combination of cells in a “tile world”. The behaviour being simulated was the application of force; behaviour rules were setup to control when an agent needs to use force to modify its location, and when it does not.

- **Flexibility**: MABS support flexibility across scale [77]; it is not very difficult in most cases to add or remove agents to a MABS. This is advantageous with respect to complex systems for which the number of participants may be changing constantly, for example simulating visits to an amusement park to enhance social coordination [78] such that the MABS system coordinates the demands of a changing number of visitors (to reduce queuing). A second example involves analysing police patrol routes by simulating the physical reorganisation of agents [69]; here the MABS is expected to adapt to a runtime modifying structure through the addition, removal and/or substitution of components. Although this may also be a disadvantage when there is some limitation in computing resources [79].

- **Adaptability**: MABS provides for an adaptable simulation framework, which can be adjusted even at its structural level. For instance, although MABS is fundamentally supported by agents which have autonomy as a key characteristic, MABS supports the simulation of reactive [72] and proactive [80] entities. A reactive entity within a social system is one that depends on external event to direct its actions or inactions within a social system [81], whereas a proactive entity is one that does not depend solely on external events to direct its activities [81].

- **High level of abstraction**: MABS models can be specified to a very high level of abstraction, using techniques like Beliefs, Desires and Intentions (BDI) [38], (the BDI concept is discussed in Subsection 2.5.3). A high level of abstraction makes it easier for users from other disciplines to participate in the MABS design process.

2.4.4 Limitation of MABS

The limitation of multiagent based simulation, again based on consideration of the wide variety of application domains identified in the foregoing section, is that they require a considerable amount of computational resources. This limitation is significant because it is highlighted accross various MABS application domains; examples include crowd simulation [82], characterisation of the aggregation of pedestrians [75], analysing social values [73]. Multiagent based simulation requires significant computing resources for several reasons, some include;

1. **Large number of agents**: Multiagent simulations usually involve large numbers of agents as is the case of applications such as crowd simulation [82] or aggregation of pedestrians [75]. Additionally, the operation of each agent will include some kind of interaction mechanism within the MABS.

2. **Requirements for MABS Visualisation**: The visualisation technology for MABS also requires substantial computational resources [25].

3. **MABS is used for complex behavioural problems**: Practical representation of complex practical behaviour, such as human behaviour using MABS [76], will require significant computational resource. Human characterisation include mechanisms for deliberate decision making, desires, responses and reactions.

The next section considers general techniques which support the operation of MABS, for behaviour studies.

2.5 Techniques For Supporting the Operation of behaviour MABS

In the previous section an overview of MABS was presented, including its advantages and limitations. It was also highlighted in the previous section that MABS has been adopted with respect to a wide range of application domains. However, in this thesis we are interested in MABS for behavioural simulation, especially animal behavioural simulation. This section is thus directed at previous work on behaviour MABS with respect to which a significant amount of work has been directed at human behaviour [11–13, 73, 83] and relatively less on animal behaviour [5, 14, 39, 68, 84]. This work can be broadly categorised according to the techniques used to support the operation of MABS. From the literature the predominant techniques of note are (loosely listed in chronological order):

1. Individual Modelling.
2. State machines.
3. The Beliefs, Desires and Intentions formulation.

Each of these is discussed in further detail in the following three sub-sections, Sub-sections 2.5.1,2.5.2 and 2.5.3. In each case the advantages and disadvantages of the technique are considered, and comparisons made with the proposed Behaviour Graph concept. Recall that the behaviour graph is a sophisticated form of FSM.

2.5.1 Individual Modelling

In the individual modelling approach the behaviour of individual agents is "hard coded" in a bespoke manner without recourse to some generalised idea of behaviour [68]. Individual modelling has been utilised extensively with respect to behaviour MABS. Examples include crowd behaviour simulation with respect to evacuation processes [13, 82, 85, 86] and scheduling [78]. In the context of animal behaviour MABS examples where individual modelling has been adopted can be found in [14], [39] and [5]. In [14] the authors simulated sheep grazing, resting and moving around a field. In [39] the author simulated the grazing behaviour of animals in a pastoral system. In [5] individual modelling was used to simulate how ants collectively choose the best of several nest sites.

The advantage offered by individual modelling is that they tend to result in effective simulations because they do not rely on generalisations as in the case of FSM and BDI models [87]. The significant disadvantage of individual modelling is that it is very time consuming. Its main limitation is scalability [77]. Consequently the individual modelling approach was considered entirely unsuitable with respect to the MBMABS at which the work presented in this thesis was directed.

2.5.2 Finite State Machines

An alternative to individual modelling is the Finite State Machine (FSM) approach; as noted above the proposed behaviour graph concept can be viewed as a sophisticated form of FSM. The concept of FSM was first proposed in the early 1960s [88]. A Finite State Machine (FSM) is a mathematical abstraction that has been used in the context of behaviour MABS. As in the case of the proposed behaviour graphs, FSMs comprises: (i) a finite number of "states" (ii) a set of transformations that occur between these states (transactions have *input states* and *output states*), and (iii) a set of events and/or actions that occur as a result of the transformations [4, 89]. As such FSM have some similarities with the behaviour graph concept presented in this thesis.

As in the case of behaviour graphs FSMs can be conceptualised as graphs where states are represented as vertices and state changes as edges connecting vertices; although the term *state diagram* is usually used to refer to such graphs. An example, taken from [1], is presented in Figure 2.1.

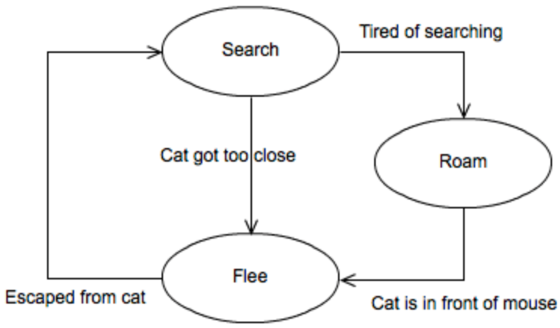


FIGURE 2.1: Example of a simple FSM graph (state diagram) for a mouse entity [1]

An alternative mechanism for representing FSM is in the form of a state transition table (the term characteristic table is also sometimes used). These can take various formats, from the literature we can identify one-dimensional and two dimensional state tables. An example, using the format adopted with respect to this thesis, and in terms of the FSM graph given in Figure 2.1, is given in Table 2.1.

The table has three columns showing: (i) the current state of the mouse, (ii) an event and (iii) a next state for the mouse. The "Current state" of the mouse is the present activity which the mouse is performing. The "Event" is an input condition, which directs a change in activity of the mouse. The "Next state" is the output of an event. For example, row 1 describes the current activity of the mouse to be "Search", if a Cat got too close "event happens". The mouse adopts the follow on state "Flee".

Such tables can be viewed as *decision tables* or *truth tables* where the input is the current state, and the output the selected follow state.

TABLE 2.1: FSM Truth Table for FSM Graph given in Figure 2.1

Current State	Event	Next State
Search	Cat got too close	Flee
Search	Tired of searching	Roam
Roam	Cats is in front of mouse	Flee
Flee	Escaped from cat	Search

Examples where the FSM concept has been used with respect to MABS can be found in [5, 14, 15, 84, 90, 91]. In [15] the FSM it was used in the context of a behaviour MABS to encapsulate the behaviour of mice responding to external event; similar to an idea presented later in this thesis. In [15] state changes occur as a result of some event, which consists of either communication from other agents (mice) or changes to the environment. Note that in the context of FSM there are various mechanisms whereby state changes can be affected depending on the application domain at which they are directed.

In [5] Individual Modelling (see above) was combined with the FSM concept to formulate the behaviour of a group of ants, which collectively attempt to select a best nest site from a number of different alternatives. State transitions occured in a probabilistic manner. Each state details how an "ant agent" encounters a new site, and communicates the location to other ant agents. In the simulation, agents will either reject or accept a site based on assigned probabilities. States are grouped according to categories of action, which occur during four phases of decision-making when selecting the best of several nest site: (i) exploration (ant searching for potential new home), (ii) assessment (found a site to evaluate), (iii) canvassing (provisionally accepts site) and (iv) committed (completely accepts site).

In [84] the FSM concept was used within a behaviour MABS to represent the behaviour of animals in temperate European climates, with the field vole used as a case study. State transitions occurred as a result of conditions described as being either internal or external events. Conditions could have probabilities associated with them (for example, the probability that a male vole will kill a young vole is dependent on the age of the male and how many female voles defend it). As in the case of the work described later in this thesis, external events are not controlled by agents, whereas internal events are.

In [14] the behaviour of foraging sheep was simulated using a variation of the FSM machine concept within a behaviour MABS . The primary behaviour investigated in this work was movement and social behaviour of ewes. Within this behaviour MABS simulated events are scheduled and the state transitions are dependent on: (i) a combination of these simulated events, and (ii) the time spent already in the current state (an agent can remain in a particular state for only a specified amount of time).

Given the above examples of the use of FSM it will become apparent later in this thesis that there are similarities between the idea of a behaviour graph and FSM. The distinction is that the behaviour graph is a sophisticated form of FSM due to the use of the concept of desires to direct state changes, rather than events. Within the behaviour graph concept; events directly influence the operation of desires in such a way that desires may either become stronger, or diminish. This is discussed in detail in Chapter 4.In the next subsection some advantages of finite state machines is presented.

2.5.2.1 Advantages of FSM for Behaviour MABS

The advantages advantages offered by for using Finite State Machines (FSM) with respect to behaviour MABS are briefly highlighted as follows:

- **Easy to implement**: Finite state machines are easy to implement, even when used for the representation of complex behaviour such as animal behaviour [5, 14].
- **Flexible**: Finite state machines are flexible. They can be implemented in several ways. For example, a state can be implemented to represent the behaviour of a single agent [5] or a group of agents [14]. In the same vein the concept of events as triggers for state transitions can be used to represent aspects of agent behaviour within behaviour MABS. Internal events can be used to support one internal agent decision making process, alternatively an external stimuli can be used [15]. An example of the latter might be some kind of interaction with another agent, or component in the environment.
- **Expressive**: Finite state machines are expressive, they are suitable for relaying abstract ideas about behaviour, into meaningful concepts which can then be analysed or tested for correctness [4].
- **State diagrams and transition tables**: State diagrams and transition tables can be usefully employed to represent the components of FSM [4]. For example, using a state diagram and/or transition table it can easily be seen how and why a particular state transition occurs.

2.5.2.2 Limitation of FSM for Behaviour MABS

Traditionally the most frequently quoted limitation of FSM is that they struggle with efficiency, especially with respect to very large or dynamic systems [4]. The larger the system being represented using the FSM technique, the more difficult it will be to describe, manage and understand such a system using the FSM approach, even with the use of state diagrams [4]. Also FSM are not deliberative; in other words, they do not provide a facility to allow agents to "plan ahead" [4]. This is a challenge when using FSM for behaviour MABs of animals because animals are capable of such planning.

2.5.2.3 FSM and the Behaviour Graph

The behaviour graph concept is a form of FSM specifically designed to support MBMABS. As already noted the behaviour graph may be considered to be an extension of FSM which incorporates the following concepts: (i) randomness, (ii) the use of desires to trigger state changes, (iii) time out actions. Each of these is discussed in further detail below:

- **Randomness**: FSM tend to be deterministic. The behaviour graph incorporates randomness with respect to the selection of states. As in the case of FSM the vertices of the behaviour graph represents states, and the directed edges represent state transitions. The selection of follow on states within the behaviour graph concept is conducted in a probability driven random manner. This is discussed in detail in Chapter 4.
- **The use of desires to trigger state changes**: As mentioned earlier, the FSM concept uses state transitions to represent changes in the behaviour of agents within a behaviour MABS. Events trigger state changes within a FSM. This is comparable to the behaviour graph concept. However the behaviour graph uses both events and desires to direct state changes. Desires have strengths (a value associated with each desire) which are in turn influenced by events, in such a way that they may increase or reduce. Desires are linked to weightings associated with follow on states. Again this is discussed further in Chapter 4.
- **Time out actions**: Actions are represented by states within the behaviour graph. Within a behaviour graph, as in the case of the FSM concept, agents can only exist in one state at a time. State changes, as noted above, occur as a result of events. An additional feature of behaviour graphs is that state changes can be triggered as a result of a "time out" event. This amount of time agents can exist in a particular state is probabilistic, and this is discussed in Chapter 4. When an agent using the behaviour graph times out, it is expected that it will select a new state, using the desires concept, as mentioned earlier.

2.5.3 Beliefs Desires and Intentions

The Beliefs Desires and Intentions (BDI) framework is underpinned by ideas concerning how humans conduct rational thinking [92]. As such the framework incorporates some element of reasoning [92]. The BDI concept was originally proposed in 1987 by Bratman [93] as a systemization for practical reasoning with which to describe human reasoning. In the context of MABS each agent maintains a model of their world in terms of beliefs concerning that world. Desires are then goals which each agent wishes to attain, whilst intentions are selected goals that agents commit to, and consequently make plans to accomplish. Beliefs are obtained via various means of input for example from sensors [94]. Beliefs are turned into intentions in a manner guided by agent desires. The selected intention is regarded to be the current task to be performed by the agent.

The BDI framework is more suited to incorporating reasoning into MABS than FSMs. BDI is an improvement on behaviour MABS techniques for agent decision making which use ranking mechanisms, where potential actions are weighted, and the highest weighted action is chosen. Such ranking mechanisms idea may produce a realistic simulation for certain systems, like thermostats and cars, but when applied to complex systems such as human behaviour it will be unreliable in many cases because human or animal behaviour is highly unpredictable and involves deliberation.

Hence, the BDI concept is more suitable for representing human or animal behaviour [92]. Example application domains where BDI has been applied include: (i) the simulation of human behaviour in warfare [80] and (ii) the simulation of social behaviour on the basis of values and drives [76]. In [92] a BDI based behaviour MABS was used to simulate the dynamics of modern irregular warfare, such as terrorist attacks; here agent mental models were structured using the BDI concept. In [76] social behaviour was simulated through implementing agents endowed with "values" and "drives". Drives represented the internal needs of the agents, values were used to rank these drives according to importance. The simulation was used to consider scenarios where agents faced conflicting choices, and where agent behaviour would have a negative effects on other agents. In both examples [76, 80], the behaviour MABS integrated beliefs through a "perception mechanism" which included the use of specific parameters for a given agent, such as change in current location or the presence of another agent.

2.5.3.1 Advantages of BDI for Behaviour MABS

From the literature a number of advantages of BDI can be identified:

- **Intuitive representation**: The BDI mechanism used in the context of behaviour MABS provides an intuitive, easy to understand, high level representation of agent behaviour [76].
- **BDI is well researched**: The BDI concept is well researched; significant work has been done with respect to its implementation to the extent that multiagent system frameworks for creating BDI agents are well established and have been applied to various domains, an example of such framework is JACK [95].
- **BDI Agents can recover from failure**: BDI agents can replan or recover from failure because intentions can be dropped when no longer achievable [96].

2.5.3.2 Disadvantages of BDI for Behaviour MABS

Within the BDI concept, desires are required to be logically consistent; this means that each desire intuitively corresponds to the task allocated to it, and this is not usually the case in real world representations of human or animal behaviour [96]. This disadvantage is sometimes referred to as the "BDI's inability to adapt easily to unplanned changes" [97], otherwise referred to as the *learning problem* [98]. Mechanisms have been proposed to address this issue. One idea is to select plans based on probabilistic methods that identify their likelihood of success; the more a plan is likely to succeed the more its likelihood of selection [99]. However, it can be challenging to determine such probabilities given complex simulation applications.

As established in Subsection 2.5.3.1, the BDI concept provides a clear and conceptual model for representing agent behaviour, but another limitation of the BDI approach is that is assumes agents to always behave rationally [100]. This creates difficulty when considering the BDI approach for use in investigating behaviour MABS for complex behaviour, such as human or animal behaviour because humans and animals are significantly irrational decision makers. This is primarily because human and animal behaviour is usually influenced by a nontrivial combination of internal (for example, disposition or inclination towards an activity or action) and external (for example, social interactions) components. The behaviour graph concept is suitable for such complex behavioural simulation because it uses the concept of desires as the main driver of actions. The concept of desires is influenced by both internal and external events; which can be construed to be a representation of the internal and external components of behaviour.

2.5.3.3 BDI and the Behaviour Graph

There are similarities between the BDI behaviour MABS technique discussed above and the behaviour graph MABS approach proposed in this thesis in that desires have an important role in deciding intention (states using the terminology adopted with respect to the proposed behaviour graph concept). In other words, using the BDI based MABS approach, desires guide the selection of intentions or plans; whilst using behaviour graph MABS approach desires influence the selection of states.

The behaviour graph was chosen over the BDI model because the behaviour graph offers more flexibility with respect to the implementation of desires. In the case of the behaviour graph described in this thesis, desires may be fixed or dynamic, and combine to different degrees to influence follow on states. In the behaviour graph, the dominant desire is the most influential to the decision making process of an agent and depending on the agent's circumstances, this desire may become less dominant as the simulation progresses.

2.5.4 Behaviour MABS Evaluation

In Subsection 2.2.3 a discussion was presented concerning the evaluation of computer simulation environments in terms: (i) corroboration and (ii) internal consistency. This also applies to MABS. From Subsection 2.2.3, corroboration refers to the requirement that for a simulation to be useful it must provide for the accurate representation of real world scenarios. Internal consistency means that the constituent parts of a simulation environment must operate in line with acknowledged concepts [23].
To corroborate a behaviour MABS the following will require consideration:

- **Data Comparison**: Establishing similarities between the behavioural model and the real world scenario. The mechanism for comparing the behavioural model to the real world scenario may include extracting data from real world scenarios and simulations, and comparing the two as was done in [5, 14].

- **Visualisation**: Verifying similarities between behavioural model and real world scenario using visualisation tools available within computer simulations; for example using trace maps to outline the movement behaviour of agents within a simulation environment and comparing to real world scenario [25].
- **Demonstration to domain experts**: Checking that simulation output is correct by presenting results to domain experts to verify [14].

To check for internal consistency, within a behaviour MABS the following need to be considered:

- **Theory**: Checking that the mechanisms used to achieve the desired simulation are in line with current theory concerning the application domains.
- **Mathematical Foundations**: Checking that any mathematical foundations used to encapsulate behaviour operate as expected.

The corroboration and internal consistency approach to MABS evaluation was adopted with respect to the work presented.

2.6 Multiagent Based Simulation Platforms

For the work presented in this thesis a MABS platform of some form was required so as to evaluate the ideas presented. From the literature there are many MABS platforms that have been proposed, each with their own particular unique features. In general MABS platforms are intended to provide support for [13]:

1. The building of an artificial environment.
2. The population of the environment with autonomous agents, which are then able to interact with each other.
3. The operation of a MABS.

In the context of the proposed MBMABS the work is not directed at any particular platform, indeed it can be argued that the proposed MBMABS can be implemented using any platform. With respect to the evaluations presented later in this thesis a MABS platform was simulated so that performance data could be easily extracted (a simulation of a simulation!). However, for completeness a review of a number of commonly used platforms is presented in this section.

Examples of MABS platforms that have been proposed include: (i) Swarm [79], (ii) MASON (Multi-Agent Simulator of Neighbourhoods) [101], (iii) REPAST (Recursive

Porous Agent Simulation Toolkit) [102], (iv) the Animal Landscape and Man Simulation System (ALMaSS) [84], (v) GAMA (Gis and Agent-based Modelling Architecture) [103], and (vi) NetLogo [104]. These can be categorised as being either:

1. General purpose platforms (Swarm, MASON, REPAST and NetLogo).

2. Domain specific platforms (ALMaSS, GAMA).

Each approach is discussed in further detail in the following two subsections.

2.6.1 General Purpose MABS Platforms

The key idea behind general purpose MABS frameworks is that the user need not concern themselves with implementation detail [2]. Hence they are sometimes referred to as toolkits [102]. Usually, general purpose MABS frameworks already have defined aspects or dimensions for MABS components (agents, environment, communication, scheduling, interaction and so on). The user simply has to "populate" the platform with respect to a particular MABS problem.

As noted above, MABS platforms that fall in to this general purpose category include: (i) Swarm, (ii) MASON, (iii) REPAST and (iv) NetLogo. Of these the earliest is Swarm, first proposed in 1994 [79]. In the Swarm MABS Framework the basic unit of simulation is the Swarm which is a grouping of agents executing a schedule of actions. Swarm was developed in Objective C because of its lack of strong typing; this idea is in line with the philosophy behind its development which is to include modelling tools likely to be useful for many models but not specific to any domain [95]. The main limitation of Swarm is that it has very limited GUI features and presents a very steep learning curve for researchers [95].

MASON is another general MABS platform that incorporates 2D and 3D visualisation [101]. Within MASON, agents are designed as computational entities that can be scheduled to perform some action [101]. MASON only provides core tools common to most simulation needs [101]. It does not provide graphing, charting or statistical facilities [95]. MASON is developed in Java and updating Java distributions has the unwanted effect of sometimes further complicating the installation and operation of MASON. For example to use MASON in 3D, you must install Java 3D and installing Java 3D is now very complex due to compatibility issues with various versions of Windows and OSX [105]. REPAST was originally proposed to support social sciences simulation, and includes tools specific to the social science domain, however it also has general applicability [95]. Agents within REPAST are designed as mobile entities with their own rule sets for representing their behaviour [106]. REPAST has implementations in Java and Microsoft's C++. Although REPAST has great functionality, in terms of its ease of use and available documentation, it struggles with computationally demanding models where many agents are executed over many iterations [107]. NetLogo is another high level MABS platform. NetLogo fundamentally designs agents as programmable mobile agents called "turtles", which move over a grid of "patches", which are also programmable agents [108]. NetLogo is written in Java to ensure that it works on Windows and Mac operating systems. It is the most reliable of the general purpose platforms highlighted in this thesis because, it also has associated with it, extensive documentation. Both have contributed to its widespread use [95]. However because it has its own programming language, there is still an issue with the time required to learn a new programming language [95]. In terms of its usage, although NetLogo provides an error checker, it lacks a significant integrated development environment featuring a stepwise debugger [108].

In general, the disadvantage of using general purpose platforms is that, by definition they are generic, and thus typically are not ideally suited to a specific simulation application [84]. It is also sometimes difficult to know in advance if all the tools required for a simulation task will be available with respect to a particular platform. Another issue is that, currently, many available MABS platforms lack complete documentation, and/or lack a clear underpinning philosophy and decision making process [95]. An exception is NetLogo which has extensive documentation. (Note that some of the existing MABS platforms itemized above are either still undergoing development or are undergoing change [103].)

2.6.2 Domain Specific MABS Platforms

Construction of a bespoke MABS platform allows the modeller flexibility to investigate different aspects specific to a particular simulation domain; as noted above, usage of generic platforms tends to restrict modellers and researchers [104]. Additionally, it is usually easier to describe all the components of a particular simulation domain using a domain specific MABS platform because the limitations caused by compromises when using a generic platform can be avoided. Some highlighted examples of domain specific frameworks include, GAMA [103] and ALMASS [84].

GAMA is a domain specific MABS framework for investigating spatially explicit multiagent based simulations. The main objective is the provision of a MABS platform which non computer scientists can use to analyse complex Geographical Information Systems (GIS) data [103]. The ALMaSS platform [84] was designed specifically to find solutions to problems relating to landscape management with respect to key species of animals in Denmark. Agent behaviour was represented using the FSM approach coupled with "behaviour rules" to determine follow on actions (states). These rules were stored in an event handler, and organised in the form of an event-action matrix, such that for every event there was a corresponding action. Animals were represented as individual agents. ALMaSS is thus a good example of a domain specific MABS platform. A potential disadvantage of ALMaSS is that if the set of behavioural rules becomes large, the complexity of the simulation may become problematic.

2.7 Summary

A review of previous work related to the work presented in this thesis has been presented in this chapter. The review commenced with a discussion of computer simulation, and then moved on to MABS. Three mechanism for the realisation of behavioural MABS were then considered: individual based modelling, Finite State Machines (FSM) and the Beliefs, Desires and Intentions (BDI) models. The merits and demerits of individual modelling, FSM and the BDI model were discussed and their relationship with the proposed behaviour graph concept presented. The chapter was completed with a review of existing MABS platforms, both generic and domain specific. In the next chapter the application domain at which the work presented in this thesis is directed, animal behaviour, specifically mouse behaviour, is discussed.

Chapter 3

Application Domain

3.1 Introduction

As established in the foregoing two chapters the particular focus for the work presented in this thesis is animal behaviour simulation, especially in the context of mouse behaviour. The study of animal behaviour has seen a significant increase in the level of research interest over the last decade, as evidenced by the growth in the number of publications available on the subject [104]. From a computer science perspective there has been a parallel growing interest in techniques and mechanisms to support animal behaviour simulation, as shown (similarly) by the amount of recent related work which has been conducted (as indicated by reference to Chapter 2). This chapter considers the animal behaviour application domain in more detail.

The chapter is organized as follows. In Section 3.2 the concept of animal behaviour simulation is considered in general. This is followed in Section 3.3 with a discussion of mouse behaviour simulation in particular, and in Section 3.4 with a brief overview of the nature of harvest mouse behaviour; harvest mice behaviour is used for evaluation purposes throughout this thesis. The chapter is concluded with a short summary provided in Section 3.5.

3.2 Animal Behaviour Simulation

Animal behaviour is the term used to describe a combined set of actions which an animal may perform either in response to certain events or to cause the initiation of certain events. The behaviour of animals can be linked to: (i) movement [109], (ii) sound [110], (iii) body posture [111], (iv) odour changes [112] and (v) scent markings [113]. The last two are typically used in the context of asynchronous communication. From such simplistic patterns of behaviour, such as movement and vocal communication, more sophisticated patterns of behaviour can evolve [114]. For instance, through consideration of a combination of certain kinds of movement, one can understand that an animal is engaged in a hunting activity. It is equally correct to consider animal behaviour in terms of a system of mechanisms and processes by which animals react to changes or events in an environment, and also cause changes and events in that environment, either directly (by themselves) or indirectly (through other animals).

The study of animal behaviour is referred to as ethology, and is a branch of zoology [115]. Ethology is a relatively new branch of zoology concerned with the study of exterior noticeable changes that cause communication and reveal behavioural patterns in animals. Animal behaviour may be studied in various contexts including:

1. **The study of social issues concerning human behaviour**: There is an increasing acceptance by animal behaviourologists and social scientists that animal behaviour is a good platform for understanding human social issues [116]. For example animal behaviour study has contributed to an understanding of the factors surrounding child abuse as described in [116], while in [117] animal behaviour was used to underline the validity of some early determinants of positive and negative behaviour. Animal behaviour is helpful in other human related domains as well; for example in [118] an investigation was reported directed at the emotional responses which influence blood pressure.

2. **Disease spread**: Studies of animal behaviour is also seen as beneficial with respect to controlling diseases spread by animals [114]. A very good example of this is the recent outbreak of the very deadly ebola virus in West Africa, which has been linked to certain species of animals, such as fruit bats, which often coexist with humans [119].

3. **Pest control**: Pest control is a well established motivation for the study of animal behaviour in the context of both homes and business premises and agriculture. The latter especially in the context of food production; crop damage and global food supply is a major world issue [120].

4. **Increased scientific understanding**: The study of animal behaviour is of interest in its own right; significant scientific insights can be obtained from the observance of animal behaviour [121]. Traditional animal behaviour study technique involves the setting up controlled experiments either in the wild or by recreating animal habitats on a smaller scale, possibly in a laboratory setting, and then observing the way in which the target animals behave. Such observation can be done using video recordings or manually. The disadvantage of these traditional methods is that they are time consuming, resource intensive, often invasive, and subject to human error [115]. The use of simulation techniques can serve to address these disadvantages, and also provide for the systematic repetition of scenarios [115].

From the above its is therefore desirable to have simulation techniques which can support the work of behaviourists. In chapter 2 , some techniques which have been used to conduct animal behaviour simulation tasks were discussed. The main objective of this thesis is to provide a platform by which any animal behaviour simulation task can be conducted in a clear and concise manner, although the focus is more on mouse behaviour.

3.3 Mouse Behaviour Simulation

This section provides some further background concerning the mouse behaviour simulation application domain used as a focus with respect to the work presented in this thesis. More specifically the focus is on male harvest mouse behaviour. Male and female mouse behaviours differ, male mouse behaviour is more complex because of the nature of their social interactions. Male mice are fiercely territorial. They usually control a territory larger than that of the female. The females usually control smaller territories within a male's territory. Male mice defend their territory vigorously, prefer to keep away from other male mice and will sometimes attack and kill other male mice within close proximity [22].

It was thus decided, in the context of this thesis, to concentrate on male mice behaviour. The assumption was made that in future work female mouse behaviour could be incorporated into the proposed framework using the mechanisms and techniques developed with respect to male mice.

In the context of mouse behaviour study several mechanisms have been developed to support "real life" experimentation. The studies considered with respect to this thesis, and used to both drive the research work and populate elements of the proposed MBMABS for evaluation purposes, were studies typically conducted by behaviourists in laboratory settings. More specifically studies by the Mammalian Behaviour and Evolution group, Institute of Integrative Biology at the University of Liverpool were conducted using 1.22m $\times$ 1.22m "box environments" with various kinds of objects placed in them; objects such as wooden blocks were used to represent obstacles which in turn form pathways and also alter the shape of the environment. Depending on the nature of the study, one or more mice were introduced into such box environments and their behaviour observed. In some cases the behaviour would be recorded using a video camera suspended over the environment. Two stills from video data collected in this manner are presented in Figures 3.1 and 3.2. Both figures show a "single mouse in a box" scenario. Referring to the figures there are two noticeable features in the box; (i) four identical objects in each of the four corners of the box and (ii) markings on the floor and walls of the box. The four identical objects at corners are possible nest sites. The idea was to observe the behaviour of the mouse with respect to nest site selection. The markings indicate locations where chemical scent marking might be placed; for the evaluations presented in this thesis, such experiments were not used, thus the markings could be ignored.

The advantage offered by such laboratory experiments is that the environment can be controlled; thus different kinds of scenario can be created for the purpose of a variety of behaviour studies. The disadvantage is that the number of studies that can be conducted in this way is limited because of the resource required; this limits the scope of (say) "what-if" style experiments.

The work described in this thesis, as noted earlier, utilises a technique known as Multi-Agent Based Simulation (MABS) . The MABS concept was introduced in Chapter 1 and was discussed in further detail in Chapter 2. For the purpose of the work presented

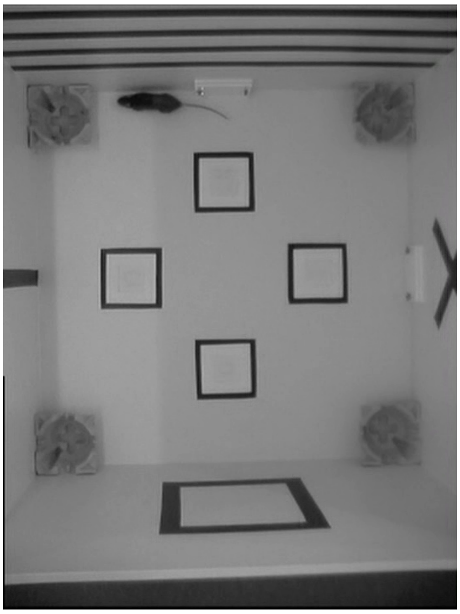


FIGURE 3.1: Mouse Behaviour Video data still, Example 1

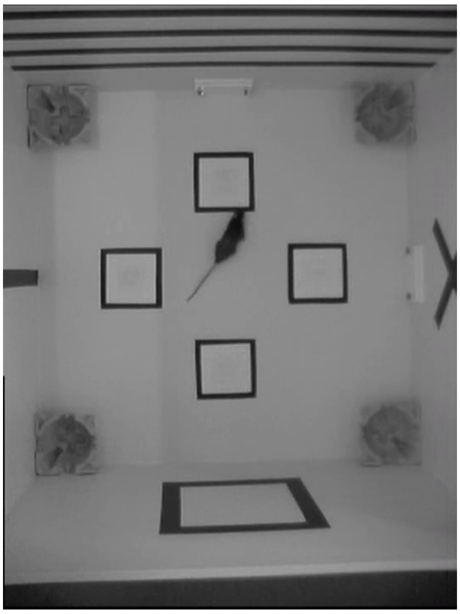


FIGURE 3.2: Mouse Behaviour Video data still, Example 2

in this thesis expert domain knowledge was obtained to provide input to the MBMABS design, knowledge obtained from studies such as the box environment studies described above. The most important aspects of mice behaviour, as highlighted by the domain experts consulted during the course of the programme of work, included the following observations:

1. Where possible mice will prefer to move along the walls of the boxes used in the experiments because they are thigmotaxic (have preference for walls).
2. When placed in a new box environment there is a strong tendency for mice to move around the box and explore the environment.
3. Occasionally mice venture into open space for reasons that are often unclear.
4. Male mice prefer to avoid other male mice which come within close proximity.
5. Male mice defend their territory when necessary.

The above aspects were used to inform the design of the desired Mammalian behaviour MABS so that it operated in as realistic a manner as possible. How this is achieved was the subject of the research presented in the remainder of this thesis. With respect to the above, although all five behaviours are well recognised, there is little published work concerning these individual behaviours with the exception of thigmotaxis (see [122, 123]).

3.4 Harvest Mice

Harvest mice were used as the subject matter for the simulation considered in this thesis because they are regularly used with respect to traditional, real life, behaviour studies; and because they have been well studied [32]. Outside of the laboratories harvest mice are known to be active during both day and night [33]. Behaviourists believe harvest mice are nocturnal in summer and more diurnal in winter. They spend a lot of time underground travelling through tunnels and pathways which they create in their natural environment and prefer areas with predominantly long grass [124]. Harvest mice are natural climbers and feed in “stalk areas” of long grasses and reeds around dusk and dawn where available [35], although in the context of box environment studies such behaviour is difficult to replicate. They prefer seclusion and perceived safety even when feeding. Harvest mice prefer nests above ground and in dense vegetation like grasses [124]. The size of a nest can be around 5cm in diameter for non breeding nests and up to 10cm for breeding nests [124]. Harvest mice use nests for breeding and as safe locations for resting. The location of the nest is normally very carefully chosen and improved upon. Predators include foxes, cats, crows and pheasants [125]. Thigmotaxis (as defined above) is the most prominent, safety related response, exhibited by rodents [122]. Harvest mice are known to exhibit this behaviour [122].

Harvest mice are very timid and extremely cautious [122]. They are usually slow and hesitant when moving. They spend a lot of time examining routes for danger before moving. In the event of danger their first reaction is to stop completely, making no sound or movement. Thereafter they travel quickly, preferably through already known paths, to the closest location which they consider safe. In a natural habitat this would be an underground location. In the event of danger the overall aim is to get to the nest location. When very familiar with specific routes they normally travel relatively fast along them. When moving across unsafe areas, like areas which lack vegetation or feature “open ground”, they tend to move very swiftly.

Scent marks are deposited by many species of animals, including harvest mice; primarily for communication, sexual selection, territorial ownership and social dominance purposes. There is currently ongoing research work relating to scent marking and territorial protection, it is however known that harvest mice use scent marking primarily to indicate territorial ownership.

3.5 Summary

This chapter has presented some background information with respect to the mouse behaviour application domain that acts as a focus for the work presented in this thesis. The chapter commenced with a discussion concerning the study of animal behaviour in general. It was observed that animal behaviour studies are significant with respect to a variety of reasons. Next mouse behaviour simulation was considered and the main factors to be included in any computer simulation of mouse behaviour identified. The particular environment considered with respect to the work presented in this thesis was also considered, namely $1.22m^2$ square box environments populated by one or more harvest mice. In the penultimate section some detail concerning the behaviour of harvest mice in their natural environment, as opposed to a laboratory environment, was considered. In the following chapter, the central concept of the MABS framework, the behaviour graph, will be discussed from an abstract perspective. In later chapters it will be considered from a more application directed perspective.

Chapter 4

Behaviour Graphs and Desires

4.1 Introduction

The behaviour graph concept is the central component of the proposed Mouse Behaviour Multi-Agent Based Simulation (MBMABS) framework presented in this thesis. It is one of the most significant contributions of the work described. This chapter thus presents a complete and in depth review of the behaviour graph concept by describing, in an abstract (generic) manner, the key elements that make up a behaviour graph. In subsequent chapters the utilisation of the behaviour graph concept will be considered in more detail by considering its application in the context of the harvest mouse behaviour application domain that was selected to act as a focus for the work described. The intention of the latter is to illustrate the operation of the behaviour graph concept by example; in this chapter the concept is considered purely from an abstract perspective. This chapter also considers the concept of desires, an important element with respect to the operation of behaviour graphs.

The rest of this chapter is structured as follows. In Section 4.2 the behaviour graph concept is presented. As will become apparent the nodes in behaviour graphs have action and state change methods associated with them, these are discussed in Sections 4.3 and 4.4 respectively. In Section 4.5 the concept of desires is described; the significance is that the nature of desires in many cases influences the state change process. Section 4.6 describes how desires are used to effect state changes. Section 4.7 illustrates the decision making mechanism of a MBMABS agent. Section 4.8 then describes the formulation of the concept of agents within the context of the proposed MBMABS. An overview of the main components of the proposed MABS Framework is then presented in Section 4.9 together with a high level description of the operation of the proposed MBMABS. The validation of the framework, is then presented in Section 4.10. Finally, in Section 4.11, the chapter is concluded with a summary.

4.2 Behaviour Graph

Definition 4.2.1. A behaviour graph is a mechanism that is used to represent/store agent behaviour that lends itself to usage in the context of MABS while at the same time providing for scalability. A behaviour graph comprises a tuple of the form $\langle V, E, L_V, L_E \rangle$ where: (i) $V = \{v_1, v_2, \ldots,\}$ is a set of nodes or vertices each describing a "state" (a "state" is a representation of a current, specific, unique action or behaviour), (ii) $E = \{e_1, e_2, \ldots,\}$ is a set of directed edges describing permitted "state changes", (iii) $L_V = \{l_{v_1}, l_{v_2}, \ldots,\}$ is a set of vertex labels describing individual states and (iv) $L_E = \{l_{e_1}, l_{e_2}, \ldots,\}$ is a set of edge labels describing state changes.

As noted previously a behaviour graph is a sophisticated form of Finite State Machine (FSM), some of the terminology used in this section has therefore been taken from the FSM domain. As noted in definition 4.2.1, a behaviour graph, B comprises a tuple of the form $\langle V, E, L_V, L_E \rangle$ where: (i) $V = \{v_1, v_2, \ldots,\}$ is a set of nodes or vertices each describing a "state" , (ii) $E = \{e_1, e_2, \ldots,\}$ is a set of directed edges describing permitted "state changes", (iii) $L_V = \{l_{v_1}, l_{v_2}, \ldots,\}$ is a set of vertex labels describing individual states and (iv) $L_E = \{l_{e_1}, l_{e_2}, \ldots,\}$ is a set of edge labels describing state changes. There is a mapping of L_V and L_E to V and E[1]. Note that any directed edge $e \in E$ may be either: (i) inward with respect to a reference vertex (state) or (ii) outward with respect to a reference vertex (state). Inward edges indicate that the reference vertex is a possible follow on state with respect to the "from" vertex, while outward edges indicate a possible follow on state with respect to the reference vertex. There can be any number of vertices in a behaviour graph, but as a rule; each vertex or state must have at least one outward edge and one inward edge. The exceptions are the "start" and "end" vertices (states), such vertices have a special meaning. A start vertex describes some start state and does not have any inward edges. An end vertex describes some end state and does not have any outward edges. Note that it is possible that a particular simulation does not have a start and/or end state as described above. For some simulations it might be appropriate to use some other vertex as the start as opposed to a specifically defined "start" vertex. Similarly for many behaviour simulations there may be no specific end goal, so no specific end state, the simulation continues till the user decides to end it. In other cases there may be specific end vertices, for example "eaten by predator".

An agent in a MABS will, conceptually, be located somewhere in a behaviour graph in the sense that it will have a particular state, represented by a vertex $v_i \in V$. A MABS agent can have only one state at any particular simulation time t_i, we refer to this state as the "current state". All agents in a MABS may subscribe to the same behaviour graph or different behaviour graphs, or a group of agents may subscribe to a particular behaviour graph while another group subscribes to a different behaviour graph.

[1]In the literature on graph mining, the fact that a graph has a set of vertices V which can be labelled one label per vertex, using a set of labels L_V, is referred to as a mapping. How the mapping operates is application dependent, there is no generic formal mapping mechanism. This also applies to the "mapping" between the set of edges E and the associated set of labels L_E.

Each vertex in a behaviour graph will have at least two methods associated with it: (i) an *action method* and (ii) a *state change method*. Each is discussed in further detail in the following two sections. Each vertex will also have a set of weightings $W = \{w_1, w_2, \ldots, w_n\}$, the significance of these weightings will become apparent later in this chapter. In some cases a vertex may also include a maximum time T that an agent can be in the given state, again the significance of this value T will become apparent later in Section 4.3 below.

4.3 Action methods

Definition 4.3.1. An action method is a mechanism associated with a vertex $v_i \in V$ in a behaviour graph which is used to implement the functionality associated with the state represented by the vertex (this may of course be "do nothing").

Actions methods are conducted over one or more iterations of the simulation; simulation time $t_i, t_{i+1}, t_{i+2}, \ldots t_e$. An action will typically continue to be conducted until the action goal is achieved or the action "times out" (when simulation time t_e is reached). Thus we can identify two broad categories of action that may be operationalised by action methods:

1. **Goal Driven Actions**: Actions that continue until some end goal is realised. An example of a goal driven action might be "find food".
2. **Timed Actions**: Actions that continue for a number of iterations of the simulation and then cease when some simulation time t_e is arrived at. An example of a timed action is "sleep". (Note that t_e is defined in a probabilistic random manner, thus the duration of a specific timed action will not be the same on each occasion that a simulation is run).

The timing of our process merits some further discussion here. Timing out is concerned with the duration whereby an agent may remain in some state; agents are assumed to be unable to remain in any one particular state indefinitely. Where applicable timing out is implemented using a value p (a field in each agent's definition) that is set to 1.0 when the agent moves into a relevant timed action state (vertex in the behaviour graph). This value is then decremented, according to the definition of a cosine curve (Figure 4.1 where, for convenience, state time is presented in terms of degrees), on each iteration of the simulation. More specifically where relevant, on each simulation iteration, the value p is calculated as shown in Equation 4.1 where: $stateTime$ is the time the agent has been in the current state; and T is the maximum, pre-specified, state time. On each iteration of the simulation a random number r $(0.0 \leq r \leq 1.0)$ is generated. If r is less than p simulation time t_e has been reached and a state change triggered.

From Figure 4.1 it can be seen that at time 0 (degrees) the probability that an agent will remain in its current state is 1.0 (definitely remain), at time 90 (degrees) the probability that an agent will remain in its current state is 0.0 (definitely not remain); thus, as time progresses, the likelihood of a state change increases. The state time associated with the 90 degree value will depend on the nature of the state under consideration. Instead of a cosine probability curve a linear probability curve, or some other alternative, could have been adopted; however, the cosine probability curve has the desirable feature that the likelihood of a state change increases with time.

$$p = cosin\left(\frac{90 \times stateTime}{T}\right) \tag{4.1}$$

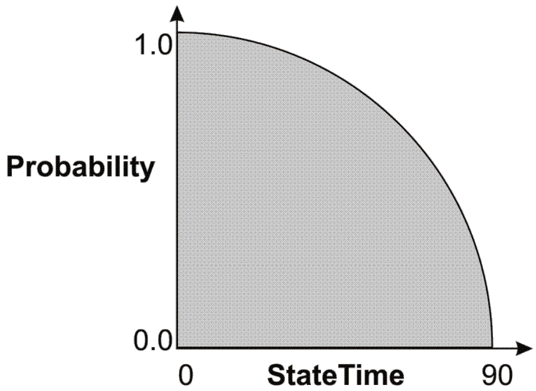


FIGURE 4.1: Cosine Probability Curve

4.4 State Change Methods

Definition 4.4.1. A state change method is a mechanism used to identify a follow on state and undertake any processing required before an identified follow on state can be commenced.

The second method associated with every vertex (state) in a behaviour graph is the state change method. The idea of state changes is central to the operation of the behaviour graph and is therefore discussed in some detail in this section. A state change can be conceptualised in terms of the movement of an agent from one vertex in the behaviour graph to another along a directed edge connecting the two vertices.

A state change is triggered as the result of some event. Two types of event are identified: (i) internal events and (ii) external events (a similar idea was presented in [84], discussed in Chapter 2, where a FSM was used simulate vole behaviour). The distinction is that an internal event is within the control of the agent while an external event is outside of the control of an agent. An internal event is typically the natural completion of some goal driven or timed action (such as “find food" or “sleep”); it is concerned with an agent completing some self appointed task. An external event is some unexpected happening that interrupts (terminates) the current action, for example “predator arrival”.

The state change method’s function is, on completion or termination of an action, to identify a follow on state and undertake any processing required before the follow on state can be commenced. Only certain states follow on from others as specified in the behaviour graph. Follow on states are selected in either: (i) a fixed manner or (ii) a probabilistic random manner. Fixed selection occurs where, as a result of some event, there is only one possible follow on state. Probabilistic state selection occurs where there are a number of competing alternative follow on states, in which case one is chosen in a probability influenced random manner whereby the weightings introduced in Section 4.2 above are used to influence follow on state selections according to the "desire strengths" of an agent. The concept of desires is thus significant with respect to the operation of behaviour graphs and is therefore considered in further detail in the following section, Section 4.5. The process whereby desires are used to effect state changes is then described in Section 4.6.

4.5 Desires

Definition 4.5.1. Desires are "objectives" which agents operating within a MABS framework wish to achieve. The central significance of desires, with respect to behaviour graphs, is that they control the selection of "follow on" states where there are more than one follow on state to select from. An agent has a predefined set of n desires $D = \{d_1, d_2, \ldots, d_n\}$ where each $d_i \in D$ is a numeric quantity of between 0 and 1. We refer to this numeric quantity as the desire "strength".

There are several factors that indirectly influence the behaviour of agents that may be "contained" within a behaviour graph. We have already noted that one category of factor is the occurrence of some event (internal or external); a second category is the concept of desires. A desire in this context is a mechanism for representing the motivation behind the actions of MBMABS agents, a similar concept is used in connection with BDI frameworks as discussed in Chapter 2 . Broadly a desire is a wish for some resource or knowledge. Examples include a desire for sustenance or sleep, or a desire to explore or find a place of safety (such as a nest site). The central significance of desires, with respect to behaviour graphs is that they control the selection of "follow on" states where there is more than one follow on state to select from. The nature of desires and the mechanism whereby they are utilised to select follow on states is thus presented in this section.

As noted above in definition 4.5.1, an agent has a predefined set of n desires $D = \{d_1, d_2, \ldots, d_n\}$ where each $d_i \in D$ is a desire "strength". We can identify two broad categories of desire: (i) static and (ii) dynamic. Each is discussed in further detail in the following two sub-sections, Sub-sections 4.5.1 and 4.5.2. How desires are used to select follow on states depends on the nature of the follow on state. The mechanism whereby the concept of desires is applied in the context of state changes is discussed in further detail in Section 4.6 below once the ideas supporting static and dynamic desires have been established.

4.5.1 Static Desires

Definition 4.5.2. A static desire is one whose "strength" remains fixed throughout a simulation.

Static desires are not affected by changes in the environment such as the completion of self appointed tasks (internal events) or the occurrence of unexpected happenings (external events). In practice most desires are dynamic. It is difficult to give many realistic examples of a static desire, but for completeness the behaviour graph mechanism should support the idea of static desires. In the case of harvest mouse simulation thigmotaxis is an example of a static desire.

4.5.2 Dynamic Desires

Definition 4.5.3. A dynamic desire is one whose "strength" changes with time.

The character of a particular dynamic desire at a given simulation time t_i can be increasing, decreasing or constant. The increase or decrease in desire strength can be either: (i) gradual or (ii) sharp. We model the gradual increasing and decreasing aspects of desire strengths using a cosine curve (as also used to implement the idea of timed actions as described above). A sharp increase or decrease in a desire, in turn, can be thought of as a sudden jump in the strength of a desire typically associated with some event, for example the detection of a predator may trigger a sudden increase in the desire for a safe location. As the result of some event a dynamic desire can also either come into being (emerge) or cease to exist (disappear). A desire that ceases to exist has its strength set to 0, a desire that comes into existence has its strength changed from 0 to some non-zero value. In fact a desire never really ceases to exist, it simply becomes dormant; an agent has a constant set of desires D throughout a simulation. Figure 4.2 presents two examples of how the strength associated with dynamic desires can change overtime. The figure shows two desires (top and bottom) associated with either the same agent or two different agents. In the figures the time stamps $t_1, \ldots, t_5$ are associated with events. In the case of the "top desire" the strength is fixed till time t_1 when it becomes decreasing, disappearing at time t_2. At time t_3 it re-emerges, jumping at time t_4 and then disappearing again at t_5. In the case of the "bottom desire", the strength is fixed from t_0 to t_2, although it features a downward jump at t_1; it disappears at t_2. Later, at t_3, it reappears, decreasing at t_4, and disappearing at t_5.

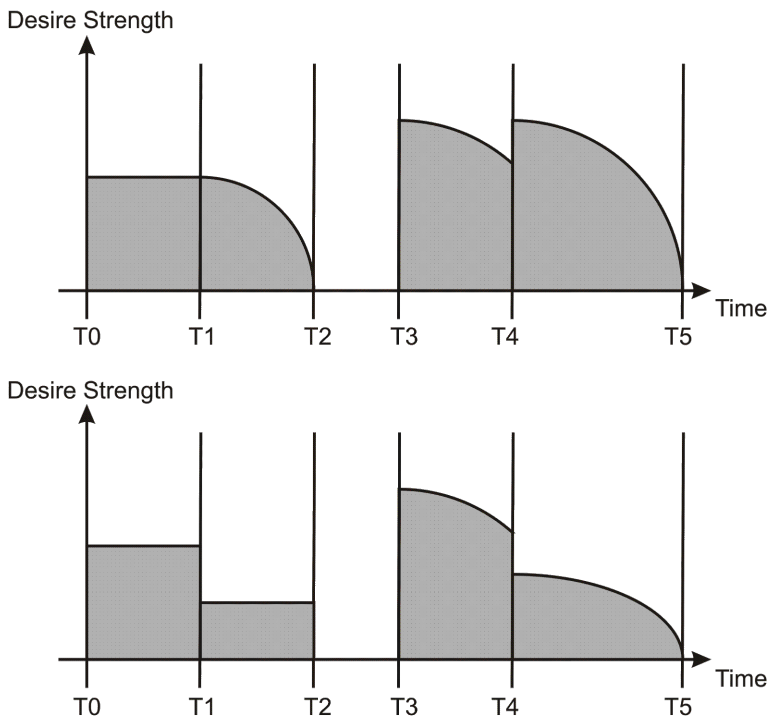


FIGURE 4.2: Examples of agent desires

4.5.3 The Naturalness of Animal Desires

In real world situations, desires can be considered to be a "state of mind" whereby an individual has a personal motivation to perform an action or a set of actions in order to achieve a goal [126]; the objective or aim which an individual has elected to complete. With respect to the role of desires in human behaviour, there is a strong conviction that desires direct the selection of goals [127, 128]. Real world desires play the same kind of roles in animals; for example animals have a desires for food and safety which influences the goals they wish to achieve and the consequent activities (actions) they perform [129]. This means that desires play a significant role in behavioural control because they influence fundamental aspects of behaviour; namely decision making, which in turn leads to a choice of actions to perform. Desires are an important component for explaining an individual's decision making [126].

From the foregoing the concept of desires is an important component of the proposed MBMABS framework where they are conceptualised according to a generic idea of the function of desires. As noted above, desires are quantified according on their "strength", which may either: (i) increase (for example when hungry a desire for food is expected to increase until the desire is satisfied), (ii) decrease (for example when a desire for food has been addressed) or (iii) remain constant (as in the case of thigmotaxis). Also noteworthy is the idea that several desires are competing for dominance, and when one diminishes, another may increase, thus a particular behaviour can occur as a result of a combination of several desires.

4.6 Using Desires to Affect State Changes

Definition 4.6.1. A preference is a value which defines a character agent's inclination towards a particular state. Suppose that we have a current state with a set F of m follow on states: $F = \{f_1, f_2, \ldots, f_m\}$, for each state f_i we calculate a preference value p_i so that we have a set of preferences $P = \{p_1, p_2, \ldots, p_m\}$ such that $|P| \equiv |F|$ and $\sum_{i=1}^{i=m} p_i \in |P| \equiv 1.0$. A preference is derived using a combination of the elements in a set of desires, D and their associated weightings from the set of weightings, W. There is a one-to-one correspondence between the elements in the set D and the elements in the set W ($|D| \equiv |W|$). Thus $W = \{w_1, w_2, \ldots, w_{|D|}\}$, where each element w_i has a value between 0 and 1.

Having established the idea of desires this section describes how they are use to effect state changes. In other words this section describes the operation of the state change methods introduced in Section 4.4. Typically an agent has several competing desires of different strengths at a given time point in a simulation; given a change state situation these may have to be reconciled with a number of follow on states (if there is only one follow on state there is no issue). Different desires are associated more strongly with particular follow on states. For example a follow on state to find food will be closely linked with a desire to satisfy hunger. Recall that in Section 4.2 it was noted that each vertex has a set of weightings W associated with it. The significance of desires with respect to a particular follow on state is expressed in terms of these weightings; the weightings are used to express the relevance of a particular follow on state with respect to each individual desire. In some cases the relevance may be very low; for example a foraging for food state will be of little relevance with respect to a desire for (say) sleep, whilst it will be of considerable significance with respect to a desire to (say) satisfy hunger. As noted in definition 4.6.1, there is a one-to-one correspondence between the elements in the set D and the elements in the set W ($|D| \equiv |W|$). Thus $W = \{w_1, w_2, \ldots, w_{|D|}\}$, where each element w_i has a value between 0 and 1.

The mechanism whereby desires and weightings are used to effect a state change can best be described by considering an abstract example. It was supposed from definition 4.6.1, that we have a current state with a set F of m follow on states: $F = \{f_1, f_2, \ldots, f_m\}$. For each state f_i we calculate a preference value p_i so that we have a set of preferences $P = \{p_1, p_2, \ldots, p_m\}$ such that $|P| \equiv |F|$ and $\sum_{i=1}^{i=m} p_i \in |P| \equiv 1.0$. Each value p_i is calculated as shown in equation 4.2.

$$p_i = \sum_{i=1}^{i=|D|} d_i \times w_i \tag{4.2}$$

Once the preferences have been calculated the follow on state is not simply selected according to the highest preferences value. Instead, because we want to introduce an element of randomness, the preferences are used to weight a random selection. Thus a follow on state with a high p value is more likely to be selected, but is not guaranteed to be selected. We can conceptualise the process in terms of a number line from 0 to 1. We range the follow on states along the number line so that the number line is divided into m parts each of a length dictated by the associated preferences value. We then generate a random number r between 0 and 1, plot this on the number line and select the relevant follow on state.

Algorithm 1: Determining preference for behaviour graph nodes

Input: $F = \{f_1, f_2, f_3 ... f_m\}$, set of follow on states
Input: $D = \{d_1, d_2, \dots\}$
Output: b' = the selected follow on node from F
1 b' = Variable in which to store identified follow on state;
2 P = Set of preferences of size $|F|$;
3 **for** $i = 0$ *to* $i = |F|$ **do**
4     $p_i = 0$
5   **for** $j = 0$ *to* $j = |D|$ **do**
6         $p_i = p_i + (d_j \times w_j)$, $(w_j \in |W|)$;
7         $P = P \cup p_i$;
8   **end**
9 **end**
10 $sum = p_1$;
11 r = random number $(0 \leq r \leq 1)$;
12 **for** $i = 0$ *to* $i = |P|$ **do**
13   **if** $r < sum$ **then**
14         $b' = b_i$;
15   break;
16   **else**
17         $sum = sum + p_i$;
18   **end**
19 **end**
20 **return** b'

Algorithm 1 describes, in a procedural manner, the mechanism whereby an agent selects a new activity from a set of possible activities. The input is a potential set of follow on nodes F and a set of desires D with respect to the current agent. The output is the identifier for the selected follow on node b'. We commence line 2 by initialising a set P of size $|F|$ in which to hold the preference values for each follow on state. Then, for each follow on state f_i the preference value p_i is calculated (lines 4 to 8). In lines 9-20, the selection of a follow on state, b_i is determined in a randomised probability driven manner using a number line, and a random number, r between 0 and 1. Thus the follow on state with the highest preference, is not always chosen as the selected follow on state.

Three illustrations of the above process are presented below using the fragment of behaviour graph given in Figure 4.3 comprised of three states: “start”, “work” and “sleep”. The current state of the agent under consideration is the “start” state. The “start” state has two follow on states: “work” and “sleep”. The set of desires D for the agent in question is assumed to be $D = \{d_w, d_s\}$ where d_w is the desire to go to work, and d_s is the desire to sleep. Thus each state has a set W corresponding to the agents set of desires D, we will indicate the weightings for the work and sleep states using the notation $W_w = \{W_{w_w}, W_{w_s}\}$ and $W_s = \{W_{s_w}, W_{s_s}\}$ respectively, and assume $W_w = \{1.0, 0.0\}$ and $W_s = \{0.0, 1.0\}$ (the ordering corresponds to the ordering in which desires are listed).

Example 1: Given $D = \{d_w, d_s\} = \{1.0, 0.0\}$: the preference values associated with the state “work” and state “sleep”, p_w and p_s, will be calculated as follows:

$$\begin{aligned} p_w &= d_w \times w_{w_w} + d_s \times w_{w_s} = 1.0 \times 1.0 + 0.0 \times 0.0 = 1.0 \\ p_s &= d_w \times w_{s_w} + d_s \times w_{s_s} = 1.0 \times 0.0 + 0.0 \times 1.0 = 0.0 \end{aligned} \tag{4.3}$$

In this case because the desire for sleep is zero the “work” state will be selected.

Example 2: Given $D = \{0.0, 1.0\}$:

$$\begin{aligned} p_w &= d_w \times w_{w_w} + d_s \times w_{w_s} = 0.0 \times 1.0 + 1.0 \times 0.0 = 0.0 \\ p_s &= d_w \times w_{s_w} + d_s \times w_{s_s} = 0.0 \times 0.0 + 1.0 \times 1.0 = 1.0 \end{aligned} \tag{4.4}$$

In this case because the desire for work is zero the “sleep” state will be selected; in practice desires are unlikely to be zero as shown in the next example.

Example 3: Given $D = \{0.5, 0.5\}$:

$$\begin{aligned} p_w &= d_w \times w_{w_w} + d_s \times w_{w_s} = 0.5 \times 1.0 + 0.5 \times 0.0 = 0.5 \\ p_s &= d_w \times w_{s_w} + d_s \times w_{s_s} = 0.5 \times 0.0 + 0.5 \times 1.0 = 0.5 \end{aligned} \tag{4.5}$$

In this case a selection will be made according to the “number line” process described above.

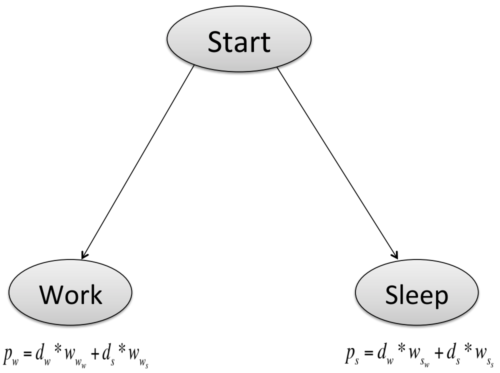


FIGURE 4.3: Illustration of the effect of desires on state changes

4.7 Selection of Direction of Travel

The actions that can be performed by individual agents utilising the behaviour graph concept often entail the agent moving from one location to another. This in turn entails the selection of direction of travel. Because this is a frequent occurrence the ideas supporting the proposed mechanism for direction of travel selection are presented in this section, further detail is provided later in the thesis. In some cases there may be only one direction of travel, in other cases there may be a number of alternatives. An agent placed in an environment has freedom of movement in n directions. The nature of n will depend on the mechanism used to model the environment in which character agents operate in the context of the simulation of interest. With respect to the proposed MBMABS framework a "tile world" environment model was adopted, this is discussed in detail in Chapter 5. The value for n in this case was therefore defined in terms of the number of immediate neighbouring tiles into which an agent can legally move. Given a tile world environment there are thus a maximum of 8 possible neighbouring tiles that an agent can move to, the set L of possible tile locations: $L = \{l_1, l_2, \ldots, l_n\}$ where $0 \leq n \leq 8$. Each tile will be assigned a code, a Ground Type Indicator (GTI), indicating what kind of location the tile represents. Table 4.1 gives some examples. The "Label" column gives the code assigned to a particular kind of tile, the "Name" column specifies the tile type, and the "Description" column specifies the features of the tile. For example, the tile encoding "O", denotes an Open Space tile. Using the agent's desires and tile encodings a set of matched probabilities $\{p_1, p_2, \ldots, p_n\}$ are derived and used to direct movement. An example of how this is done is given in Chapter 5 once further detail of the tile world used to model environments within the context of the proposed MBMABS has been discussed.

Table 4.1: Ground Type Identifiers (GTIs)

Label	Name	Description
N	No-go location	Tile that represents a location that cannot be reached by an agent, because it represents an obstruction.
W	Wall location	Tile that represents a location that cannot be reached by an agent, because it is too close to a wall.
C	Choice points	Tile location where we wish change direction should be considered, e.g a corner location.
T	Tunnel location	Tile location which is within a tunnel, or location where movement is restricted.
O	Open Space	Tile location which is not any of the above.

4.8 MBMABS Character Agents

Definition 4.8.1. An MBMABS character agent is an agent which uses the behaviour graph and fundamentally, has attributes which must include its own set of desires.

The nature of the individual agents that make use of a behaviour graph, generically referred to as a *character agents*, are considered in this section. We use the term character agent to distinguish between agents that use the behaviour graph and other types of agents, representing (say) obstructions, that might feature in a MABS environment (such alternative types of agent are considered in further detail later in this thesis). Additionally the operation of agents, as it relates to the behaviour graph, will be presented in this section.

Each character agent has a number of basic attributes and methods as shown in the class diagram presented in Figure 4.4 which list six attributes and three methods. From Figure 4.4 it can be seen that a character agent will feature at least the following attributes:

1. A set of desires, D (as described above).

2. A location within the environment (conceptually it will also have a location on the behaviour graph). A character agent's location is expressed in terms of x-y coordinates referenced to the origin on the environment in which it will operate.

3. A current state, defined by a vertex in the behaviour graph associated with the character agent.

4. A "stateTime" (recall that the usage of this attribute was concerned with timed actions as presented in Section 4.3 above).

5. A direction in which it is facing or travelling in (expressed in terms of the four cardinal (north, south, east, west) and four inter-cardinal (North West, North East, South West and South East) directions.

6. An identification number.

Additional attributes that a character agent might possess will depend on the nature of the application simulation domain.

As noted above sometimes a character agent may possess other attributes, depending on the application domain the character agent is to be used in. An example with respect to this thesis is the scent attribute associated with mouse agents (as demonstrated in Chapter 6).

It is well established that when a mouse moves it leaves scent markings as a way of demarcating its territory; this plays an important role with respect to intersexual relationships and to deter territorial intruders. Thus in the case of the mouse simulation application used as the focus for the work described in this thesis, the agents have an additional scent attribute. The idea is that the mouse agents in a simulation leaves a scent trail at locations of its choosing or along routes it chooses as it moves around its environment. For this to happen each location within the environment has a record of any scent at that location as well as the identification number of the mouse agent which has left that scent. Scent is measured using a scent strength variable (str). Scent strength is defined as an integer; different mouse agents have different scent strengths according to their "dominance". Scent "degrades" with time, so on each iteration of a simulation, scent strength is degraded by *degradation factor*. The degradation factor (df) is a global parameter specified for each simulation run, it is a numeric value of between 0 and 1.

Referring back to Figure 4.4 character agent will also feature the following three methods:

1. **selectNextState**: A method to allow the selection of follow on states as described above (note that this will include a mechanism allocating preferences to follow on states).

2. **adjustDesires**: A method to adjust the set of desires D on each iteration of the simulation.

3. **selectNextDirectionOfTravel**: A method to select a direction of travel where appropriate.

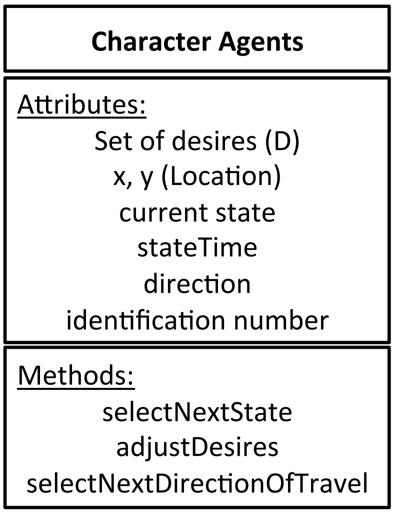


Figure 4.4: Generic class diagram for a MBMABS character agents

4.9 The MABS Framework

In this section an overview of the generic MABS Framework is described, as a precursor to the more detailed discussion of it as presented later in this thesis (specifically Chapter 5). The idea is that the proposed MABS framework will allow the development of behaviour simulations in an extensible manner. More specifically the MABS framework has been designed to provide a generic simulation facility that allows the inclusion of a range of desires and behaviours. Once the framework has been populated with desires and associated behaviours the simulation can be run. The simulation operates on an iterative basis. On each iteration agents either perform some action according to their current "state" (for example move a certain distance in some direction) or undertake a state change. State changes happen instantly (for example an agent might go from a moving state to a stopped state in a single iteration).

The fundamental components which make up the MABS framework are as follows:

1. The behaviour graph.
2. Events.
3. Character agents together with their desires.
4. Static agents.
5. Housekeeping agents.
6. Utility agents
7. Simulator interface.
8. Visualisation.

Some of these components have already been discussed in detail. The concept of the behaviour graph, together with the nature of events and states, was described in detail in Section 4.2. The idea of desires was described in Section 4.5.

Four different types of agent are identified:

1. Character agents
2. Static agents
3. Housekeeping agents
4. Utility agents

The notion of character agents was presented in section 4.8. The distinction between character agents and static agents is that character agents display some reasoning capability whilst static agents do not. The most significant kind of static agent, with respect to the simulations considered in this thesis, is the environment agent (this is discussed further in Chapter 5 where a number of example environment agents are presented). Housekeeping agents facilitate the operation of the framework; they assist in operations such as simulating external events, and monitoring the progress of the simulation. Utility agents facilitate the operation of agents within the MBMABS framework by supporting the completion of specialized tasks. Housekeeping and Utility agents are both discussed in further detail in Chapter 5.

The simulation interface enables the end user to interact with the system and to setup specific simulations. With respect to the implementation of the MBMABS presented later in this thesis a menu driven interface was adopted. An important element of the proposed interface is the output from the simulation as it progresses in the form of a visualisation. The visualisation component allows the end user to observe a specified simulation in a real time.

The operation of the proposed MBMABS framework is presented in Algorithm 2. The input to the algorithm is a set of character agents A and a behaviour graph B. We indicate the fields belonging to agent $a_i \in A$ using dot notation; for example an agent's current state will indicated by $a_i.state$. This Algorithm (Algorithm 2) describes in a procedural manner, operation of the proposed framework. The input is a set of agents, $|A|$, each with a set of desire, $a_i.D$. Lines 2 to 4 initialise the default start state for all the agents. Lines 5 to 6 terminate an agent based on some exit condition, such as a terminal state. Lines 9 to 11 describe the current activity of the agent in the simulation to be the moving state. Lines 13 to 15 show how an external event causes the desires an agent, $a_i.D$ to be updated leading to the selection of a new state. Lines 16 to 18 describe a change in state influenced by timing out. Lines 19 to 20 update desires of an agent based on internal events.

Algorithm 2: Determining how agent performs node changes

Input: $A = \{a_1, a_2, a_3 ... a_n\}$, set of agents with set of desires

Output: $a_i.newstate$, new state, selected by an agent

```
forall the a_i in |A| do
    a_i.state = start (gate);
end
Loop;
if exit condition then
    break;
end
forall the a_i in |A| do
    if moving then
        update location;
    end
    if external event then
        update a_i.D with respect to external event;
        select a_i.newstate;
    else if timed out then
        select a_i.newstate;
    else
        update a_i.D set of desires for a;
    end
end
End Loop;
```

4.10 Validation

This section presents and discusses the outcomes of the validation conducted to appraise the performance of the behaviour graph concept in its generic form, as presented above, by assessing its scalability. The objectives of the validation were as follows:

1. To check the impact of increasing the maximum number of inward edges on the performance of the behaviour graph.
2. To assess the impact of increasing the maximum number of outward edges, on the performance of the behaviour graph.
3. To test the impact of increasing the number of nodes/vertices on the performance of the behaviour graph.

4. To check the impact of increasing the number of agents on the performance of the behaviour graph.

With respect to each of the above objectives the validation was conducted by randomly generating twenty behaviour graphs using the following parameters:

1. Max_{Out}: The maximum number of outward edges that a behaviour graph can have. Note that the minimum number of outward edges is 1 (except in the case of an end node which has no outward edges).
2. Max_{In}: The maximum number of inward edges that a behaviour graph can have. Note that the minimum number of inward edges is 1 (except in the case of a start node which has no inward edges).
3. $|V|$: The number of states (vertices/nodes) in the behaviour graph.

The other important settings for the experiment include the following:

1. The weightings, W, as discussed in Section 4.6 which were assigned randomly to states with respect to each simulation experiment.
2. A set of desires, D comprising three desires, assigned to each agent namely; (i) an increasing dynamic desire (idd), (ii) a reducing dynamic (rdd) desire and (iii) a constant desire (cd). This was to represent the three types of desires in the MABS framework. The following start up values for each desire was used: (i) $idd = 0.2$, because the desire strength is designed to increase during the simulation. (ii) $rdd = 0.7$, because the desire strength of rdd, is designed to decrease during the simulation. (iii) cd was chosen to be 0.1.

All the experiments were conducted on an Intel Core $i5$ $iMac$ with a processor of 2.7 GHz and 8 GB 1600 MHz DDR3 RAM. The validation was conducted in terms of simulation time (seconds).

Each of the above listed validation objectives is considered in more detail with respect to the results obtained in the following four subsections.

4.10.1 Effect of Changing the Maximum Number of Inward Edges (Max_{In})

To determine the effect of increasing the maximum number of inward edges on the operation of the behaviour graph the maximum number of outward edges and number of vertices was kept static at $Max_{Out} = 2$ and $V = 100$. A sequence of values for Max_{In} was considered from 2 to 10 increasing in steps of 2. A range of different numbers of agents was also considered from 100 to 1600 increasing in steps of 500. The results are presented in Table 4.2 and illustrated in Figure 4.5.

From Table 4.2 and Figure 4.5, the effect of changing the maximum number of inward edges (Max_{In}) on the Simulation Operation Time was not very significant; indicating that increasing Max_{In} does not significantly affect the performance of the behaviour graph.

However, inspection of Table 4.2 and Figure 4.5 indicates that as the Max_{In} value was increased, the simulation became more efficient. This was because agents were able to move around the behaviour graph in a more efficient manner as Max_{In} increased.

TABLE 4.2: Effect of changing the maximum number of inward edges (Max_{Out} =2, $|V| = 100$) in terms of simulation time

	Max_In				
Number of Agents	2	4	6	8	10
100	0.089	0.068	0.071	0.071	0.069
600	0.165	0.133	0.135	0.126	0.119
1100	0.194	0.173	0.178	0.173	0.170
1600	0.232	0.214	0.226	0.211	0.209

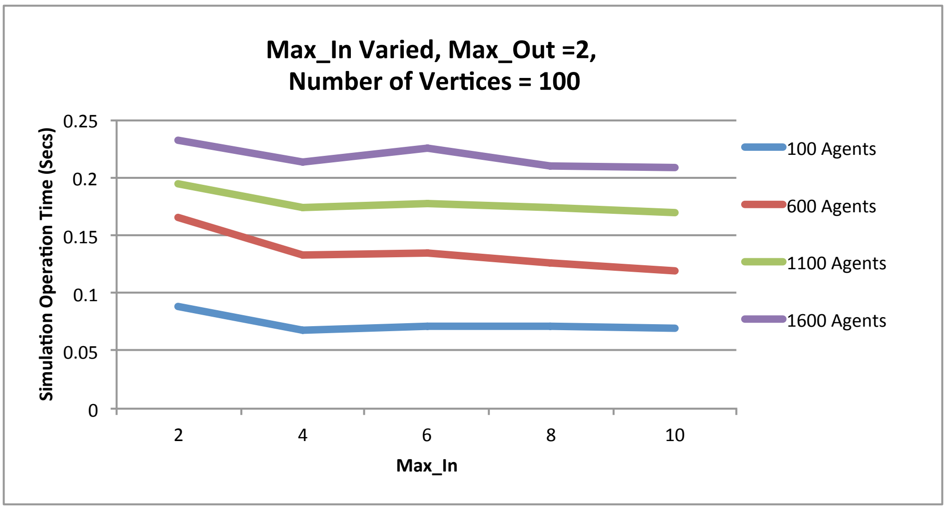


FIGURE 4.5: Visualisation of results presented in Table 4.2

4.10.2 Effect of Changing the Maximum Number of Outward Edges (Max_{Out})

To determine the effect of increasing the maximum number of outward edges on the operation of the behaviour graph the maximum number of inward edges and number of vertices was kept static at $Max_{In} = 2$ and $V = 100$. A sequence of values for Max_{Out} was considered from 2 to 10 increasing in steps of 2. A range of different numbers of agents was again considered from 100 to 1600 increasing in steps of 500. The results are presented in Table 4.3 and illustrated in Figure 4.6.

From Table 4.3 and Figure 4.6, it can be again observed that the effect of changing the maximum number of outward edges(Max_{Out}) on the performance of the behaviour graph is not very significant.

From inspection of the table and figure, it can be seen that as the Max_{Out} parameter was increased the simulation becomes more efficient. A similar situation was observed with respect to the Max_{In} parameter. This was because agents were able to move around the behaviour graph in a more efficient manner as Max_{out} increased.

TABLE 4.3: Effect of changing the minimum number of outward edges ($Max_{In} = 2$, $|V| = 100$) in terms of simulation time

	Max_Out				
Number of Agents	2	4	6	8	10
100	0.086	0.081	0.065	0.071	0.069
600	0.147	0.118	0.109	0.118	0.118
1100	0.214	0.187	0.164	0.175	0.175
1600	0.243	0.238	0.211	0.221	0.223

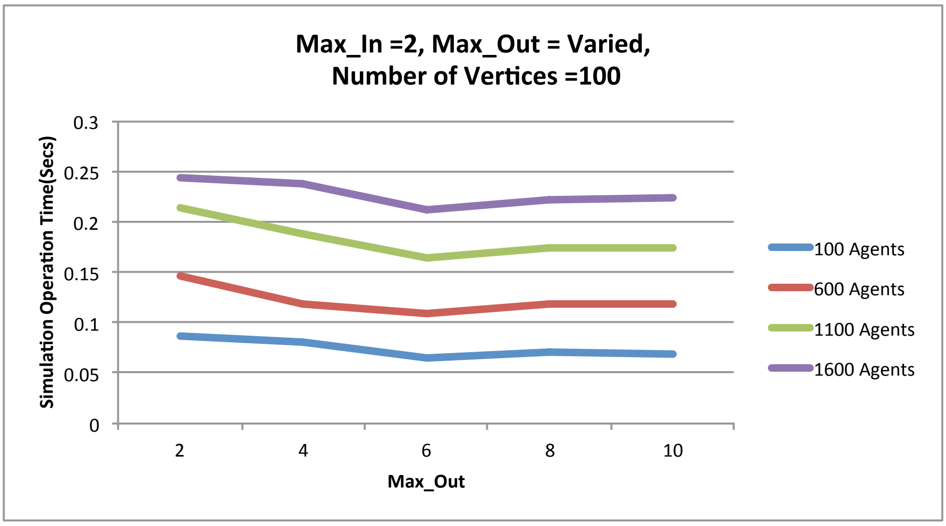


FIGURE 4.6: Visualisation of results presented in Table 4.3

4.10.3 Effect of Changing the Number of Vertices ($|V|$)

To determine the effect of increasing the number of vertices on the operation of the behaviour graph the maximum number of inward edges and outward edges was kept static at $Max_{In} = 3$ and $Max_{Out} = 3$. A sequence of values for V was considered from 100 to 900 increasing in steps of 200. As in the case of the previously reported experiments a range of different numbers of agents was also considered from 100 to 1600 increasing in steps of 500. The results are presented in Table 4.4 and illustrated in Figure 4.7.

From Table 4.4 and Figure 4.7, it can be seen that the effect of increasing the number of vertices is very significant, there is a steady increase in the Simulation Operation Time. An increase in the number of agents, relative to the increase in the number of vertices, also affects the performance. The reason for this is that more system resources were required to process the increasing number of vertices (and agents). The selection of the values for Max_{In} and Max_{Out} also has an effect on efficiency. As can be noted from previous experiments, Figures 4.5, and 4.6, simulation operation time was highest when $Max_{In} = Max_{Out}$.

TABLE 4.4: Effect of changing the number of Vertices($Max_{In} = 3$, $Max_{Out} = 3$, Number of Agents varied) in terms of simulation time

	Number of Vertices				
Number of Agents	100	300	500	700	900
100	0.077	0.458	0.807	1.147	1.639
600	0.125	0.508	0.839	1.273	1.798
1100	0.168	0.512	0.877	1.356	1.910
1600	0.234	0.580	0.899	1.389	1.994

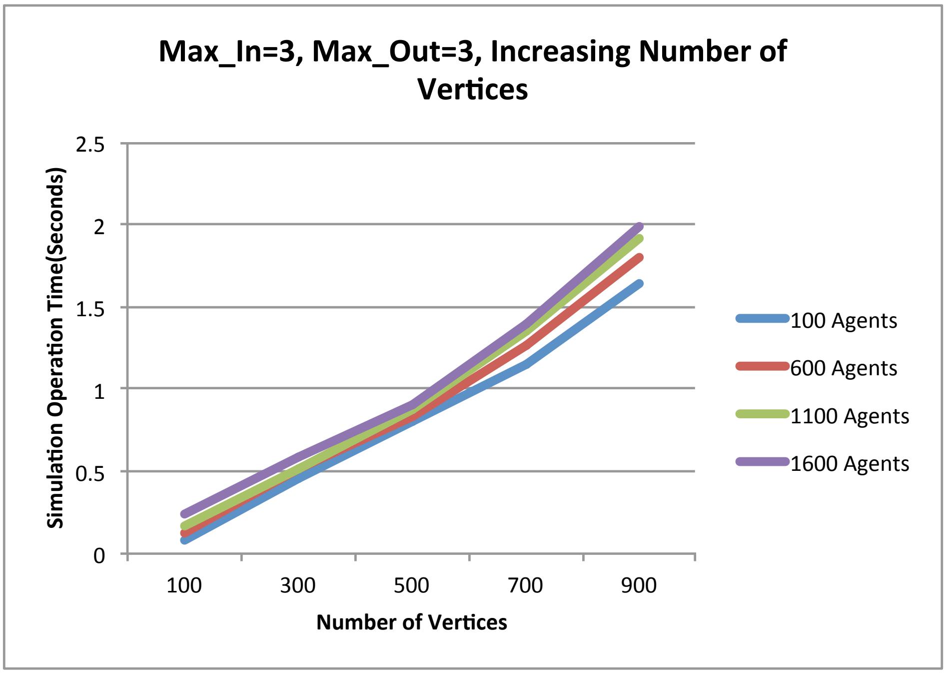


FIGURE 4.7: Visualisation of results presented in Table 4.4

4.10.4 Effect of Changing the Number of Agents ($|A|$)

To determine the effect of increasing the number of agents on the operation of the behaviour graph the maximum number of inward edges and outward edges was kept static at $Max_{In} = 3$ and $Max_{Out} = 3$. A sequence of different numbers of agents was considered from 1000 to 9000 increasing in steps of 2000. A range of values for V was also considered from 100 to 900 increasing in steps of 200. The results are presented in Table 4.5 and illustrated in Figure 4.8.

From Table 4.5 and Figure 4.8 the effect of increasing the number of agents was that the Simulation Operation Time steadily increased. This was to be expected because more system resources will be required to process the operation of the increasing number of agents on the behaviour graph. The selection of the values for Max_{In} and Max_{Out} also has an effect on efficiency. As can be noted from previous experiments, Figures4.5, and 4.6, simulation operation time was highest when $Max_{In} = Max_{Out}$.

Table 4.5: Effect of changing the number of agents ($Max_{In} = 3$,$Max_{Out} = 3$, Number of Vertices varied) in terms of simulation time

	Number of Agents				
Number Of Vertices	1000	3000	5000	7000	9000
100	0.385	0.457	0.581	0.719	0.888
300	0.645	0.788	0.888	1.073	1.234
500	0.898	1.019	1.289	1.444	1.658
700	1.144	1.418	1.565	1.815	1.942
900	1.591	1.728	1.870	2.121	2.497

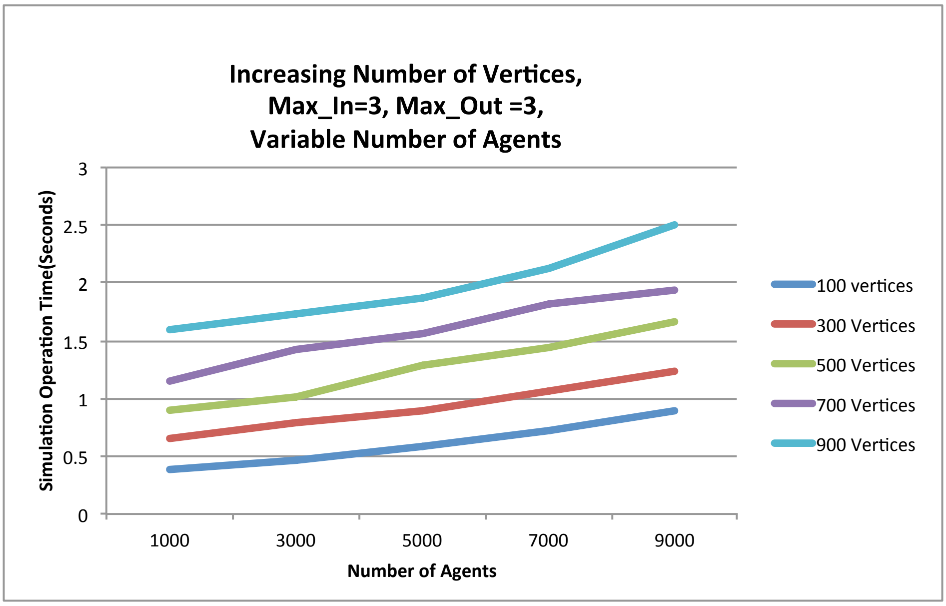


Figure 4.8: Visualisation of results presented in Table 4.5

4.11 Summary

In this chapter the concept of the generic behaviour graph has been described in detail. The role of desires was also described and shown to be a fundamental part of the operation of the graphs. A validation of the behaviour graph concept was also reported on to assess its operation and scalability. The objectives of the validation were to assess the impact on the performance of the behaviour graph with respect to: (i) increasing the maximum number of inward edges, (ii) increasing the maximum number of outward edges, (iii) increasing the number of behaviour graph nodes/vertices and (iv) increasing the number of agents. The validation indicated that, as might be expected, as the number of agents increased, the performance of the behaviour graph was affected in the sense that it required more resources to operate. In the following chapters the behaviour graph is considered in terms of the animal behaviour application that acts as focus for the work presented in this thesis.

Chapter 5

The Mammalian Behaviour Framework

5.1 Introduction

This chapter describes how the abstract MABS framework presented in the previous chapter (Chapter 4) can be used to realise a mammalian (mouse) behaviour MABS directed at harvest mouse behaviour simulation, the application domain used as a focus for the work presented in this thesis. The resulting framework is referred to as the Mammalian Behaviour MABS (MBMABS) Framework. In the context of the MBMABS presented in this chapter both character and static agents were used. Character agents, as noted in the previous chapter, display some reasoning ability, while static agents have no such ability, they are simply objects within the MABS. Two types of static agent were used with respect to the MBMABS: (i) environment agents (only one of these) and (ii) obstruction agents. The nature of this configuration is illustrated in Figure 5.1. Note that the configuration is conceptualized in the form of a “cloud” in which agents exist along with other elements that support aspects of the mechanism. The figure includes N character agents, M obstruction agents and a single environment agent.

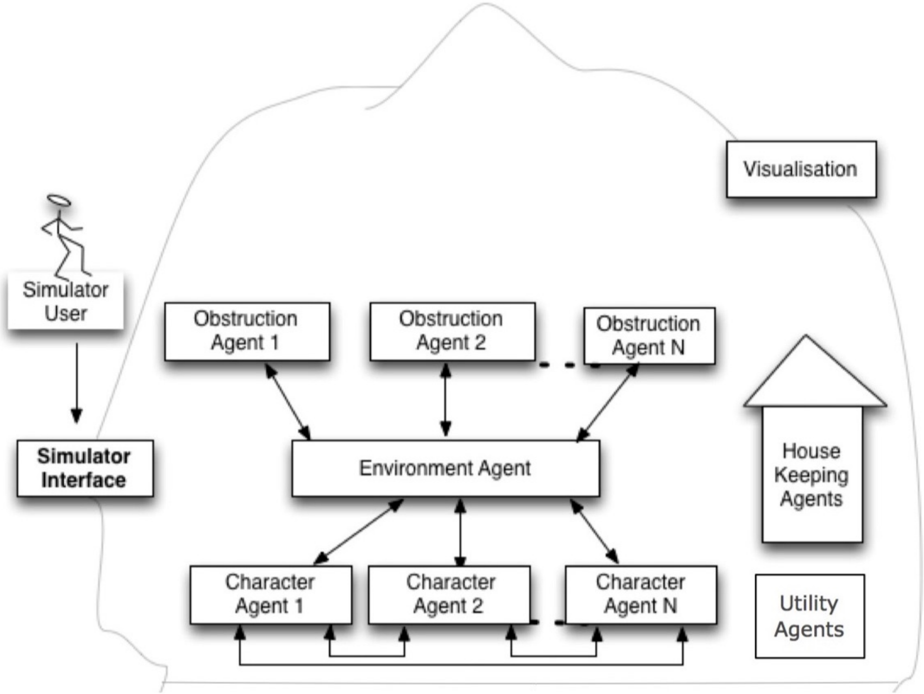


FIGURE 5.1: Components of the Mammalian Behaviour MABS Framework

For most simulations only a single environment agent will be required although in some cases more than one may be appropriate. One example where multiple simulation environments were used with respect to the work presented in this thesis was for the purpose of conducting experiments (reported on later) involving four parallel environments. The aim here was to confirm that the proposed MBMABS framework incorporates a degree of randomness and that consequently the character agent behaviour associated with a number of parallel runs of the same scenario will not be identical.

The other components shown in Figure 5.1 are: (i) a housekeeping agent (designed to provide "administrative" support with respect to the operation of the MABS),(ii) a utility agent to support specialised tasks necessary for the operation of the MABS (iii) the simulation interface with which the end user can setup individual simulations and (iv) a visualization component that enables the user to observe simulations.

The rest of this chapter is structured as follows. The nature of the environment, character and obstruction agents, in the context of the mammalian behaviour MABS, are discussed in further detail in Sections 5.2, 5.3 and 5.4 respectively. The functions of the house keeping and utility agents used to support the operation of the MBMABS framework are briefly discussed in Section 5.5, and 5.6. The simulation interface is briefly described in Section 5.7. Some discussion on the temporal considerations of the framework is provided in Section 5.8. The chapter is concluded with a brief summary in Section 5.9.

5.2 The MBMABS Environment Agent

Conceptually, as noted in chapter 4, the environment in which character agents exist is also regarded as an agent (a static agent). As noted previously the environment is defined in terms of a bounded "tile world" comprising a set of tiles E and measuring $w \times h$ tiles. To indicate a particular tile e in E located at x-coordinate i and y-coordinate j we use the notation e_{ij}. Each tile e, except at the boundary of the environment, has eight neighbours. For the MBMABS framework each tile equated to an area 8×8 cm. This size was selected because it equates approximately to the length of a Harvest Mouse. For the evaluations presented later in this thesis, and as noted previously, "mouse in a box" case studies were used because these could easily be evaluated in the context of real life experiments. The boxes in question measure 122×122 cm; thus our tile worlds comprised 15×15 tiles ($122/8 = 15.25$).

In a single simulation iteration, and in the absence of any obstructions, a character agent at some location e_{ij} is free to move to any of its neighbouring tiles. In other words, agents can have up to eight degrees of movement.

As noted previously in Section 4.7 depending on the nature of the scenario to be simulated the tiles in an environment will have different GTIs associated with them defined by a a set of tile labels $L = \{l_1, l_2, \dots\}$. As also noted in Section 4.7 for the work presented in this thesis L was defined as follows:

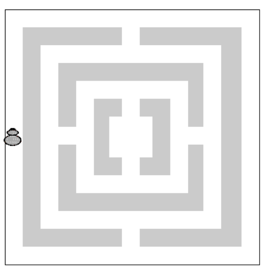
$$\{No - go\ location, Wall\ location, Choice\ point, Open\ space, Tunnel\ Location\}$$

These labels were defined previously in Table 4.1, for convenience this table is presented again here in Table 5.1. The significance of the individual labels will become clearer later in this thesis.

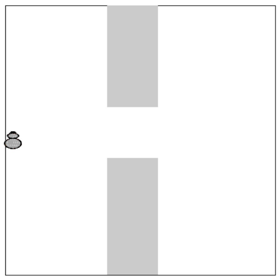
Table 5.1: Ground Type Identifiers (GTIs)

Label	Name	Description
N	No-go location	Tile that represents a location that cannot be reached by an agent, because it represents an obstruction.
W	Wall location	Tile that represents a location that cannot be reached by an agent, because it is too close to a wall.
C	Choice points	Tile location where we wish change direction should be considered, e.g a corner location.
T	Tunnel location	Tile location which is within a tunnel, or location where movement is restricted.
O	Open Space	Tile location which is not any of the above.

Figure 5.2, gives a number of example environments. Note that the H-box environment has twelve choice points (the corner locations). The outer edges of the environment are surrounded by wall locations. In the middle two obstruction agents (described below) are used to create additional wall locations in the middle of the box, adjacent to each other. The maze environment is made up of tunnel locations and choice points. The tunnel environment features the tunnel, space and wall GTIs.



(a) Maze



(b) H-Box

FIGURE 5.2: Two example environments

5.3 MBMABS Character Agents

In this section the operation of the character agents in the MBMABS Framework is presented. The discussion includes the nature of some additional attributes to the standard attributes that were listed in Section 4.8 in the previous chapter.

Recall that the nature of character agents was discussed, from an abstract perspective, in Chapter 4. As noted previously every character agent in the proposed MABS has six main attributes:

1. A set of desires, D.
2. A location within the environment. A character agent's location is expressed in terms of x-y coordinates referenced to the origin on the environment (tile world) in which it will operate.
3. A current state, defined by a vertex in the behaviour graph (discussed in Chapter 4) associated with the character agent.
4. A "stateTime".

5. A direction in which it is facing or travelling in (expressed in terms of the four cardinal (north, south, east, west) and four inter-cardinal (north west, north east, south west and south east) directions.

6. An identification number.

A character agent is introduced into an environment through a “gate". The “gate" is an entry point defined by an $x - y$ location, and a tile designation. The character agent has the ability to recognise the nature of locations in the environment when it visits them (the set of GTI labels L). It also “knows” if a location is currently occupied by another agent.

At the start of a simulation a character agent will feature a desire to explore its environment and create a “mental map” of its environment through a learning mechanism called *mental mapping*. A character agent’s mental map is essentially a graph where the vertices are “points of interest” (such as its nest site) and the edges are desired routes between points of interest. The map is used in the event of danger so that character agents can attempt to reach a place of safety. The process whereby character agents generate a mental map is presented in detail in Chapter 6.

Character agents of course also have a view of their current surroundings although harvest mice do not use vision in a primary manner in the same way that humans do. The concept of vision is defined in terms of a *vision radius* V_r. The disc defined by this radius then defines a character agent’s *vision map*; essentially a 2D model of the area surrounding a character agent’s location up to a distance of V_r. Of course a character agent cannot “see” behind obstructions. The variable, V_r is known as the "Visibility Constant" for the character agent, and it is a variable parameter. Note that a character agent’s vision map is updated after every iteration of the simulation. Note also that character agents utilise the vision map to determine distances to other character agents.

5.3.1 Simulating Social Behaviour

Each character agent in the MBMABS Framework will be associated with a single behaviour graph that describes its social behaviour (as noted previously alternative arrangement include character agents having individual behaviour graphs, or one group of agent being associated with one behaviour graph while another group of agents is associated with another behaviour graph). Recall from Chapter 4 that the vertices in a behaviour graph represent states; every character agent is associate with one, and only one, state at any simulation time t_i. States define the current activity of character agents. The action method associated with each state is used to implement the activity (for example mapping, nesting and so on). When a state change is required for a character agent in a current state s, all of the follow on states connected to s by directed edges in the behaviour graph become available for selection. State changes occur as the result of events (internal or external). Selection of follow on states is directed by desires. As discussed previously, a new state will be selected in a probabilistic random manner by assigning weightings to edges in the behaviour graph according to the strength of individual desires. In this manner the dominant desire is likely to determine the follow on state, but not necessarily so. Using the behaviour graph concept both simple behaviour, such as selecting a direction of travel, and complex social behaviour, such as nesting, can be simulated.

Movement is an important aspect of behaviour simulation and by extension the operation of the proposed MBMABS Framework; behaviourologists claim that the behaviour of mammals can best be decoded in terms of movement [109]. It is argued that through movement patterns complex social behaviour can be better understood, because most movements are caused by some motivation. In the context of the proposed MBMABS movements are states (as will become clear later in this chapter the MBMABS behaviour graph includes a number of states that feature movement). The general mechanism for simulating movement, especially selection of direction of movement, was described in chapter 4. The implementation of this mechanism is described in Chapter 6. In the context of the proposed MBMABS, speed is measured in centimetres per second. Currently a MBMABS character agent moves at a constant speed; the option of having variable speeds is considered as an item for future work.

5.4 MBMABS Obstruction Agents

Recall that the kinds of agents identified in the context of the generic MABS framework described in Chapter 4 included static agents. As noted in Section 5.2 one kind of static agent is the environment agent. Another kind of static agent is the obstruction agent. The nature of obstruction agents is discussed in this section. An obstruction agent consists of a set of locations, for example given an obstruction agent O_i this may be defined as: $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$. Obstruction, O_L will thus have the appearance of a two dimensional block measuring 2×2. Obstructions are "no-go" areas for character agents and are marked as such using the *no go* label from the set of tile labels L (see above). Zero, one or more obstruction agents can be located within an environment, thereby creating a landscape within the environment. Obstruction agents are very useful for setting up various experimental models as illustrated on Figure 5.2 (a) and (b), they can also be used to represent many kinds of physical objects which may be found in a real world mammalian environment.

The interaction between character agents and obstruction agents is such that obstruction agents are recognisable to character agents and consequently they can be utilised by character agents, for example to satisfy the behaviour called thigmotaxis (this was discussed in Chapter 3), which causes a strong desire for walls. The vision map that individual character agents posses do not feature locations obfuscated by obstructions, consequently a character agent will be unaware of other nearby character agents "hidden" behind obstruction agents. Note also that when a character agent encounters an obstruction agent in its environment this may lead to a state change. For example a character agent in a moving state may be forced to adopt a stop sate or a turning state when encountering an obstruction agent, the encountering of an obstruction agent in this case is an *external event*.

Returning to Figure 5.2. The H-box environment contains two obstruction agents so that the environment, when observed in plan view, forms an "H" shape. The Maze-box environment had six obstruction agents arranged in a "maze" formation. Each of these configurations was applied in various experiments used with respect to the cases studies used to evaluate the proposed MBMABS as presented later in this thesis in Chapter 7.

5.5 MBMABS Housekeeping Agents

As noted previously, house keeping agents are used to enable the operation of the MBMABS framework. With respect to the implementation of the proposed MBMABS only one house keeping agent was used (it is difficult to identify a situation where we might want more). In the context of the MBMABS framework the house keeping agent performs the following functions: (i) monitoring of character agent, (ii) recording the simulation (significant with respect to later evaluation), (iii) controlling the operation of the simulation, and (iv) interfacing with the other components of the MBMSBS such as the visualisation module and the user interface (as discussed further in Section 5.7).

5.6 MBMABS Utility Agents

Utility agents help to facilitate the operation of agents within the framework by supporting the completion of specialised tasks. The most important utility agent within the MBMABS framework is the Bressenham agent used to find obstruction free paths between interesting locations in the environment. Bressenham line algorithm [130] used by this agent. The significance of the Bressenham agent will become apparent in Section 6.6.

5.7 Simulation Interface and Visualisation

The MBMABS framework interface is a menu driven mechanism for setting up and running simulations. Setting up includes the definition of the environment (incorporating one or more obstruction agents if desired) and specifying the number of agents to be included and their start (gate) locations. The idea is that by using this interface the users can create agents, set their parameters and configure their environments for experimental simulation purposes. The visualisation component of the proposed MBMABS framework is for observing the output from the framework relevant to a specified simulation.

Figure 5.3 illustrates the implementation of the visualisation tool of the MBMABS interface. Note that the interface includes a visualisation window.

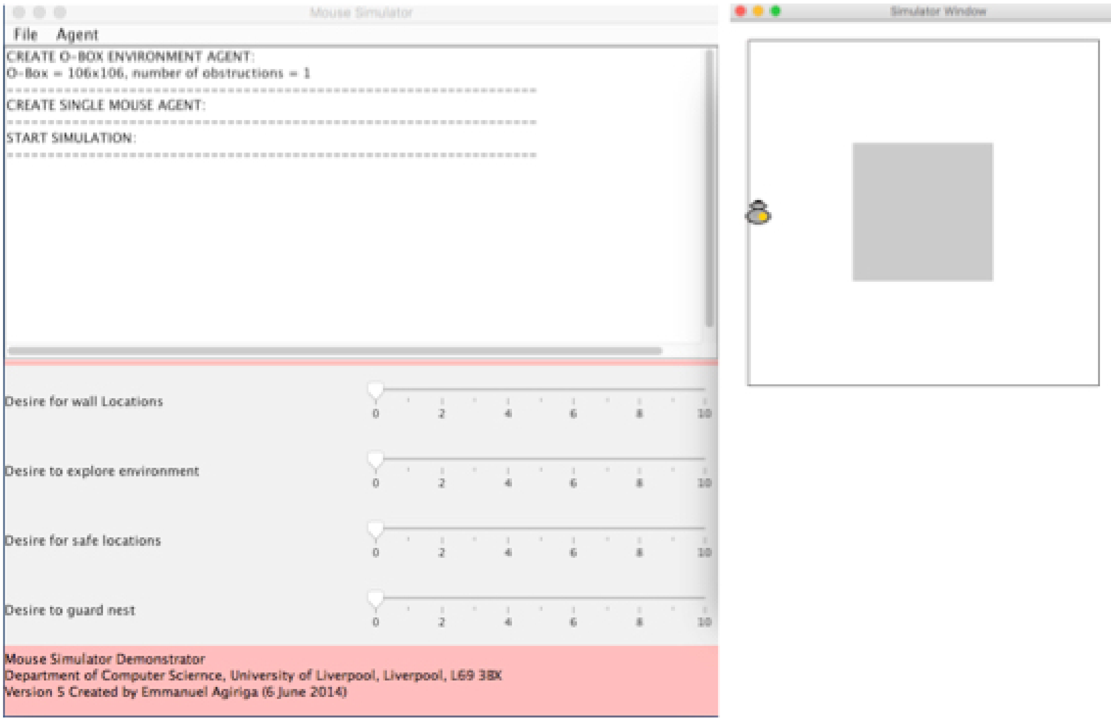


FIGURE 5.3: The MBMABS Framework

5.8 Temporal Considerations of the MBMABS Framework

This section provides a brief description of the operational temporal aspects of the proposed MBMABS Framework. Simulation systems, such as the proposed MBMABS framework, produce empirical measurements. It is through these empirical measurements that quantitative analysis can be done (as required). Therefore a discrete time model was required with respect to operation of the proposed MBMABS Framework. In the context of the proposed MBMABS time measurement was based on simulation time.

Simulation time equates to iteration time (loop time), the time between iterations equates to a simulation time of one. On each iteration simulation time is incremented by one. At the start of a simulation simulation iteration time equates to zero. We use the notation t_i to indicate the simulation time at iteration i, thus $t_0 = 0$. Note that simulation time is not the same as state time although both are measured using the same units. State Time is the amount of simulation time that an agent spends in a particular state. Note also that the time whereby dynamic desires are incremented/decremented is also governed by simulation time.

For evaluation purposes one unit of simulation time was equated to 40 milliseconds. In terms of the visualisation associated with the MBMABS framework this was equivalent to 25 frames per second. Although this value may of course be adjusted if necessary, it was found that this setting gave a good visualisation while at the same time providing for adequate computation time. Using this setting, and given that character agents will move at a rate of one grid square per iteration, and (as noted above) a grid square measures 8×8 cm a mouse character agent travels at $20 \times 60 \times 8 = 9600$ cm per minute when moving in one of the cardinal directions, or $20 \times 60 \times 11.3 = 13560$ cm per minute when moving in one of the intercardinal directions. This equates to 5.76 and 8.14 km per hour respectively ($\frac{9600 \times 60}{10000} = 5.76$ and ($\frac{13560 \times 60}{10000} = 8.14$). A mouse runs at a top speed of about 10 km per hour so this is a fairly realistic simulation speed.

5.9 Summary

This chapter has briefly described the structure of the desired Mammalian (mouse) Behaviour MABS, MBMABS, in terms of the generic framework presented previously in Chapter 4. The five different categories of agent used within the MBMABS were fully described: (i) environment agents, (ii) obstruction agents and (iii) character (mouse) agents, (iv) housekeeping agents and (v) utility agents. The operation of the framework was also fully discussed including the associated temporal considerations. The next chapter describes a number of case studies that utilised the MBMABS framework, the aim being to provide the reader with a more complete understanding of the operation of the proposed MBMABS framework.

Chapter 6

Mouse Behaviour MABS Realisation and Case Studies

6.1 Introduction

In Chapter 4 an abstract (generic) MBMABS framework was presented founded on the behaviour graph concept. This abstract framework was then, in Chapter 5, used to define a MBMABS framework, a framework for mammalian (mouse) behaviour MABS. This chapter describes how the MBMABS framework presented in Chapter 5 can be used to implement various mouse behaviour simulation case studies. The aim is to illustrate the operation of the proposed MBMABS framework in terms of mouse behaviour simulation. To this end three categories of “mouse in a box" case study were considered: (i) single mouse in a box without obstructions, (ii) single mouse in a box with obstructions, (iii) mouse in a box responding to danger. These case studies were chosen because they had been used for real life mouse-in-a-box experimentation (they were originally designed and undertaken by a team of animal behaviourists led by Prof. Jane Hurst of the Mammalian Behaviour and Evolution group, Institute of Integrative Biology at the University of Liverpool). Expert domain knowledge concerning these case studies was thus available which in turn could therefore be used to (i) populate the required behaviour graphs, (ii) inform the nature of individual agent desires and (iii) conduct evaluations.

For the purpose of using the proposed MBMABS framework described in the previous chapter, to implement particular simulation case studies, it was found that the case studies to be simulated could best be considered in terms of the activities to be considered. To this end, a categorisation of activities made up of “Primary Activities” and Secondary “Activities” was considered. A primary activity incorporates a number of secondary activities each represented by states contained within the behaviour graph. Note that secondary activities can be shared between primary activities.

In the context of the three categories of case study considered in this chapter a number of primary activities were identified as follows (the applicable case study number is indicated in parentheses):

1. Movement (case study 1)
2. Exploration (case study categories 1 and 2)
3. Nest Discovery (case study categories 1 and 2)
4. Safe Travel Route Identification (case study category 3)
5. Nest Defence(case study category 3)

The focus of this chapter is the implementation of these primary activities in the context of the MBMABS framework presented in the previous chapter.

The remainder of this chapter is structured as follows. In Section 6.2 a more detailed overview of the mouse-in-a-box case studies considered in this chapter is presented. This is followed in Section 6.3 by a discussion on the behaviour graphs associated with each case study category. Section 6.4 then discusses the nature of environment agents used in the context of the case studies; six different environments were considered. The relevance of these environments is that they were used for evaluation purposes as described in the next chapter, Chapter 7. Section 6.5 then provides some specific detail concerning the definition of the mouse agents used with respect to the realisation, including: (i) the nature of a number of additional attributes not considered previously, and (ii) the desires featured by the mouse agents (in the context of the considered case studies). Section 6.6, then considers the five above listed primary activities. The chapter is concluded with a summary presented in Section 6.7.

6.2 Overview of The Mouse in a box Case Studies

Recall that the "mouse in a box" case studies are a series of case studies where one or more mice are placed in a 1.22×1.22m box[1], together with various features (obstructions, tunnels, nest sites, etc.). For each case study the mouse behaviour was observed by behaviourologists (the domain experts), and the knowledge gained used to inform the construction of the proposed MBMABS. The knowledge was gained through regular unstructured interviews conducted by the author with the behaviourologists. In addition, in the single mouse-in-a-box with no obstructions case, a video camera was suspended over the box and the behaviour recorded. This was not done in all cases because of the resource required to do this. Two stills from the collected video were given in Chapter 3, for completeness these stills are given again in Figures 6.1 and 6.2 below. Further detail concerning the three categories of case study is given below:

[1]The value 1.22 is a result of the fact that the board from which the boxes are typically fashioned comes in $2.44 \times 1.22m$ sheets

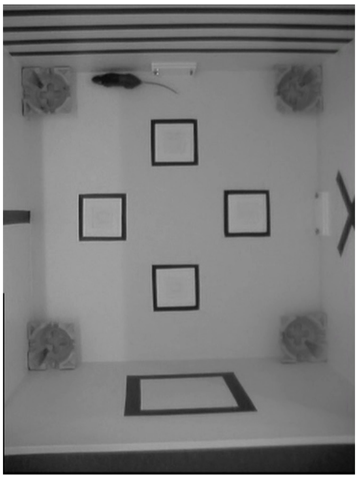


FIGURE 6.1: Still from Mouse Behaviour Video Data - Example 1

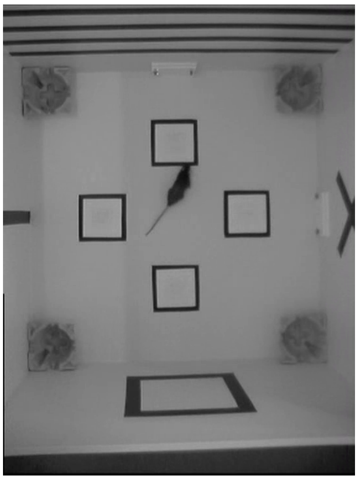


FIGURE 6.2: Still from Mouse Behaviour Video Data - Example 2

Single Mouse In a Box Without Obstructions. In this setting the mouse agents were expected to exhibit a common mouse characteristic known as *thigmotaxis*, an affinity to walls [21], explore their environment, and to find a nest site. The mouse agents in simulations of this first category of case study would therefore be expected to have a tendency to move along the sides of the box (although not exclusively so) as they moved round there environments. This tendency would be driven by an appropriately formulated desire. Note that the box used for this category of case study has no obstructions within it.

Single Mouse In a Box With Obstructions. Mice are interested in exploring their surroundings, the ultimate goal is to find and maintain an "optimum" nest location. Simulations related to this second category of case were thus expected to feature mice agents that wish to firstly explore their environment and secondly identify an appropriate nest site. As such they mouse agents were facilitated with desires to explore, at the same time they would also feature a desire for wall locations. The box used for this category of case study has obstructions, distinguishing it from case study category 1.

Single Mouse Responding to Danger. The third category of case study considered was the most complex in that it included a broader range of behaviours. More specifically this third category of cases study included mouse agents responding to danger and defending their nest sites from intruders. Thus in this category of case study the mouse agents featured desires for safety and guarding of their chosen nest site. The mouse responding to danger category of case study features box with and without obstructions.

With respect to the above, and as noted previously, it should be noted that the case studies considered in this thesis were only directed at male mice due to the relative complexity of their behaviour as discussed in Chapter 3. This was also because the behaviourologists consulted considered that male mouse behaviour was more significant with respect to pest control, one of the motivations for the work presented in this thesis. Further case studies involving female mice are suggested as a potential direction for future work.

In the next section, the behaviour graph for each of the above case study categories are provided.

6.3 Behaviour Graph For Mouse In A Box Case Studies

In this section each of the above categories of case study is considered individually with respect to the behaviour graphs required for their implementation.

6.3.1 Behaviour Graph For Case Study Category 1

The behaviour graph for case study category 1 is given in 6.3. From the figure it can be observed that the behaviour graph features 8 vertices (states): (i) Start, (ii) MovingAlongWall, (iii) StoppedAtWall, (iv) MovingInSpace, (v) Turning, (vi) StoppedInSpace, (vii) StoppedAtCorner, and (viii) CreatingNest. Recall that each state represents a particular activity which the mouse agent may be performing at a particular time t_1 in the simulation. The meaning of each state can be derived from its nomenclature. For instance the "Start" state is the current state at the beginning of the simulation and the "MovingAlongWall" state is associated with movement along wall locations. Each of these States have one or more permissible follow-on-states. The directed edges of the behaviour graph given in Figure 6.3 indicate a transition from a current state to a follow-on-state as indicated by the direction of the arrows. Some states have several possible follow-on-states. For example, at the start of a simulation a mouse agent takes on the "Start" state, at time, t_0. The mouse can then either: (i) move along a wall to begin exploring, in which case it has transited to the "MovingAlongWall" state; or (ii) it may choose to stay at its current wall location, in which case it is in the "StoppedAtWall" state, or (iii) it may choose to move to a space location, hence a transition to a "MovingInSpace" state. This is further described by the behaviour matrix presented in Appendix C. The columns of the behaviour matrix are: the current state, the follow on state, event (may be internal or external as described in Chapter 4), and selection refers to the nature of the state change which may be fixed or probabilistic, as discussed in Chapter 4.

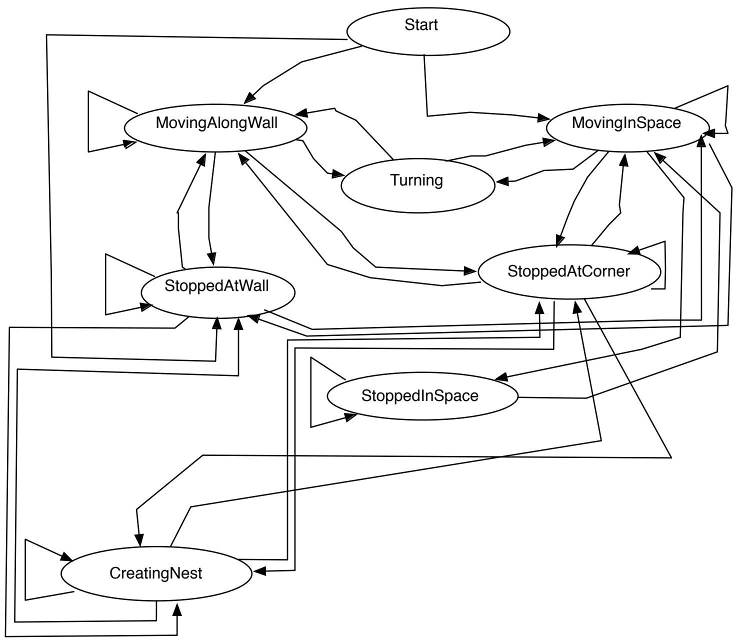


Figure 6.3: Single Mouse Without Obstructions (case study category 1) Behaviour graph

6.3.2 Behaviour Graph For Case Study Category 2

The behaviour graph for case study category 2 is given in Figure 6.4. From the figure it can be observed that the behaviour graph features 10 vertices (states), they include: (i) Start, (ii) MovingAlongWall, (iii) StoppedAtWall, (iv) MovingInSpace, (v) Turning, (vi) StoppedInSpace and (vii) StoppedAtCorner. (viii) StoppedInTunnel (ix) MovingInTunnel (x) CreatingNest. This is further described by the behaviour matrix presented in Appendix D

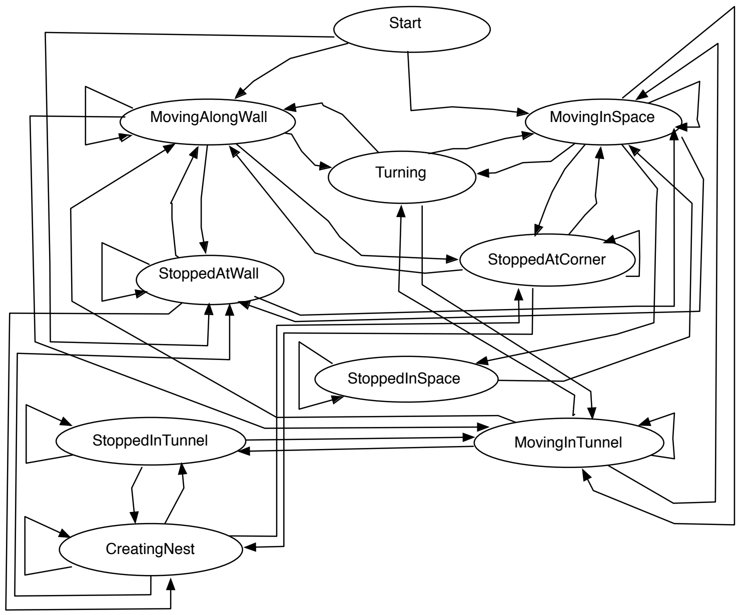


Figure 6.4: Single Mouse With Obstructions (case study category 2) Behaviour graph

6.3.3 Behaviour Graph For Case Study 3

The behaviour graph for the mouse behaviour case study category 3 is given in the figure 6.5. From the figure it can be observed that the behaviour graph in this case features 16 vertices (states), they include: (i) Start, (ii) MovingAlongWall, (iii) StoppedAtWall, (iv) MovingInSpace, (v) Turning, (vi) StoppedInSpace and (vii) StoppedAtCorner (viii) MovingToNearestSafeLocation (ix)MovingAlongTravelLines (x) StoppedAtNestSite (xi) Resting, (xii) GuardNestSite (xiii) AvoidNestSite (xiv) MovingInTunnel (xv) StoppedInTunnel (xvi) CreateNestSite. Each representing a particular activity which the mouse agent may be performing at a particular time t_1 in the simulation. As before the meaning of each state can be derived from its nomenclature. This is further described by the behaviour matrix presented in Appendix E.

To assist in the understanding of the behaviour graph in Figure 6.5, an example scenario that uses this behaviour graph is given in Figure 6.6. The numbers used to label the states indicate a sequence of states. In the example the mouse agent starts in the "Start" state (top of the figure). The example assumes that the start location is a wall location. The mouse agent proceeds through a series of state changes to create a new nest site. At state 1 (the "Start" state), the mouse agent is at a wall location, and chooses to turn in order to change its direction, hence the "Turning" state is adopted in state 2. State 3 indicates that the mouse agent has not completed a turn, hence the current state remains, "Turning". State 4 indicates that the mouse agent is now moving along walls searching for a suitable nest site. State 5 demonstrates that the mouse agent is still moving along walls, searching for a suitable location to create its nest. State 6 indicates that the mouse agent has arrived at a corner location. State 7 indicates that the mouse agent has selected that corner location as a suitable site to create its nest.

State 8 demonstrates that the mouse agent is still creating its nest. At state 9 the mouse agent has created its nest site and at State 10 the mouse agent is still at this nest site.

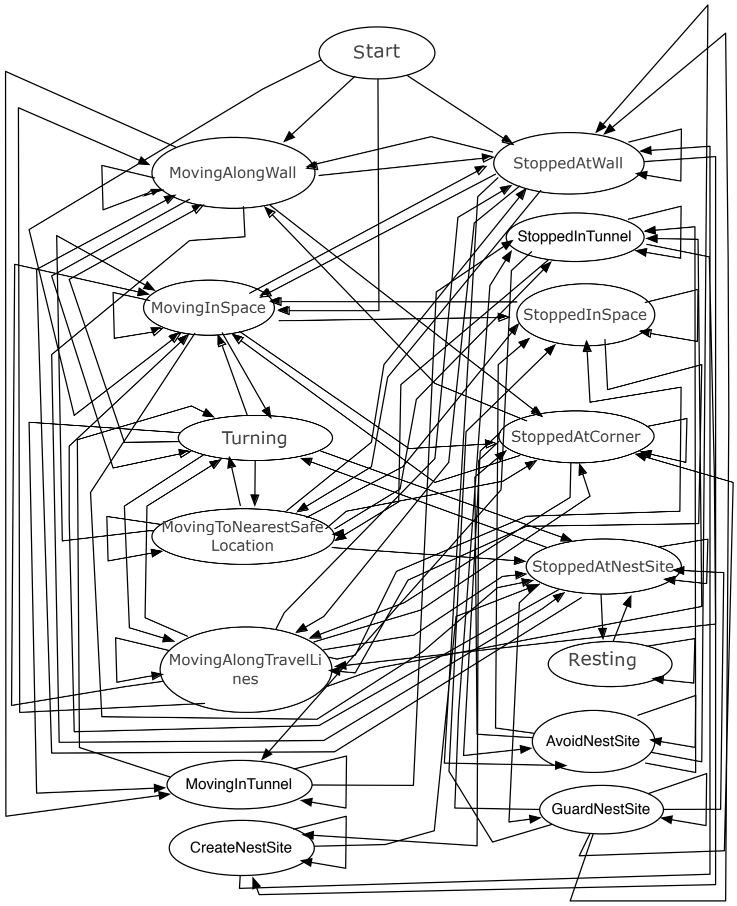


FIGURE 6.5: Mouse In A Box Responding to Danger (Case study category 3) Behaviour Graph

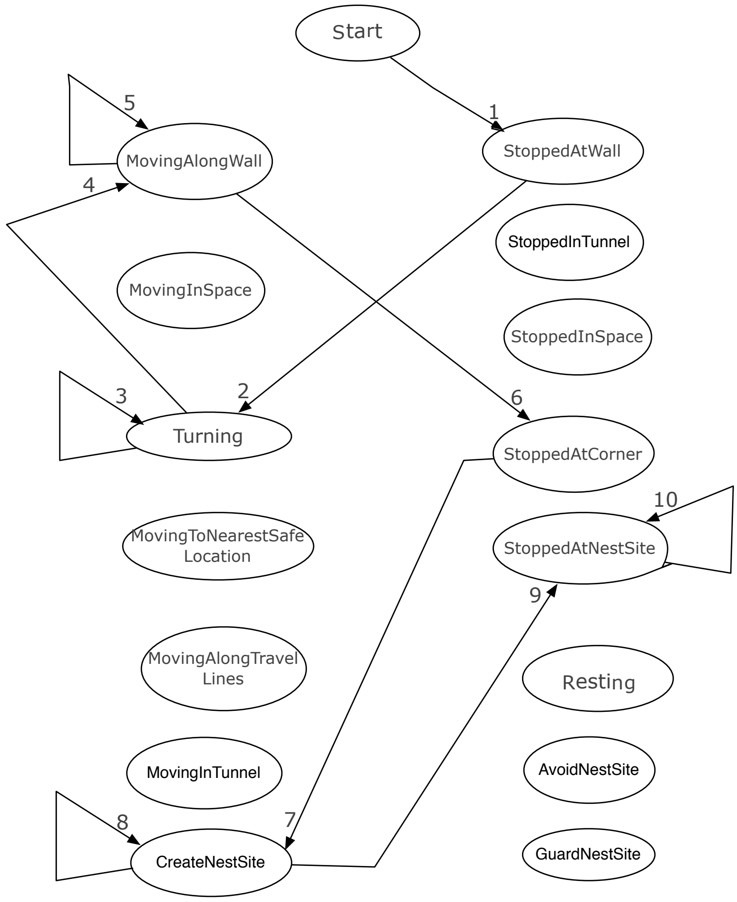


FIGURE 6.6: Extract from Fig. 6.5 Behaviour Graph Showing Possible Example of Nest Creation

6.4 The Environment Agent

As noted in earlier chapters the environments in which MABS agents operated were defined in terms of tile world's 2D grids; the reasons for this were presented earlier in the thesis (see Chapter 5). This section presents the nature of the environments used with respect the realisation of the case studies considered in this chapter and used for evaluation purposes in the following chapter.

Environment agents were defined as having the following fields:

1. $widthX$, the width of the environment, in terms of grid squares, in the X (East-West) direction.

2. $widthY$, the width of the environment, in terms of grid squares, in the Y (North-South) direction.

3. $groundArea$, the two dimensional grid describing the *locations* that make up the entire playing area.

4. *gateCoords*, one or more "gates" where mouse agents can enter the environment (start points).

5. *obstructionList*, a list of zero, one or more obstruction agents that the environment needs to know about.

Each grid square (location), in turn, featured two fields: (i) a Scent Record (SR) and (ii) a Ground Type Identifier (GTI). Each is considered in some further detail in the remainder of this section.

According to behaviourologists, when a mouse moves it leaves scent markings as a way of marking out its territory, this plays an important role with respect to intersexual relationships and to deter territorial intruders. In the context of the proposed MBMABS, as a mouse agent moves around its environment it leaves scent trails at "interesting" locations or along routes (one particular location is its nest location). Thus each location (grid square) within an environment has a record of any scent at that location, as well as the *id* of the mouse agent to which the scent sample belongs. More specifically the scent field comprises a set SR where $SR = \{S_1, S_2, \dots\}$. Each element S_i in this set is a tuple of the form $\langle str, ID_m \rangle$ where *str* is the current scent strength (thus at the current simulation time t_i) and ID_m is the associated mouse agent identifier. The set SR for every location will be empty at start up. Scent strength is defined as an integer, different mouse agents have different sent strengths according to their "dominance". Currently the maximum scent strength is 255. Scent traces persist for 8 to 24 hours depending on the strength/dominance of the mouse. Mouse scent "degrades" with time. In the MBMABS, on each iteration, scent strength is degraded by a *degradation factor*. The degradation factor (*df*) is a global parameter specified for each simulation run, it is a numeric value of between 0 and 1. The default setting is 0.01 because it supported more realistic mouse behaviour, as confirmed by behavioural experts. The degradation is calculated using Equation 6.1.

$$df = \frac{MaximumScentStrength}{Num.OfSimulationIterationsPerSec} \tag{6.1}$$

Thus, for example if the maximum scent strength is 1.0 and the number of simulation iterations per second was 25 the *df* value would be $1.00/25 = 0.04$. Thus if the scent strength *str* value at time t_i is 0.80, the *str* value at time t_{i+1} will be: $0.80 - 0.04 = 0.76$.

Recall that the GTIs were expressed in the form of a set of labels, L each indicating the nature of the terrain that a grid square (tile) might represent. With respect to the case study implementation five different GTI codes were used. The GTI codes were presented in Sections 4.7 and5.2. For easy reading, it is presented again in Table 6.1. It should be noted that the available set of GTIs was defined in consultation with mammalian behaviourologists and were designed to reflect the real world mouse in a box case studies considered in this chapter.

TABLE 6.1: Ground Type Identifiers (GTIs)

Label	Name	Description
N	No-go location	Tile that represents a location that cannot be reached by an agent, because it represents an obstruction.
W	Wall location	Tile that represents a location that cannot be reached by an agent, because it is too close to a wall.
C	Choice points	Tile location where we wish change direction should be considered, e.g a corner location.
T	Tunnel location	Tile location which is within a tunnel, or location where movement is restricted.
O	Open Space	Tile location which is not any of the above.

With reference to Table 6.1 and given a mouse in a box case study without any obstructions all grid squares within three units of a wall would be considered to be no-go tiles (N) (to take account of a mouse agent's "size"), squares exactly four units away from a wall would then be labelled as wall tiles (W), and all squares more than four units away from walls, as open space tiles (O). Choice points are associated with grid squares representing wall locations where current movement may proceed in more than one direction (for example in the case of the maze environment presented later in this chapter). Tunnel locations are constructed using obstruction agents placed so that one or more tunnels are formed. Note that the open space GTI (O) is the default GTI. Figure 6.7 shows an environment that features all of the above GTIs.

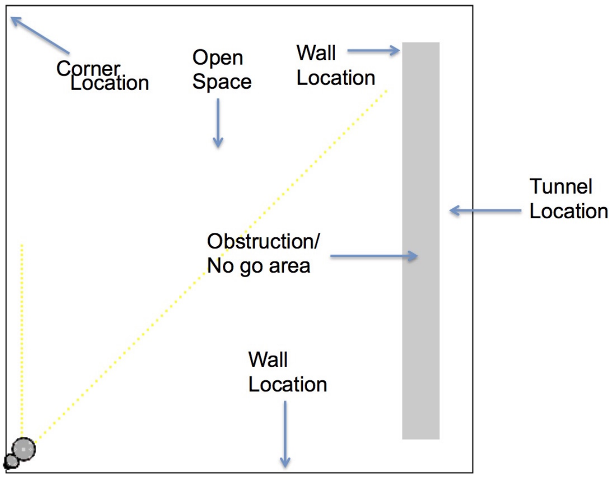


FIGURE 6.7: Example environment featuring a selection of the possible GTIs

For the case studies considered in this chapter, and the evaluation considered in the following chapter, six different environment agents were considered, each designed with the objective of testing some specific form of behaviour. The environments were as follows: (i) Box, (ii) Four-Box, (iii) Maze, (iv) H-Box, (v) O-Box, and (vi) Tunnel. Some of these environments were presented and illustrated in Chapter 5, Figure 5.2. All six are illustrated in Figure 6.8. Table 6.2 provides a summary of each environment. Of course the nature of the proposed MABS framework is such that any number of environments may be created.

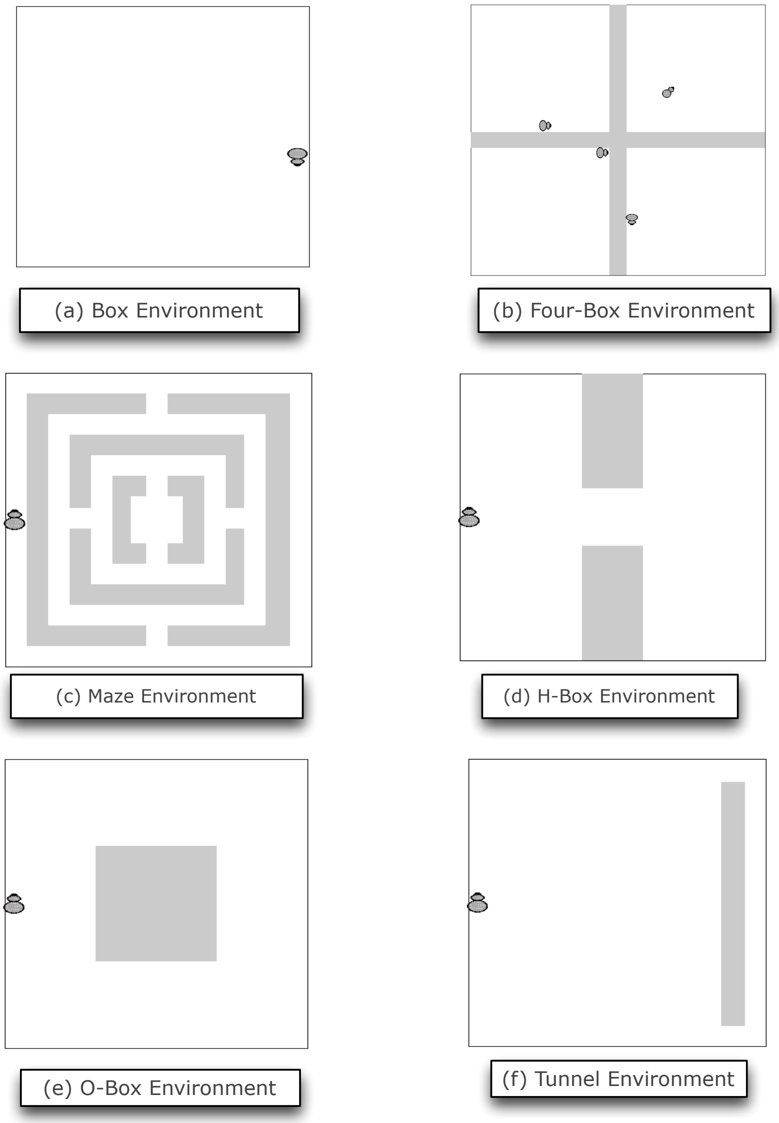


Figure 6.8: Environments considered for Case Study Categories listed in Section 6.2

TABLE 6.2: Description Of Environment Types

Environment Type	GTI Labels (described in Table 6.1)	No. of Obstructions	Evaluation Objective
Box	W, O, C	0	To observe behaviour without obstructions
H-Box	W, O, C	2	To observe behaviour in environment divided into areas
O-Box	W, O, C	1	To observe behaviour with obstruction in the middle of a box
Four-Box	W, O, C	0	To observe four mice agents operating concurrently in similar spaces
Tunnel	W, O, C, T	1	To observe behaviour in restricted spaces
Maze	W, O, C, T	6	To observe behaviour in many restricted spaces

6.5 The Mouse Agent

As noted in the previous chapter mouse agents are the central players in the proposed MBMABS framework. With respect to the usage of the MBMABS framework presented in the previous chapter to realise the case study categories discussed in Section 6.2, mouse agents were realised using character agents. They therefore have all the attributes of character agents listed in Chapter 5 namely:

1. A set of desires, D.
2. A location within the environment. A character agent's location is expressed in terms of x-y coordinates referenced to the origin on the environment (tile world) in which it will operate.

3. A current state, defined by a vertex in the behaviour graph associated with the mouse agent.

4. A "stateTime".

5. A direction in which it is facing or travelling in (expressed in terms of the four cardinal (north, south, east, west) and four inter-cardinal (north west, north east, south west and south east) directions.

6. An identification number.

However, in the context of the case studies, the mouse agents were also required to possess a number of additional attributes: The nature of these additional attributes are as follows:

1. *goalDirection*, the direction the agent wishes to face (only applicable when in a turning state as discussed further below).

2. *turnDirection*, the "turning direction", either *clockwise* or *anticlockwise* (also only applicable when the mouse agent has adopted a turning state).

3. *scentStrength*, the strength of the mouse agent's scent.

4. *visionMap*, a disc of locations, with radius V_r, representing the part of the environment which a mouse agent can "see". Thus a mouse agent's field of vision. A mouse agent knows nothing about the locations of other mice until they appear in its vision map. The radius of the vision map (V_r) was set to 20 grid squares. However if the location of another mouse agent is obscured by an obstruction agent the current mouse agent will not know anything about this other mouse. To ensure the mouse agents do not actually crash into each other a buffer region of 10×10 grid square was placed round other mouse agents.

6.5.1 Mouse Agent's Desires

The concept of desires was discussed in an abstract manner in Chapter 4 and in the context of the proposed MBMABS framework in Chapter 5. In Chapter 4 it was noted that the proposed generic MABS framework supported two kinds of desire: (i) dynamic (strength changes with time) and (ii) static (strength remains fixed). It should also be recalled that the desire strength associated with dynamic desires can either change abruptly, jump from one value to another, or increase/decrease in a steady manner. The mechanism whereby the latter was achieved was also discussed in Chapter 4. In the context of the usage of the MBMABS framework to implement the case studies considered in this chapter, four desires were required:

1. A static desire to stay close to walls (d_w) (Thigmotaxis).

2. A dynamic desire to explore an environment (d_e).

3. A dynamic desire for safety (d_s).

4. A dynamic desire to guard a nest site (d_g).

In other words the set D for each of our agents comprised $\{d_w, d_e, d_s, d_g\}$. Each of the above listed desires is briefly described in the following four subsections below. Note that in each case the presented numerical definition for each desire was determined through a process of consultation with domain experts and the analysis of simulation runs in terms of realism.

It should be noted here that there are a number of additional desires that would be of further benefit to behaviourists if included in the proposed MBMABS but were not included because of time constraints. These additional desires include: (i) the desire for food (hunger) and (ii) the desire to find a mate. These are discussed further in the future work section included in Chapter 8.

6.5.1.1 Desire for walls

The desire for walls (d_w) is a static desire. The desire strength was a fixed value of $d_w = 1$ throughout the simulation. Real world mice have a constant affinity for walls or relatively enclosed spaces as discussed in some detail in Chapter 3. Each mouse agent has this desire, regardless of the nature of the case study under consideration.

6.5.1.2 Desire to explore

The desire to explore (d_e) is a dynamic desire. At the start of a simulation, simulation time $t = 0$, $d_e = 1$. The d_e desire strength then steadily decreases as the simulation progresses and jumps back to 1 whenever the mouse agent encounters a new location of interest within its environment. The definition of the concept of an "interesting location" will be presented later in this chapter in Subsection 6.6.2 in the context of the exploration and nest discovery primary activities. The minimum value for d_e is 0 (note that dynamic desires do not necessarily have to reduce to 0).

6.5.1.3 Desire for safety

The dynamic desire for safety, d_s,was set at 0.5 at simulation time $t = 0$, and jumps to $d_s = 1$ whenever there is perceived danger in the environment. Otherwise, it remains at its default value of 0.5. It remains at 1 while the perceived danger continues to exist and decreases slowly back to 0.5 when danger is no longer perceived.

6.5.1.4 Desire to guard nest

Real mice considered their nest site to be a place of refuge, a resting place or a kind of home. With respect to the MBMABS the nest location in the environment is chosen by the mouse agent. The desire to guard the nest d_g is set to 0 at $t = 0$ and jumps to 1.0 whenever another mouse agent intrudes within a radius rd of the mouse agents nest, essentially a 2D model of the area surrounding a mouse agent's chosen nest site up to a distance of rd. The value for d_g then remains at 1.0 until the intruder moves out of the rd radius when it decreases steadily back to 0. Where the mouse agent has not chosen a nest location, this desire constantly remains at 0. The value for rd can be specified prior to the start of a simulation, alternatively a default value 60.0 grid squares can be used. This value was chosen because it created a sufficient space between an intruder and a nest, to realistically simulate nest site intrusion.

6.5.2 Mouse Agent States

In the introduction to this chapter is was noted that the implementation of the case studies of interest was conducted in terms of primary activities and secondary activities where primary activities comprised secondary activities and secondary activities were represented by states.

The behaviour graph, and the twin concepts of states and state changes, was discussed extensively in the foregoing chapters. This section considers the states required to realise the primary activities, which in turn were required with respect to the case study implementations. In total the realisation of the primary activities required sixteen states. These sixteen states can be grouped, for ease of presentation, as being either: start (1 state in total), moving states (6 in total), stopped states (5 in total) or nest states (4 in total). The six moving states comprise: (i) Moving In Space, (ii) Moving At Wall, (iii) Moving In Tunnel, (iv) Turning, (v) Moving Along Travel Lines and (vi) Moving To Nearest Safe Location. The five stopped states comprised: (i) Stopped In Space, (ii) Stopped At Corner, (iii) Stopped At Nest Site (iv) Stopped At Wall, and (v) Stopped In Tunnel. The four nest states are: (i) Creating Nest Site, (ii) Guard Nest Site, (iii) Avoid Nest Site and (iv) Resting.

A summary of these states, for reference purposes is given in Table 6.3

Table 6.3: Summary of Mouse Agent States

State Grouping	Individual States	Description
Start	Start	Default state of mouse agent at beginning of simulation.
Moving	Moving In Space, Moving At Wall, Moving In Tunnel, Turning,	Mouse agent is travelling with respect to: space locations, wall locations, tunnel locations or changing its direction.
	Moving Along Travel Lines	Mouse agent has previously identified paths to its nest site, which it considers safe, and is now intentionally travelling along those paths to make its way to its nest.
	Moving To Nearest Safe Location	Mouse agent travelling along known routes to find a safe location.
Stopped	Stopped In Space, Stopped At Corner, Stopped At Nest Site, Stopped At Wall, Stopped In Tunnel	Stopped at either space, corner, nest, wall or tunnel location.
Nest	Creating Nest Site	Mouse identifies a new location for nest.
	Guard Nest Site	Mouse agent moving around its nest location for an extended period to ward off potential intruders
	Avoid Nest Site	Mouse agent staying away from a nest site.
	Resting	Mouse agent is stopped for an extended period of time in a safe location within the environment. The distinction between the resting state, and the stopped state is that the agent usually rests at its nest. This activity always lasts longer than the stopped state.

Recall from Chapter 4 that within the proposed MBMABS framework character agents always have one current state and that every state has one or more possible follow on states (except an end state if present). An illustration of how state changes operate was given in Chapter 4. In Chapter 2 the concept of a state change matrix, the term behaviour matrix was used, as a tool for describing the potential follow on states for a given state and the necessary action methods required to realise that state change was highlighted.

In Chapter 4 it was also noted that agents cannot remain in a particular state indefinitely. State changes occur as a result of events, internal or external. One form of internal event was referred to as timing out, the process where by this was achieved was again presented in Chapter 4. The timing out concept was adopted with respect to the *stopped* and *moving* states. An alternative kind of internal event was the completion of an activity. This process was adopted with respect to the nest states because the amount of time spent at nest states is influenced only by the desires of the mouse agent, because the amount of time the mouse agent spends at its nest may be influenced by both internal (completed resting and wants to perform another activity) and external (guarding nest or hiding from danger, or waiting till danger event is over) factors, hence the amount of time spent in the nesting states will vary. In the case of the *moving* states the maximum period of time that a mouse agent could stay in a *moving* state was set to twice that associated with the *stopped* state. This was done because empirical evidence (visualisation of simulations and consultations with domain experts) suggested this was appropriate so as to achieve realistic simulations. Thus, on each iteration, when a mouse agent is in a moving a state a state transition will be forced when:

$$r < cosin\left(\frac{90 \times stateTime/2}{T}\right) \tag{6.2}$$

where r is a random number such that $0.0 \leq r \leq 1.0$. For further detail regarding the derivation of this equation the reader is referred back to Section 4.3 of Chapter 4.

6.6 Realisation of Primary Activities

From the introduction to this chapter five mouse behaviour MABS primary activities were identified. For ease of reading these are listed again here:

1. Movement
2. Exploration.
3. Nest site discovery.
4. Safe travel route identification.
5. Nest site defence.

Recall also that, as discussed in section 6.1, that primary activities are comprised of secondary activities represented by states. The mouse agent states involved in the realisation of the above primary activities have been discussed in Subsection 6.5.2. How each of the primary activities is realised, in the context of the proposed MABS, is considered in the following six sub-sections.

6.6.1 Movement

The movement primary activity is considered to be the most fundamental activity with respect to the proposed MBMABS. Recall that states can be shared between primary activities. Referring back to Table 6.3, all the states in the "moving" and "stopped" groups are associated with the movement primary activity.

Movement is directed by the desire for walls, d_w. Mouse agents have a constant desire for walls. The value of d_w can be set to between 0 and 1 at the start of the operation of a mouse agent, and remains the same until the mouse agent stops operating.

Given a current location, as noted earlier in this thesis, there may be between zero and eight directions of movement for a given agent depending on where the agent is located in the environment and the possible presence of obstructions and/or other mouse agents. There is also the option not to move. Conceptually each of these eight directions can be viewed as a follow on states from the current state. However, because movement is such a common activity these eight conceptual states were bundled together as part of the process of implementing movement states.

The process for selecting a direction of movement is conducted by considering the eight neighbouring available locations for a given agent's location and then assigning a weighting to each these locations. Selection is made according to these weightings. The selected location then indicates the selected direction of movement and the agent will adopt a moving state (the agent may already have been in a moving state previously but with a different direction of travel). Because of the characteristic of thigmotaxis (preference for walls) displayed by mice, wall locations are weighted higher than other locations.

Algorithm 3 describes, the mechanism whereby the movement primary activity is realised. The input is a set of potential locations, L, and the desire for walls, d_w with respect to the mouse agent. The output is a new location, L_{final} selected by the mouse agent. We commence lines 1 to 6 by initializing a number of variables. Line 1 initialises N_n, which is a variable that holds the number of non-space locations in L. Examples of non space locations include walls and corner locations. Line 2 initialises N_s, which is a variable that holds the number of space locations (open space) locations in L. Line 3 initialises W_n, which is a variable that holds the total weightings for non-space locations in L. W_n was set to 0.80 for simulation experiments to reflect the significant preference that mice have for non-space locations thereby supporting realistic experiments. Line 4 initialises W_s, which holds the total weightings for space locations. The value of W_s $= 1 - W_n$. Line 5 initialises R a variable that holds a random number between 0 and

1, and it is used to support randomness within movement activity realisation. Line 6 initialises a $Prob$, which holds the calculated probability for each location, L_i in the set of Locations, L. Lines 7 to 8 calculate the individual weightings w_n for each non-space location (L_i) in L, where L, has only non-space locations. On the condition that both space and non-space location exist in L, Lines 9 to 10 calculate the individual weightings for each non-space and each space location in L.

The variables (w_n, as already mentioned above), and (w_s) hold calculated individual weightings for individual space and non space locations in L. Lines 14 to 18 calculate the preference for each location, L_i in L based on its ground type, $L_{i.groundtype}$ and using the desire associated with this activity, the desire for walls, d_w. In lines 20 to 27, L_{final} is determined in a randomised probability driven manner using a number line, and a random number, R between 0 and 1.

Algorithm 3: Realisation of Movement Activity

Input: L = Set of Potential Locations
Input: d_w =Desire For Walls
Output: L_{final} = New location

1 N_n = Number of non-space locations in L;
2 N_s = Number of space locations In L;
3 W_n = Total weightings for non space locations in L;
4 W_s = Total weightings for locations space In L;
5 $R = RandomNumberGenerator()$;
6 $Prob = 0.0$;
7 **if** $N_s \equiv 0$ **then**
8     $w_n = 1.0/N_n$;
9 **else**
10     $w_n = W_n/N_n$;
11     $w_s = W_s/N_s$;
12 **end**
13 **for** $i = 0 \rightarrow |L|$ **do**
14   **if** $L_{i.groundType} \equiv$ *space locations* **then**
15         $L_{i.movement} = w_s * d_w$;
16   **else**
17         $L_{i.movement} = w_n * d_w$;
18   **end**
19 **end**
20 **for** $i = 0 \rightarrow |L|$ **do**
21     $Prob = Prob + L_{i.movement}$;
22   **if** $R < Prob$ **then**
23         $L_{final} = L_i$;
24   *break*;
25   **end**
26 **end**
27 **return** (L_{final});

6.6.2 Exploration

Exploration is a feature of many mammalian behaviours. Animals wish to know the environment in which they are located and, when finding themselves in a new environment will wish to establish this knowledge. The conjecture is that they do this for defensive and/or nesting and/or foraging purposes. How they do this is unclear. In the case of mice, once established in an environment, they seem to "know" the best (fastest and/or safest) route back to their nest site (possibly using scent trails).

In the context of the proposed MBMABS the objective of the exploration primary activity is for an agent to create a "mental" route map of its environment. The rational behind the route map idea is that, according to behaviourologists, as the process of learning about their environment progresses, mice gradually create "safe" routes along which they tend to travel between locations of interest, especially when there is a perceived danger in the environment. In this thesis the process whereby a mouse agents generates a route map is referred to as *route mapping*.

Thus, route mapping is concerned with the discovery of "safe travel lines". A sequence of such lines makes up a "safe route". A safe travel line has no obstructions on its path and typical follows the contours of walls and obstructions, but occasionally crossing open ground where there is no alternative.

The route mapping process involves the generation of a "route map" of the environment in which a mouse agent is operating. This map comprises a set of vertices and edges (and as such should not be confused with a behaviour graph). The vertices are *waypoints* and the edges represent *travel lines*. Waypoints are locations where a "change of direction" is required, for example to circumvent an obstruction or when following a wall contour. Consequently vertices always represent locations next to walls or obstructions which are deemed to be safer than other (open space locations). As will become apparent later in this section this has implications with respect to the nest site discovery activity and the safe travel route identification activity. A complete route map of an environment will include all corner locations as vertices and the shortest unobstructed path connecting these vertices. The Bressenham Line Algorithm [131] was used to determine straight, obstruction free lines (travel lines), between vertices. An example of a route map is given in Figures 6.9 for a simple box environment; another example is given in Figure 6.10 for a O-Box environment. Both figures illustrate examples of mental maps created by a mouse agent separately exploring the Box and O-Box environments respectively. In Figure 6.9, the mouse agent has created a mental map of the environment, by identifying the interesting locations in the environment (represented in the figure as vertices), and connecting them using travel lines. In Figure 6.10 it is can be seen that the travel lines are obstruction free.

The desire to explore plays a significant part in route mapping and the exploration activity. As noted in Subsection 6.5.1 mouse agents have a dynamic desire to explore which is initially set to 1.0 and decreases until a previously undiscovered point of interest is found. Points of interest in this context are waypoints as defined above. If no waypoint is found, by the time the desire to explore reaches 0.0, it will remain at zero for the remainder of the simulation or until such time as a new point of interest (waypoint) is discovered when the desire to explore will jump back to 1.0 before starting to decrease again. Eventually, once an entire environment has been explored, The desire to explore d_e will remain at 0.0; unless of course a new obstruction is introduced.

When "exploring" mouse agents prefer locations which have either not been visited recently, or never been visited. To recognise the locations which have not been visited before, or have not been visited recently, the mouse agent uses its scent marks. The mechanism for this was discussed in Section 6.4.

Algorithm 4 describes the mechanism whereby the exploration primary activity is realised. The input is a set of locations, L and the desire to explore, d_e. The output is a new location, L_{final}, selected by the mouse agent. We commence lines 1 to 5 by initialising a number of variables. Line 1 initialises a set of inverse scent strengths, S associated with each location in L. Inverse scent strengths were used because the mouse agent prefers locations where its own scent is not present, or at least weak. Thus each new location was assigned a weighting expressed as the inverse of the mouse agent's own scent strength (s_{inv_i}) at a given location i. If no scent is present, $s_{inv} = 1.0$. Line 2 initialises a variable *total*, used to store the total inverse scent strengths for locations in L. Line 3 initialises R a variable that holds a random number between 0 and 1, and it is used to support randomness within exploration activity to support more realistic simulations. Line 4 initialises a $Prob$, which holds the calculated probability for each location, L_i in the set of Locations, L. Line 5 initialises the factor, k, used to reduce the influence of the scent strength at recently visited locations. The current maximum scent strength was set to 255, and thus the k value has been set to 10; if we simply used the inverse of the scent strength the influence of very recent directions will be negligible, 0.004 (1/255) as compared to 0.039 (10/255). Lines 7 to 9 assigns 1 to S_i if no locations in L has the mouse agent's scent. Lines 11 to 12 use k to normalise scent strengths as described above. Lines 15 to 16 calculate the probability of selecting each location in L, L_i with respect to the exploration activity using the dominant desire for this activity, d_e. Lines 18 to 25 use a probability number line to introduce randomness into the selection of a final location in L, L_{final}.

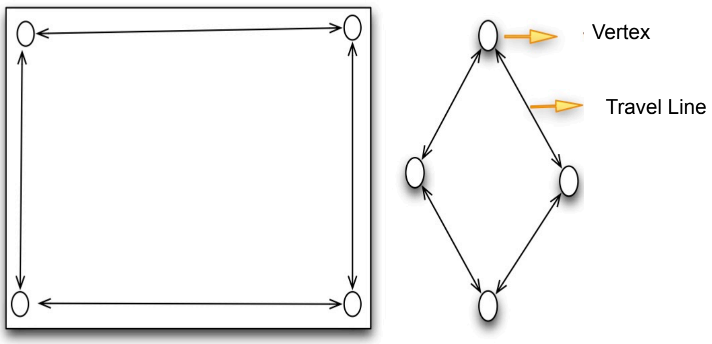


FIGURE 6.9: Mental Map for Simple Box

Algorithm 4: Realisation of Exploration Activity

Input: L = Set of Potential Locations
Input: d_e =Desire To Explore
Output: L_{final} = New location

$S = \{s_1, s_2, s_3 ... s_n\}$,set of inverse scent strengths, $|S|$=$|L|$;
$total = 0.0$;
$R = RandomNumberGenerator()$;
$Prob = 0.0$;
$k = 10$;
for $i = 0 \rightarrow |L|$ **do**
    if $L_i.ownScentStrength \equiv 0$ **then**
        $S_i = 1$;
    end
    else
        $S_i = k/L_i.ownScentStrength$;
        $total = total + S_i$;
    end
end
for $i = 0 \rightarrow |L|$ **do**
    $L_{i.explore} = (S_i/total) * d_e$
end
for $i = 0 \rightarrow |L|$ **do**
    $Prob = Prob + L_{i.explore}$;
    if $R < Prob$ **then**
        $L_{final} = L_i$;
        $break$;
    end
end
return (L_{final});

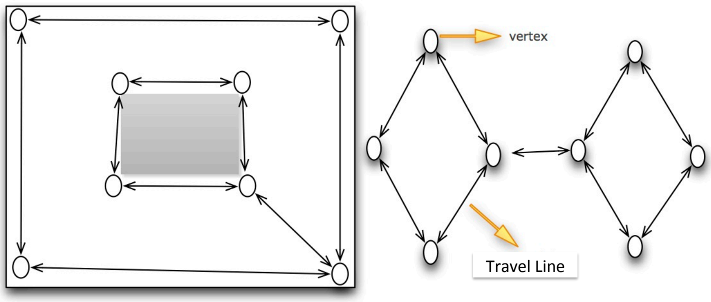


FIGURE 6.10: Mental Map for O-Box

6.6.3 Nest Site Discovery

The nest site discovery primitive activity is the process whereby a mouse agent discovers a "best" location in its environment to establish a nest. Best in this context is defined according to access to a given location, mice have a preference for nest sites that have limited access (for example in corner locations). The nest site discovery activity is driven by the desire to explore; it was deemed unnecessary to include a separate nest site discovery desire as this was covered by the desire to explore. While the mouse agent explores the environment locations of interest (waypoints) are tested for their suitability as a nest site.

For the purpose of the proposed mouse behaviour MABS the following assumptions were made in the context of nest site discovery: (i) each mouse agent is expected to identify and use one and only one nest site, (ii) two mouse agents cannot share a nest, (iii) two nest locations cannot be situated next to each other (a minimum distance r is required between nest locations, with the value for r determine as described above) and (iv) the best nest location is a available location that has the highest degree of safety.

The degree of safety of a location is defined in terms of accessibility which in turn is defined according to the GTIs of the eight neighbouring locations. More specifically wall (W), corner (C) and tunnel (T) locations are assigned a "safety weighting" of 1, 2 and 3 respectively, while space locations are assigned a weighting of 0. This sequence was chosen to reflect the ordering of desirability of these neighbouring locations. The degree of safety of a location is then calculated by summing the weightings of its neighbours and dividing by the number of neighbours:

$$Degree\ of\ Safety = \frac{Sum\ of\ Weightings\ of\ Neighbours}{Number\ of\ Neighbours} \tag{6.3}$$

Thus the idea is that every location in the environment has a degree of safety which is determined using equation 6.3.

The mouse agent selects a location which it identifies as having the best degree of safety as its nest location. This location is added to its "mental map" and is utilised when, for example, to hide from danger. Eligible nest locations must have a degree of safety greater than 0; so at start up, depending on the gate location, a mouse agent will typically not immediately find an appropriate nest site. A mouse agent may change its nest location if it finds a different location $L_{newNest}$ that has a higher degree of safety associated with it than the previous nest location $L_{oldNest}$. Where this is the case the mouse agent just replaces the former nest site with the better one.

Various experiments were conducted (see Chapter 7) whereby a mouse agent is placed in an environment which it then explores (at the same time creating its mental map) and then selects a nest site. From these experiments it was observed that in the case of a simple box environment (as expected) the mouse agent selected a corner location as its nest site. In the case of environments involving tunnels the agent (again as expected) would select a location towards the middle of the tunnel. An example of a nest site location chosen by mouse agents during evaluation exercises is shown in Figure 6.11 with respect to the box environment.

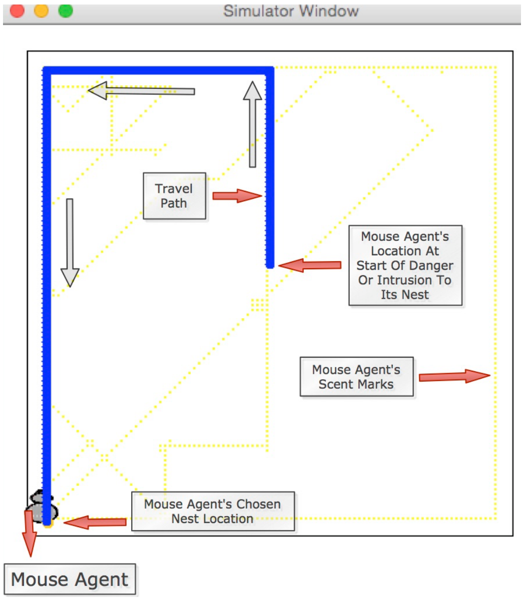


FIGURE 6.11: Mental Map for Simple Box

6.6.4 Safe Travel Route Identification

The way that agents generate route maps was described in Subsection 6.6.2, and how they are used to identify suitable nest sites in Subsection 6.6.3. Another important usage of the route map concept is with respect to the selection of safe travel routes in the context of danger, the Safe Travel Route Identification primitive activity. With respect to the proposed MsBMABS, in the event of danger, a mouse agent will wish to return to its nest site following an appropriate "best" route.

As noted previously the desire for safety is a dynamic desire. The default value is $d_s = 0.5$ and it will remain constant at this value unless the mouse agent perceives some form of danger in the environment when the desire for safety value will jump to the maximum value of $d_s = 1.0$. It will remain at this value until a threat is no longer perceived, in which case it steadily decreases to its initial value (strength) of $d_s = 0.5$. Note that the desire for safety always exists, it never drops below 0.5. When there is a jump from $d_s = 0.0$ to $d_s = 1.0$ the mouse agent uses its route map to travel to the safest known location in the environment (its nest location).

Thus in the event of danger a mouse agent needs to identify: (i) its nearest map way point and (ii) determine a best route to its nest constructed from the travel lines/paths (edges) in the route map. Rather than calculating all possible route permutations (of which there may be a great many) and selecting the shortest by distance, our mouse agents determine a best route as follows. Assuming a mouse agent is at waypoint wp_1 it will first check if wp_1 is connected directly to the desired nest location, in which case this travel line will form the "safe" travel route. Otherwise the mouse agent will select a follow on waypoint wp_2, connected to wp_1, which is nearest to the desired nest location. Nearest in this context is determined using (line of sight) Euclidean distance. The process continues in this manner until an appropriate route has been discovered. The result will not always be the shortest route but it will be a route that tends to feature safe locations because of the definition of way points (points of interest). Once a mouse agent has identified a safe route it will then attempt to travel to its nest locations using the identified safe route over the next n simulation iterations. The process might of course be interrupted by another external event such as another danger event or meeting another mouse event. Figure 6.11 illustrates an example safe travel route, with respect to a box environment, recorded as part of simulation runs used to evaluate the process for realising the safe travel route identification activity.

Algorithms 5 and 6 are used to describe the mechanism for realising the safe travel route identification primary activity. Algorithm 5 describes the process of finding nearest waypoints to nest, and Algorithm 6 describes the process for using the travel lines. The descriptions for both algorithms are provided below.

Description for Algorithm 5: The inputs to Algorithm 5 are (i) a set of waypoints which make up the mental map of a mouse agent, WP, (ii) the start waypoint, ($wp_i \in WP$), which is the first waypoint at the mouse agent finds and (iii) the current nest location, L_{nest} of the mouse agent. The output is TL, a set of locations between two waypoints which make up the travel line. Line 1 initialises, w_{ptr}, a pointer for the waypoint of mouse between its current and previous waypoints. Line 2 initialises a variable, wp_{i+1} which is the nearest waypoint to L_{nest}. Line 3 initialises variable L which holds travel line; the set of locations between a current waypoint, wp_i and the next waypoint wp_{i+1}. Lines 5 to 7 check if the current waypoint leads directly to L_{nest} in which case a travel line created to the nest. Lines 9 to 12 check for the next waypoint from wp_i which is nearest to L_{nest} in which case a travel line is created to wp_{i+1}.

Description for Algorithm 6: Algorithm 6 describes the mechanism whereby a new location is selected during the safe travel realisation activity. The inputs are: (i) set of locations, L, (ii) set of travel line locations, TL, and (iii) the desire for safety. The output from the algorithm is a new location, L_{final}. In line 1, variable N_n, the number of travel locations, in the set L is initialised. In line 2, variable N_s, the number of non travel line locations in the set L is initialised. In line 3, the total weightings for travel line locations in L, denoted by variable W_n is initialised. This value was set to 0.95 for simulations, because the mouse agent significantly prefers travel line locations to other locations in danger. In line 4, the total weightings for non travel line locations in L, denoted by variable W_s was initialised. This value is calculated as $1 - W_s$. Line 5 initialises a random number between 0 and 1. Line 6 initialises a $Prob$, which holds the calculated probability for each location, L_i in the set of Locations, L. In Lines 7 to 8 W_n is set to 1 where there are no non travel line locations in L. w_n is the weighting for each travel line location in L. Lines 10 to 11 calculate the weightings for individual locations (travel line and non travel line locations) in L, and stores in w_n and w_s respectively. In lines 15 to 18, the desire for safety (d_s) is used to calculate the preference for each location in L first by calculating preference for travel line locations in line 15, and non travel line locations in line 17. Lines 20 to 27 use a probability number line to introduce randomness into the selection of a final location in L, L_{final}, although randomness for this selection is highly limited due to the high bias in weighting towards W_n.

Algorithm 5: Algorithm for Identifying Safe Travel Lines

Input: WP = Set of waypoints (mental map)
Input: wp_i = Start waypoint
Input: L_{nest} =Current Nest Location
Output: TL = Set of Locations between two waypoints wp_i and wp_{i+1} which make up Travel Line

$wp_{ptr} = wp_i$;
wp_{i+1} = Nearest waypoint to nest from wp_i;
L = Variable which temporarily holds set of Locations between two waypoints wp_i and wp_{i+1} ;
while $wp_{ptr} \neq null$ **do**
    if $wp_{ptr} \rightarrow L_{nest}$ **then**
        $TL = L$;
        $Exit$;
    else
        $wp_{ptr} = wp_{i+1}$;
        $TL = L$;
    end
end
return TL;

6.6.5 Nest site defence

A nest site with respect to the proposed mouse behaviour MABS is conceptualised as a resting place and a place of safety. Mice are known to use nest locations to rest, hide from danger and of course for breeding. As noted above, in times of perceived danger, mice will normally attempt to get to their nest location by moving along a pre-identified safe travel route (see subsection 6.6.4). In addition mice will seek to protect their nest

Algorithm 6: Algorithm for Safe Travel Activity

Input: L = Set of Potential Locations
Input: TL = Set of Travel line Locations
Input: d_s =Desire For Safety
Output: L_{final} = New location

N_n = Number of travel line locations in L;
N_s = Number of non travel line locations In L;
W_n = Total weightings for travel line locations in L;
W_s = Total weightings for non travel line locations in L;
$R = RandomNumberGenerator()$;
$Prob = 0.0$;
if $N_s \equiv 0$ **then**
    $w_n = 1.0/N_n$;
else
    $w_n = W_n/N_n$;
    $w_s = W_s/N_s$;
end
for $i = 0 \to |L|$ **do**
    if *($L_i \equiv$ non travel line locations)* **then**
        $L_{i.safeTravel} = w_s * d_s$;
    else
        $L_{i.safeTravel} = w_n * d_s$;
    end
end
for $i = 0 \to |L|$ **do**
    $Prob = Prob + L_{i.safeTravel}$;
    if $R < Prob$ **then**
        $L_{final} = L_i$;
        break;
    end
end
return (L_{final});

sites from intruders. In real life, male mice are known to guard their nest sites from other rival male mice. Mice try to fend off intruders by exhibiting aggressive behaviour so as to discourage intruders from approaching. The nest site defence activity seeks to simulate this process.

The dominant desire for realising the nest site defence primitive activity is the "desire to guard nest location", d_g. Recall that the desire to guard the nest site is a dynamic desire. When the simulation commences, the strength of this desire is set to 0. This value however jumps to its maximum value of 1 when an intruder comes within a specified radius, rd of the mouse agent's nest. The strength of the desire to guard the nest starts to decrease once the intruder moves to a distance of greater than rd (thus $rd + 1.0$ or greater) from the nest site.

Algorithm 7, describes the mechanism whereby the nest defence primary activity is realised. The input is a set of potential locations, L, the desire to guard nest, d_g

with respect to the mouse agent and the location of the nest site, L_{nest} to be guarded. The output is a new location, L_{final} selected by the mouse agent. Line 1 initialises $L_{intruder}$, the current location of the intruder of the nest. Line 2 initialises L_{owner}, the current location of the mouse agent which owns nest, L_{nest}. Line 3 initialises the $Intruder_{intrusion.Threshold}$ which is a constant value for each simulation. It specifies how close a potential intruder must be to a nest, for an intrusion external event to begin (external events were discussed in Chapter 4). This value was set to 60.0 grid squares. Line 4 initialises the $Owner_{defence.Threshold}$ which is a constant value for each simulation run. It specifies how close a nest owner must be to its nest, to be able to notice an intrusion to its nest. This value was set to 120.0 grid squares. Lines 5 to 6 calculate distance between the nest owner and its nest L_{nest}, and between the intruder and the nest L_{nest}, respectively. Line 7 specifies the radius, rd, indicating locations around the nest. This is a constant value for each simulation and it was set to 60.0 grid squares. Line 8 calculates the number of locations in L, which are within the radius of the nest, N_w. Line 9 calculates the number of locations in L, outside the radius of the nest, N_o. Lines 10 and 11 specify the total weightings, W_w and W_o assigned to N_w and N_o respectively. Note that $W_o = 1 - W_w$; W_w was set to 90.0 for simulations conducted later in this thesis as it supported realistic simulation results. Lines 12 to 13, initialise w_w and w_o, weightings for individual locations in L, within radius, rd of nest and outside radius rd of nest respectively. Lines 14 initialises R a variable that holds a random number between 0 and 1, and it used to support randomness within nest defence primary activity realisation. Line 15 initialises a $Prob$, which holds the calculated probability (or preference) for each location, L_i in the set of locations, L. Line 16 describes conditions when an intrusion external event has occurred. Lines 17 to 19 describe the condition where no locations outside rd of nest, N_o exist in L, in which case $W_w = 1$. Lines 20 and 21 describe conditions where both N_w and N_o locations exist in L. Weightings are assigned to each location and they are calculated in lines 20 and 21. Lines 23 to 27 calculate the preference for each location, L_i in L using the desire to guard nest, d_g In lines 31 to 38, L_{final} is determined in a randomised probability driven manner using a number line, and a random number, R between 0 and 1.

6.7 Summary

This chapter has presented an overview of the way that the MBMBAS framework defined in Chapter 5, which in turn was founded on the abstract (Generic) MABS framework presented on Chapter 4, was used to implement a number of case studies. These case studies were categorised as follows:

1. Single Mouse Without Obstructions.
2. Single Mouse Exploring a Box With Obstructions.
3. Single Mouse Responding to Danger.

The chapter then established the implementation of these case studies in the context of the proposed MBMABS framework from Chapter 4 using five primary activities as follows:

1. Movement.
2. Exploration.
3. Nest site discovery.
4. Safe travel route identification.
5. Nest site defence.

The chapter then went on to consider how these primary activities could be realised using the proposed MBMABS framework. The chapter also considered a range of environments to be used to support the simulation. In the next chapter, the operation of the primary activities is evaluated in terms of a number of mechanisms.

Algorithm 7: Nest Site Defence Activity

Input: L = Set of Potential Locations
Input: d_g =Desire to guard nest
Input: L_{nest} = Current Location of Nest
Output: L_{final} = New location

$L_{intruder}$ = Current Location of Intruder;
L_{owner} = Current Location of nest owner;
$Intruder_{intrusion.Threshold}$ = Intrusion threshold, distance between intruder and nest, L_{nest};
$Owner_{defence.Threshold}$ = Defence threshold between nest owner and nest;
$Owner_{dist.to.nest} = L_{owner} - L_{nest}$;
$Intruder_{dist.to.nest} = L_{intruder} - L_{nest}$;
rd = Radius indicating area around nest, L_{nest};
N_w = Number of locations in L within rd of L_{nest} ;
N_o = Number of locations in L, outside rd of L_{nest};
W_w = Total weightings for locations in L within rd of L_{nest};
W_o = Total weightings for locations in L, outside rd of L_{nest};
w_w = Weighting for each location in L within rd of L_{nest};
w_o = Weighting for each location in L, outside rd of L_{nest};
$R = RandomNumberGenerator()$;
$Prob = 0.0$;
if $(Intruder_{dist.to.nest} < Intruder_{intrusion.Threshold}) and$ $(Owner_{dist.to.nest} < Owner_{defence.Threshold})$ **then**
    if $N_o \equiv 0$ **then**
        $w_w = 1.0/N_w$;
    else
        $w_w = W_w/N_w$;
        $w_o = W_o/N_o$;
    end
    for $i = 0 \rightarrow |L|$ **do**
        if *($L_i \equiv$ outside radius of nest locations)* **then**
            $L_{i.nestDefence} = w_o * d_g$;
        else
            $L_{i.nestDefence} = w_w * d_g$;
        end
    end
end
for $i = 0 \rightarrow |L|$ **do**
    $Prob = Prob + L_{i.movement}$;
    if $R < Prob$ **then**
        $L_{final} = L_i$;
        break;
    end
end
return (L_{final});

Chapter 7

Evaluation and Discussion

7.1 Introduction

In Chapter 4 an abstract (generic) MABS framework founded on the idea of the behaviour graph was presented. In Chapter 5 a multiagent based simulation framework for mammalian behaviour analysis, referred to as the Mammalian Behaviour MABS (MBMABS) was presented. The theoretical underpinning supporting the proposed MBMABS framework was derived from the abstract MAS framework presented in the foregoing chapter. This was then followed in Chapter 6 with a description of how the MBMABS could be used to realise mouse behaviour simulations in terms of an identified set of primary activities. In Chapter 6 the primary activities considered were grouped according to three categories of case studies. This chapter provides an evaluation of the mechanisms, in the context of the proposed MBMABS framework presented in Chapter 5, used to implement the primary activities considered in the foregoing chapter, Chapter 6.

The conducted evaluation reported in this section was organised around the categories of case study and primary activities reported on previously. The evaluation schedule is presented in Table 7.1. In the table the three categories of case study are listed in the first column and the associated primary activities in the second column. A number of different environments were identified in the previous chapter with a specific view to the evaluation presented in this chapter. Different environments were used with respect to the evaluation of the realisation of different primary activities (in each case the relevant environments were chosen so as to best support the evaluation of the primary activity in question). The fourth column in the table lists the nature of the evaluation conducted with respect to each case study category. Two types of evaluation were undertaken:

Corroboration: Corroboration was done by (i) comparing output from simulated experiments to video data recordings of the same experiment in real life (ii) recording scent trails and movement patterns using trace maps showing movement paths of the mouse and by confirming correctness of results with animal behaviourologists (domain experts) and (iii) visualisations of simulations again with reference to domain experts.

Consistency: Consistency checking was conducted by repeatedly running specific simulations using a range of parameters and analysing the outcomes recorded in terms of: (i) numerical measures and (ii) tracemaps.

Table 7.1: Case Studies and Associated Mouse Primary Behaviour

Case Study Category	Primary Behaviour/Activity	Environments Considered	Summary of Evaluation Method
Case study Category 1 - Single Mouse in box without obstructions	Movement (Thigmotaxis), Exploration, Nest site discovery	Box with no obstructions	(i) Comparison between Simulation and Real life experiment (using video)
Case Study Category 2 - Mouse in box with obstructions	Exploration, Nest site discovery	H-Box, O-Box, Tunnel Box, Maze	(i) Consistency (ii) Comparison between simulation and expert knowledge of animal behaviourists, (iii) Tracemaps
Case Study Category 3 - Mouse in box responding to danger	Safe travel identification, and Nest site defence	Box with no obstructions, O-Box, Tunnel Box	(i) Consistency and (ii) Tracemaps for movement

The remainder of this chapter is organised as follows. In Section 7.2, the video data acquisition and preparation process is discussed. In the following three Sections, 7.3, 7.4, 7.6, the evaluation with respect to each case study category is discussed.

7.2 Video Data Acquisition and Preparation

One of the three adopted corroboration evaluation mechanisms involved the comparison of simulated data from scenarios where real life video data was available concerning the same scenario. Reference to such video data has been made earlier in this thesis in Chapters 3 and 6; stills from video were given in Figures 3.2, and 6.2 . For convenience a further still is presented in Figure 7.1. Note that in the figure, as in the previously presented examples, the objects in each corner are nest boxes.

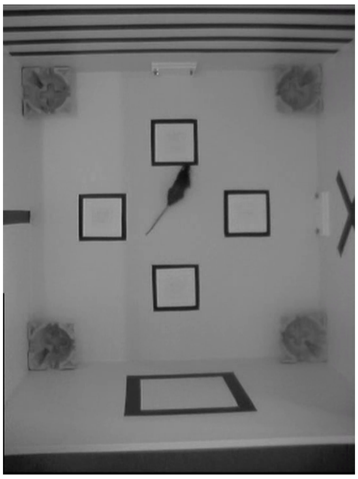


Figure 7.1: Example Video Still

The fundamental idea was that the number of times a mouse agent visited locations in a simulated environment should be comparable to the number of times a mouse visits the same location in comparable real life experiment; an idea suggested in [24] in the context of mining movement patterns from video data. For this purpose the cell size used by the individual cells in the MBMABS tile world environments were deemed to be too fine a definition of a location, instead cells were grouped into areas, effectively overlaying a second mesh over the simulation environment grid described earlier. The size of these areas was dependent on the MBMABS feature under investigation. Where a relative detailed feature was under consideration, for example when investigating the thigmotaxis characteristic of mice, a finer mesh was used then when considering some more general characteristic such as exploration. For "house keeping" purposes each area was allocated an ID number. GTIs were also allocated to each area where this was required by the investigation under consideration. In total over eight hours of video data was obtained, all featuring single mouse case studies.

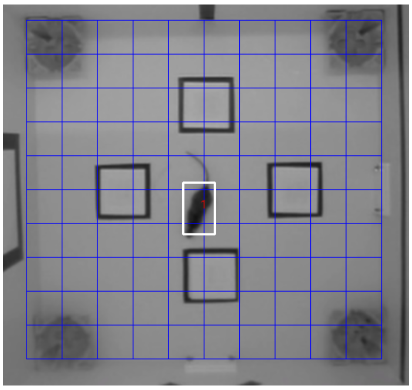


Figure 7.2: Example Mouse Video Still given in Figure 7.1 with a 10×10 grid imposed

The video analysis tool, the interface for which is shown in Figure 7.3, was first proposed in [24]. The tool is used to extract movement patterns from video and store results in a text file. This is done by first dividing the video environment into equal sized grid cells which can then be linearised by assigning sequential location IDs to each tile. The patterns collected are "from" and " to" locations of the object of interest (mice in our case).

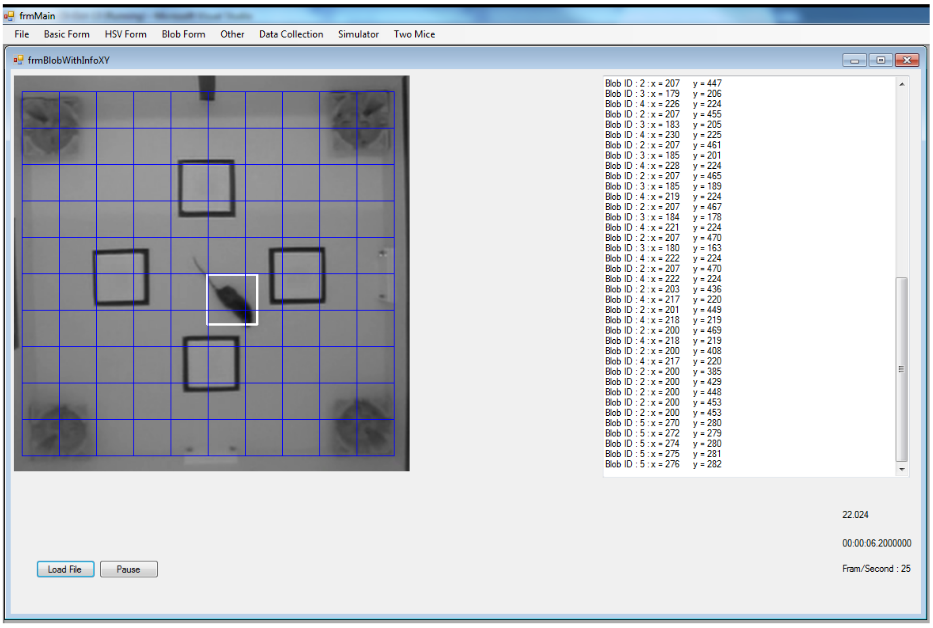


FIGURE 7.3: Video Analysis Tool

So that appropriate comparisons could be made the duration of a given simulation had to be matched to the associated video data and "area" occurrence counts recorded in both cases. The adopted video analysis tool automatically conducted location counting of the form described above and so was ideally suited to the purpose. Naively it might be thought that comparison between sets of occurrence counts could be done by summing the differences (positive and negative) for each pair of area occurrence counts and dividing by the total number of counts, but this would of course result in a value of 0. Instead absolute differences were calculated and divided by the number of areas.

7.3 Single Mouse Without Obstructions - Case Study Category 1

In this and the following three sections each of the three case study categories used for evaluation purposes is discussed. Each section considers the primary activities associated with the case study as identified earlier. For each primary activity the activity is reviewed and the objectives of the evaluation enumerated. The results obtained as a consequence of the evaluation are then presented and discussed in the context of the identified evaluation objectives. In this section the direction of movement, exploration and nest site discovery primary activities are considered. Recall that the single mouse without obstructions case study category considers the situation where a mouse is placed in a new environment (a box) which it is then expected to move around in and explore so as to learn about its environment (and produce a mental map) as described in Section 6.6 in the previous chapter. For the evaluation the simple box environment shown in Figure 7.4 was used.

The analysis with respect to of each of the above primary activities is presented in the following three subsections.

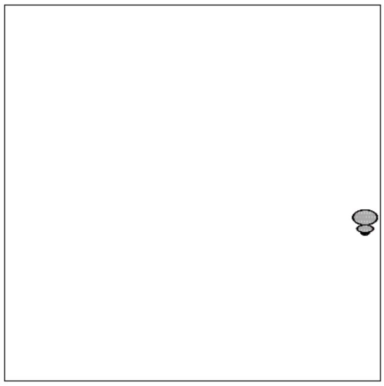


FIGURE 7.4: Box environment without obstructions

7.3.1 Evaluation of Movement (Thigmotaxis) - Mouse In A Box Without Obstructions

To test the effectiveness of the direction of movement primary activity comparisons were made with video data collected as described above. For this purpose a 10×10 grid (approximating to a 12cm $\times$12cm in the real environment) was used. This 10×10 grid is illustrated in Figure 7.2 with respect to the video still shown in Figure 7.1. A GTI set comprising three labels was used: wall, corner and open space. Note that the first two are non-space locations. The current location of the mouse agent was retrieved every second and recorded using the video analysis software described in [24]. Consequently, after 112 minutes of video time, 870 location counts had been obtained. Note that not all of the eight hours of available video data was suited to the analysis. The results obtained from the video analysis are given in Table 7.2. The table includes the number of visits (occurrence counts) for each GTI and the proportion of occurrence counts for each GTI with respect to the total number of areas. From Table 7.2 it can be seen that the mouse spends most of its time at corner locations (where the potential nest sites are).

TABLE 7.2: Recorded GTI label occurrence counts from video data using a 10×10 grid and a GTI label set of size 3

GTI	Occurrence Count	Number Of Areas	Proportion Of Total
Wall	209	24	24.0%
Corner	343	16	39.4%
Open Space	318	60	36.6%
Total	870	100	100%

For the simulation the same 10×10 grid as used for the video data capture, with the same GTIs, was superimposed on to the environment and data locations recorded with respect to this 10×10 grid. The simulation was run 100 times so that average values could be obtained.

Recall that thigmotaxis, a preference for walls, is a key feature of mouse movement behaviour (as discussed in Chapter 6). This was reflected by the desire for walls d_w included in a an agent's set of desires D. According to behaviourists, it is the most consistent and persistent activity in the behaviour of mice, and always plays a key role in movement selection. The objectives of the evaluation were thus to:

1. Test whether the number of non-space locations (areas) visited using the MBMABS framework was similar to the number of non-space locations visited in the video data.
2. To confirm that the number of number of non-space locations (areas) visited increased as the desire for walls d_w increased.

With respect to the second objective it should be recalled, from Chapter 6, that desires are measured using their desire strength, a value of between 0 and 1. Thus to test the effect of changing the desire for walls, (d_w) all other desires were held a 0.

The results for the comparison are presented in Table 7.3, the difference values included in the table were calculated with reference to the video occurrence count data presented previously in Table 7.2. The table lists the average number of visits, over 100 simulations, at wall, corner and open space locations in the same time frame as the video data. The table also lists the difference in number of visits in comparison to the video data. For reference the number of visits with respect to wall, corner and open space locations in the video data was 209, 343 and 318 respectively. The same data is presented in Figure 7.5 but in graphical form.

Table 7.3: Recorded GTI label average occurrence counts from 100 simulation runs using a 10×10 grid.

	Simulation Occurrence Counts			Difference With Video		
d_w Strength	Wall	Corner	Open Space	Wall	Corner	Open Space
0.1	206	319	345	3	24	36
0.4	217	326	327	8	17	9
0.7	227	329	314	18	14	4
1.0	230	342	298	21	2	20

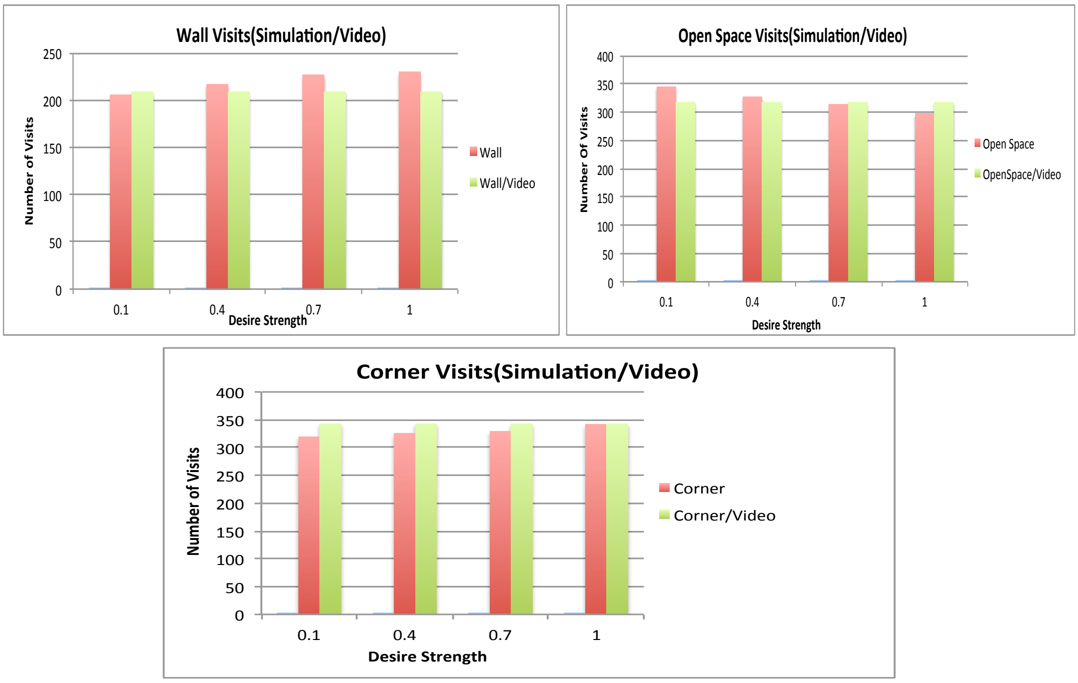


FIGURE 7.5: Box Environment No Obstructions

From Table 7.3 and Figures 7.5 it can be observed firstly that the number of simulation occurrence counts is similar to that recorded using the video data. Closer investigation of the results indicates that the most appropriate setting for d_w is between 0.4 and 0.7 because these values give the smallest difference with the video with respect to walls and corner location visits. As d_w is a static desire we can conclude that 0.5 is the most appropriate value.

In the context of evaluating the nature of the operation of d_w we can see, from the table and figures, that as anticipated the number of wall locations selected increases as the strength of d_w increases. At the maximum value of the desire for walls, i.e. 1.0, space locations are still selected by the mouse agent because as discussed in Chapter 4, the selection of states is randomised using a probability number line.

7.3.2 Evaluation of Exploration - Mouse In a Box Without Obstructions

As noted earlier, when real mice are placed in a new environment, they seek to explore it. The purpose of exploration is for the mouse agent to create a mental map of the environment, which is then used to navigate the environment when the mouse perceives danger. Recall also that the map consists of paths between locations of interest (waypoints). Exploration and the mental map concept was discussed extensively in Chapter 6 where it was noted that the desire to explore, d_e, was a dynamic desire. In other words it increases and decreases according to internal and/or external events. However it does have a starting value. The objectives of the evaluation presented in this sub-section were:

1. To establish that, in the context of exploration, the operation of the proposed MBMABS framework is comparable to the "real life" (mouse in a box) experiments.

2. To investigate and analyse the nature of the dynamic desire (d_e).

With respect to the second of the above evaluation objectives the aim was to investigate if varying the starting value of the desire strength will cause a significant change to the exploration pattern of the mouse agent. It was anticipated that so long as the conditions in the environment did not change, increasing the initial value of d_e, should not alter the pattern of exploration (and that this pattern should always be similar to the real world scenario).

The first objective was addressed by comparing the number of occurrences that each location was visited in the simulation with the number of times each location was visited in the video. In this case, unlike in the case of the evaluation of the direction of movement primary activity described above, a 5×5 grid was used because a coarser grained analysis was required (thus each area approximated to a 24cm $\times$24cm area in the real life environment). For reference purposes each area was given a sequential numeric ID from 1 to 25 (as shown in Figure 7.6). The second objective was addressed by considering a range of values for d_e from 0.1 to 0.9 incrementing in steps of 0.2 and comparing the occurrence count for each area obtained from the simulated data with those obtained from the video data.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

FIGURE 7.6: 5×5 grid area numbering

Table 7.4 shows the area occurrence counts obtained from the video data with respect to each of the 25 areas considered. Table 7.5 shows the area occurrence counts obtained from the simulation data and the calculated difference values. To support the interpretation of the results the same data as shown in Table 7.5 is given in graph form in Figure 7.7 and bar chart form in Figure7.8.

TABLE 7.4: Recorded GTI label occurrence counts from video data using a 5×5 grid

Area Code	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Occurrence Count	76	11	14	40	99	20	11	17	16	43
Area Code	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Occurrence Count	10	27	27	27	41	22	12	25	18	47
Area Code	**21**	**22**	**23**	**24**	**25**	**Total**				
Occurrence Count	66	30	27	42	102	870				

TABLE 7.5: Recorded average area occurrence counts from 100 simulation runs using a 5×5 grid

	Simulation Data					Difference with video data				
Area ID	d_e 0.1	d_e 0.3	d_e 0.5	d_e 0.7	d_e 0.9	d_e 0.1	d_e 0.3	d_e 0.5	d_e 0.7	d_e 0.9
1	68	88	73	87	72	8	12	3	11	4
2	20	23	21	28	18	9	12	10	17	7
3	20	27	16	26	24	6	13	2	12	10
4	30	41	30	46	47	10	1	10	6	6
5	72	78	73	88	78	27	21	26	11	21
6	41	45	32	23	38	21	25	12	3	18
7	4	4	7	6	2	7	6	4	5	9
8	4	12	14	16	21	13	5	3	1	4
9	16	10	9	14	6	0	6	7	2	10
10	41	42	34	45	51	2	1	9	2	8
11	58	19	19	24	12	48	9	9	14	2
12	8	14	20	18	6	19	13	7	9	21
13	4	11	0	5	37	23	16	27	22	10
14	15	10	13	20	36	12	17	14	7	9
15	66	45	53	56	50	25	4	12	15	9
16	46	63	46	46	45	24	41	24	24	23
17	8	8	17	14	17	4	4	5	2	9
18	12	14	25	19	27	13	11	0	6	2
19	11	14	23	16	12	7	4	5	2	6
20	53	56	62	57	51	6	9	15	10	3
21	93	67	88	57	59	27	1	22	9	7
22	32	33	32	18	21	2	3	2	12	9
23	26	35	28	28	27	1	8	3	1	0
24	48	39	45	36	37	6	3	3	6	5
25	74	72	90	77	76	28	30	3	25	26

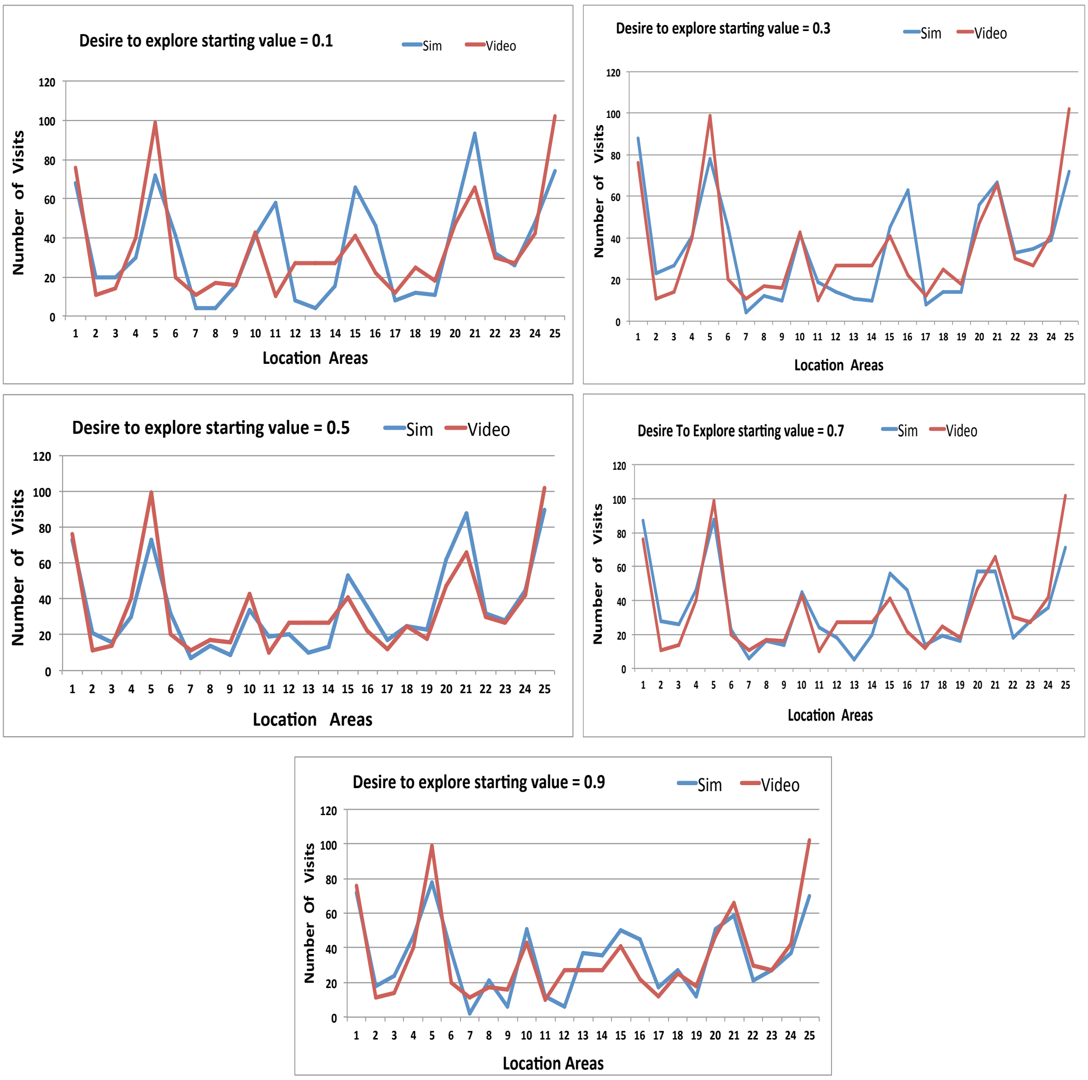


FIGURE 7.7: Data from Table 7.5 presented in graph form

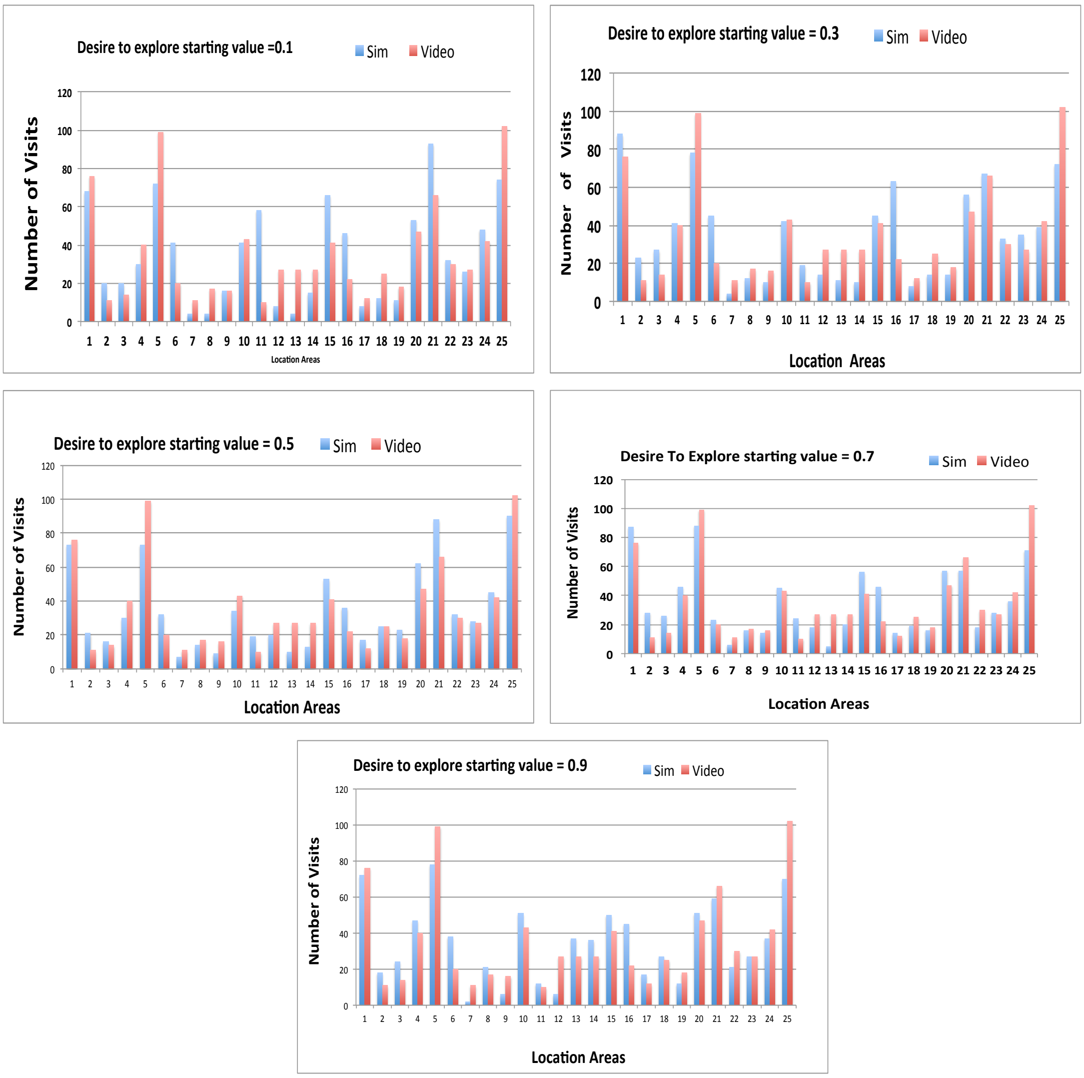


Figure 7.8: Data from Table 7.5 presented in bar chart form

From Table 7.5 (and Figures 7.7 and 7.8) it can be seen that, with respect to the first of the previously listed objectives the simulated mouse agent and the real-life mouse in the video behave in a similar manner. With respect to the second objective, to investigate the nature of the start value for d_e, it can be noted that changing the initial value of the desire strength does not alter the pattern of exploration, and this pattern should always be similar to the real world scenario.

7.3.3 Evaluation of Nest Site Discovery - Mouse In A Box Without Obstructions

The nest creation primitive activity was discussed in detail in Chapter 6. Briefly, the process by which mice identify a nest involves exploring the environment for suitable location areas based on the criteria also discussed in Chapter 6; namely the degree of safety offered by individual locations which in turn was measured by the level of accessibility of the location to intruders. For example, a corner location area would be less accessible than a wall location area, which would in turn be less accessible than an open space location. The anticipation was that locations considered suitable for nesting would be explored more than others. According to the consulted behaviourists, given a box with no obstructions of any kind the corner locations will be the most desirable nest sites. Thus four alternative nest site locations with little to differentiate them were included in the simulation; all would be equally suitable. In the real life experiment nest boxes were provided at each corner location (see Figure 7.1 given previously in Section 7.2); the mouse agent would be expected to choose one of them.

The objectives of the evaluation presented in this subsection were:

1. To determine whether the locations for nesting identified by mouse agents in the simulations were similar to those identified in the video data.

2. To determine the effect that changes in the start value for the desire to explore, d_e, had on the nest site discovery process.

For the evaluation the same 5×5 area grid used to evaluate the exploration primary activity was used (see Section 7.3.2).

The results from the video data analysis is presented in Table 7.6, while the comparison with the simulation data is given in Table 7.7. Table 7.8 also shows the average number of simulation data visits in comparison with the average number of video data visits. Table 7.9 then shows the number of occasions that each nest site was selected in the simulation (as opposed to the number of times each nest site was visited).

TABLE 7.6: Recorded GTI label occurrence counts from video data using a 5×5 grid

Nest Site Area ID	1	5	21	25	Total	Other Areas	Grand Total
Occurrence Count	76	99	66	102	343	527	870

TABLE 7.7: Recorded nest site area occurrence counts from simulated data using a 5×5 grid

	Simulation Data					Difference with video data				
Nest Site Area ID	d_e= 0.1	d_e= 0.3	d_e= 0.5	d_e= 0.7	d_e= 0.9	d_e= 0.1	d_e= 0.3	d_e= 0.5	d_e= 0.7	d_e= 0.9
1	68	88	73	87	72	8	12	3	11	4
5	72	78	73	88	78	27	21	26	10	21
21	93	67	88	57	59	27	1	22	9	7
25	74	72	90	77	76	28	30	12	25	26
Total	307	305	324	309	285	90	64	63	55	58
Average	76.75	76.25	81	77.25	71.25	22.5	16	15.75	13.75	14.5

Table 7.8: Average Number Of Visits To Suitable Nest Locations, $d_e = 0.5$

Suitable Nest Location Area ID	Average Number of Visits($d_e = 0.5$)	Video
1	73	76
2	73	99
21	88	66
25	90	102
Total Visits to Suitable Nest Location Areas	324	343
Number of visits to Other Location Areas(21)	546	527
Ratio of Suitable Nests To other locations	3.1:1.0	3.4:1.0

Table 7.9: Frequency Of Nest Location Area Selection

Suitable Location Area ID	Number of Selections For Nest($d_e = 0.5$)
1	15
5	23
21	23
25	31
Other Locations	8
Total Simulation Runs	100

Inspection of the results presented in Tables 7.7 and 7.8 indicates that the simulation results were similar to those obtained from the video data. With respect to the potential start values for the d_e desire it can also be seen that, as anticipated, this has little influence on the overall process. Consequently it was concluded that the operation of the simulation was similar to that of the real life video data (irrespective of the starting value of the d_e desire). However, from the results it is noticeable that in the video data some nest locations, for example nest area 25, are more popular than others indicating that there may be some other factors at play here. More specifically, the proximity of the gate location (entry point of the mouse into the environment); the mouse is likely to choose a nest site near to the gate location. The nearest suitable nest area to the gate location was nest area 25. With respect the simulation all four nest location areas are equally suitable. It is also interesting to note from Table 7.9 that on eight simulation occasions the mouse agent chooses a wall location instead of a corner location as its nest site. This is a feature of the randomness built into the MBMABS environment.

To illustrate that each simulation run is different Figure 7.9 presents a Four Box environment with four mouse agents in four boxes without obstructions. This is a hypothetical environment made up of four single mouse in a box environments of the form used above. In each case the “exploration” track followed by the mouse agent is indicated by the coloured lines, a different colour for each agent. In this particular example each of the four agents chooses a different corner for its nest location, of course this does not have to be the case. From the figure the random nature of operation of the proposed MBMABS can be clearly observed.

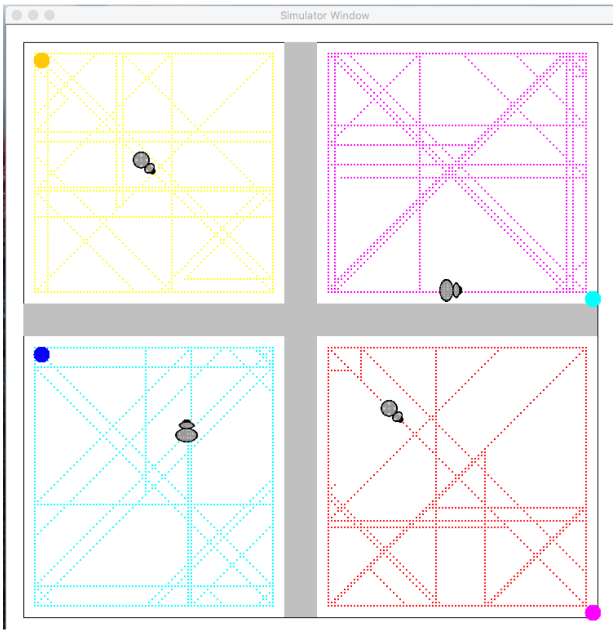


FIGURE 7.9: Nests Identification - Four Box Environment

7.4 Single Mouse in a Box With Obstructions - Case Study Category 2

This second case study category considered the situation where a mouse is placed in a new environment (a box) which it is expected to (i) explore and, (ii) identify a suitable location to create a nest site in a realistic manner. The primary activities involved are again:

1. Exploration
2. Nest site discovery

The distinction between the evaluation presented in this section with that presented in the previous section is that the evaluation involves obstructions. To this end a number of environments were used for the evaluation that feature obstructions. More specifically the "H", "O", tunnel and maze box environments of the form introduced earlier in Chapter 6. Note that the maze environment is particularly complex. Unfortunately, with respect to the second set of case studies, video data was not available because of the resource required to collect it. Instead the evaluation was conducted by:

- Comparing simulation results with expert knowledge from animal behaviourists. With respect to this point, the behaviour of mice was extensively discussed in general, in Chapter 3. Animal behaviourists directly connected to this research have highlighted that the exploration activity of mice involves seeking out interesting locations (waypoints) - which in turn were defined as locations where a change of direction might be required or might be appropriate, for example "corner locations". Mice seek out interesting locations because they are useful for nesting, and to be prepared for situations like danger, as was discussed in Chapter 6. Each interesting location (waypoint) is identified by the mouse agent within the simulation as a node on its mental map (the mental map was discussed in detail in Chapter 6). To specifically assist in the evaluation conducted in this case study, the behaviourists consulted identified locations which a mouse would consider suitable for nesting with respect to each environment and also the waypoints with respect to each environment. Figure 7.11 uses numeric labels in red to show the locations in each of the environments considered suitable for nesting, and Figure 7.10 uses numeric labels to show the locations in the environment considered by behaviourists to be interesting locations.

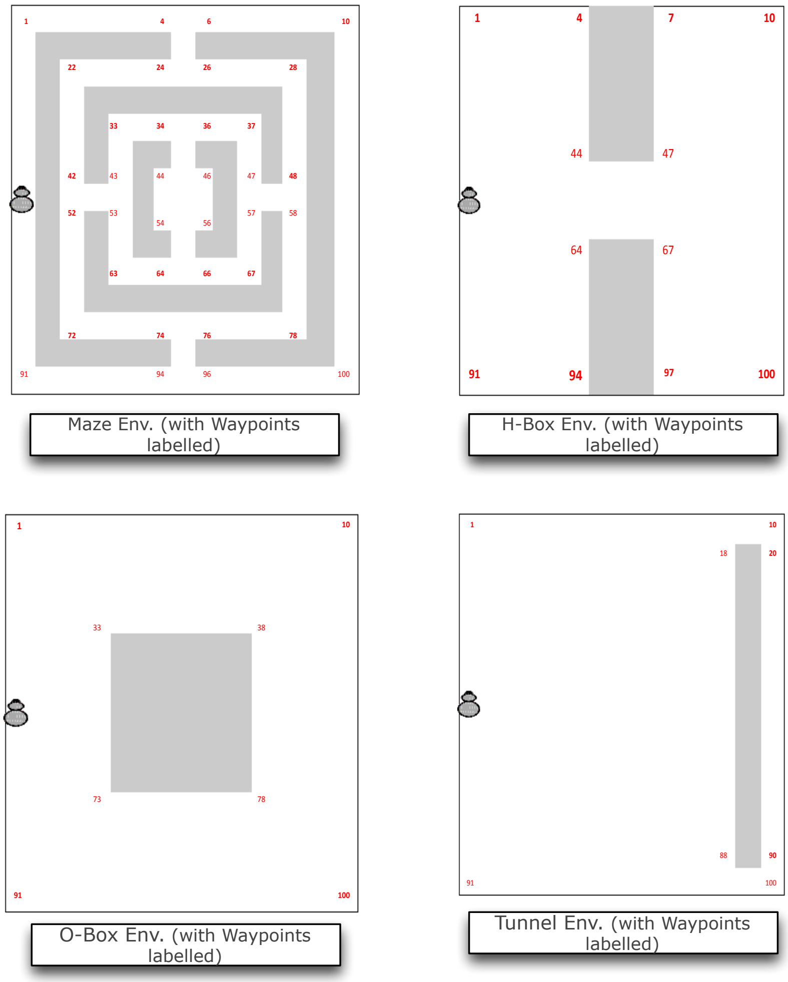


FIGURE 7.10: Environments For Case Study Category 2 with Labels In Red Indicating Waypoints As Highlighted By Animal Behaviourists

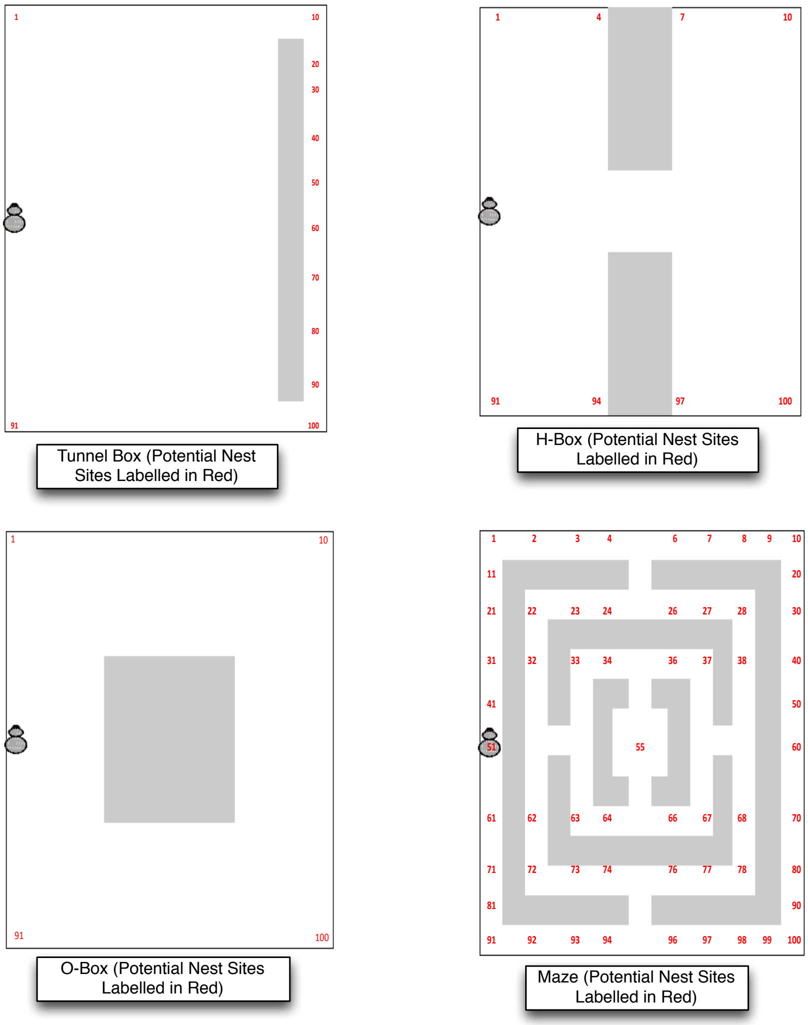


FIGURE 7.11: Environments For Case Study Category 2 with Labels In Red Indicating Suitable Nest Sites As Highlighted By Animal Behaviourists

- Checking for consistency by running the simulation repeatedly using a range of parameters and analysing the outcomes in terms of numerical measures. For example, in Subsection 7.4.2, the simulation was run repeatedly using different environments to see which locations were chosen by the mouse for its nest.

- Using tracemaps or movement traces showing what locations the mouse agent has visited in its environment, during the simulation.

As noted above, the simulation grid used to represent environments for case study category 2 was a 10×10 grid which comprised locations defined by x-y coordinates, each identified according to their ground types.

The set D was defined as follows $\{0.5, 0.5, 0.0, 0.0\}$ (walls, explore, safety, guarding). These values were chosen for the following reasons:

- The desire for walls was set to 0.5 because it replicated similar behaviour to the real life mouse as shown in Subsection 7.3.1.
- The desire to explore was set to 0.5 because it replicated similar behaviour to real life mouse as shown in Subsection 7.3.2.
- The desire for safety was set to 0.0 because it was not a significant parameter for this experiment.
- The desire to guard the nest site was set to 0.0 because it was not a significant parameter for this experiment.

Further discussion regarding the evaluation conducted in terms of case study category two is presented in the following two Subsections 7.4.1 and7.4.2.

7.4.1 Evaluation Of Exploration - Single Mouse in Box With Obstructions

The primary objective of the exploration primary activity, as noted previously, is for agents to explore their environment and create a mental map for later use in the event of danger, and for nest site discovery. The objective of the experiments presented in this sub-section was to conduct an analysis of how the exploration primary activity operates in the context of obstructions. This was carried out by recording the tracks that the agents follow and by considering the time required for the agents to generate a complete mental map. To this end four environments shown in Figure 7.10 were used.

For the evaluation each environment was divided into a 10×10 grid with each area number from 1 to 100. Each area marked in red was considered to be a potential nest site location. From the figure, it can also be seen that many other locations were considered unsuitable.

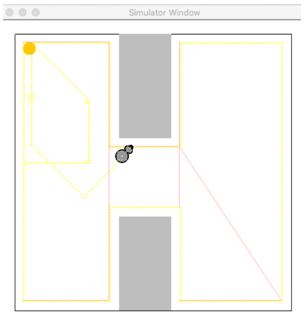


FIGURE 7.12: Nest and Map Identification - H-Box Environment (Orange dot indicates selected nest site location)

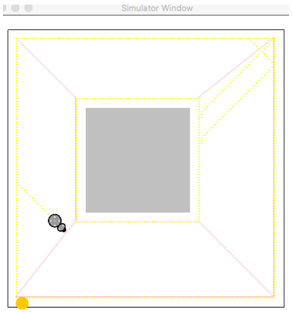


FIGURE 7.13: Nest and Map Identification - O-Box Environment (Orange dot indicates selected nest location)

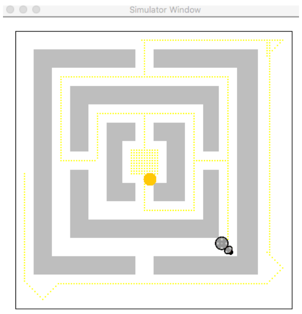


FIGURE 7.14: Nest Identification - Maze Environment (Orange dot indicates selected nest site location)

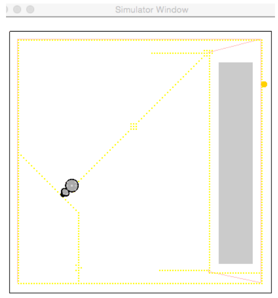


FIGURE 7.15: Nest Identification - Tunnel Environment (Orange dot indicates selected nest site location)

Figures 7.12, 7.13, 7.15 and 7.14 show indicators of example tracks followed by mouse agents to explore their environments. The yellow tracks in each environment are scent marks, but they also served as movement traces showing what locations the mouse agent visited while exploring. The red lines provide some illustration of how the mouse agent chooses to link interesting locations (waypoints) together to form a mental map of the environment. The grey objects within the environment are the obstructions. The simulation time, T, started from 0 for each experiment. The time taken for the mouse agent to find all the interesting locations in the environment, was recorded as the simulation progressed. The simulation was stopped after the mouse agent completed its mental map.

From the figures it can be observed from the yellow scent traces that the mouse agent regularly visits waypoints, consistent with the exploration activity. The exploration activity is sustained as long as new interesting locations are found.

Table 7.10 confirms that the time required for a mouse agent to create a mental map of its environment increases with the number of way points (map nodes) to be considered. Recall also from Chapter 6 that a way point represents what was termed an "interesting" location which in turn was defined as a location where a change of direction might be required or might be appropriate. Figure 7.16, shows the same data as shown in Table 7.10 but graphically so as to illustrate a consistent increase in the amount of time it takes to create the map, relative to the number of nodes in the environment.

Table 7.10: Mouse In Box With Obstructions -Mental Map Creation Time

Environment	Tunnel Box	O-Box	H-Box	Maze
Number of Nodes	6	8	12	36
Average Time to create map in Milliseconds	53892	76307.4	147784.5	918904.5

7.4.2 Evaluation Of Nest Site Discovery - Single Mouse in Box With Obstructions

The primary purpose of the nest discovery primary activity is for agents to identify suitable nesting locations. The objective of the experiments presented in this sub-section was to conduct an analysis of how the nest discovery primary activity operates in the context of obstructions. This was analysed by checking which locations were most consistently selected for nests, and also corroborating them with suitable nest locations as determined by behavioural experts as shown in Figure 7.11.

Figures 7.12,7.13, 7.15 and 7.14 show examples of nest selections. The orange dot-like markings, indicate the selected location of the nest.

Tables 7.12, 7.13, 7.11, 7.14 show the frequency of location selections of the mouse agent. The simulation was run 50 times for each environment. The tables show that the mouse agent consistently selects suitable locations as specified by behaviourists for its nest, significantly more than unsuitable locations for each environment.

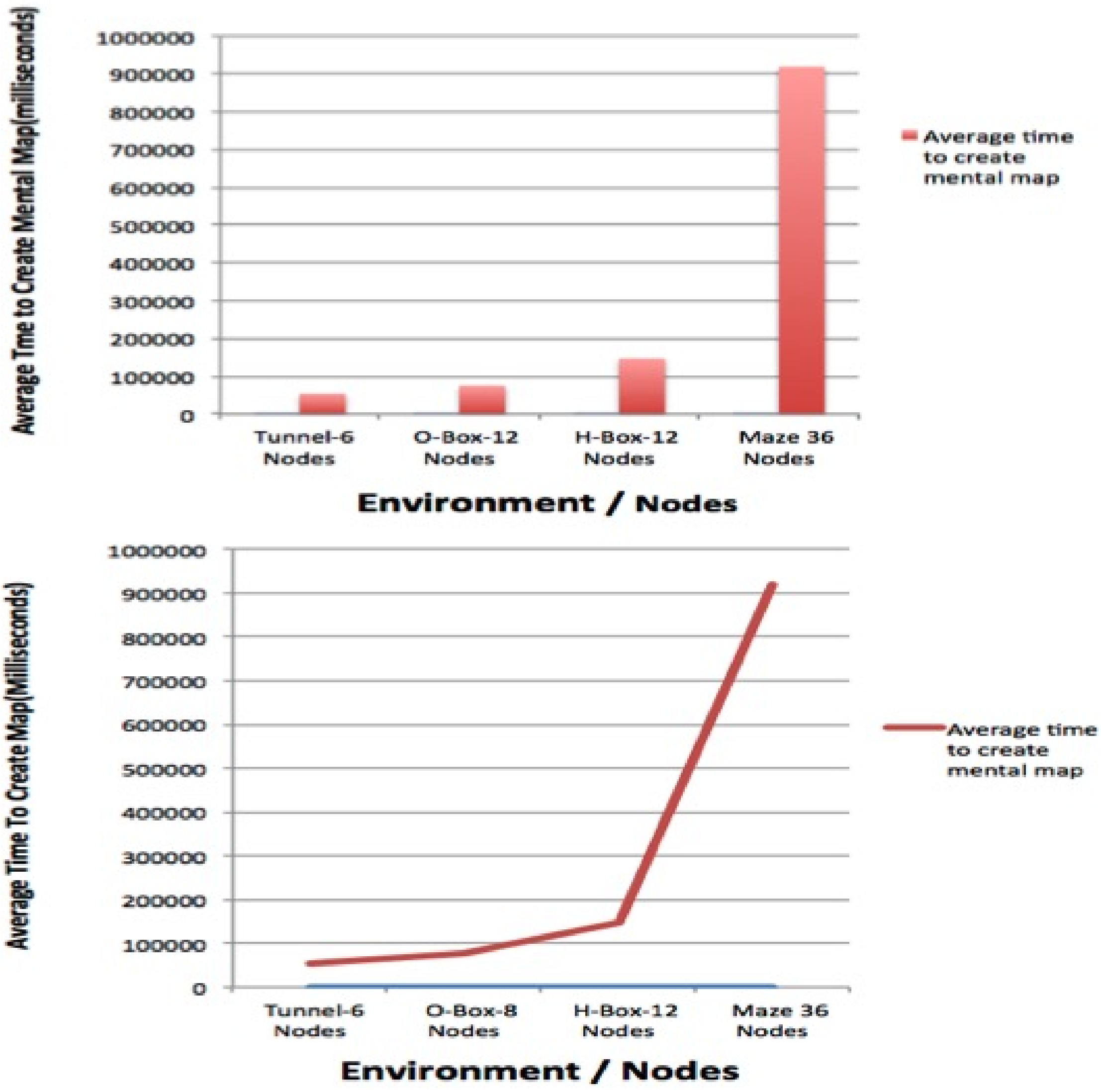


Figure 7.16: Data from Table 7.10 represented in bar chart and graph form

Table 7.11: Frequency Of Nest Location Area Selection For Tunnel Environment

Tunnel Environment	
Suitable Location Label	Number of selections
1	4
10	5
30	15
40	8
50	10
Other Locations (Unsuitable but selected)	8
Total	50

Table 7.12: Frequency Of Nest Location Area Selection For H-Box Environment

H-Box Environment	
Suitable Location Label	Number of Selections
1	14
4	5
91	10
94	11
97	2
7	1
Other Locations (Unsuitable but selected)	7
Total	50

Table 7.13: Frequency Of Nest Location Area Selection For O-Box Environment

O-Box Environment	
Suitable Location Label	Number of Selections
91	20
1	21
10	1
100	2
Unsuitable Locations	6
Total	50

Table 7.14: Frequency Of Nest Location Area Selection For Maze Environment

Maze Environment	
Suitable Location Label	Number of Selections
1	2
91	1
24	1
34	3
64	8
73	6
55	16
78	5
37	8
Other Locations (Unsuitable but selected)	0
Total	50

7.5 Mouse Responding To Danger - Case Study Category 3

As noted in Chapter 6 the third case study category was concerned with the two primary activities: (i) where a danger event occurs and the mouse agent needs to return to its nest site and (ii) where another agent appears in the vicinity of its nest site and it is expected to defend its nest site. In the first case the nature of the danger is not specified other than it necessitates the mouse agent to seek shelter in its identified nest by getting there as quickly as possible using its mental map. In other words the mouse agent has to be able to identify a safe travel route. In terms of the event categorisation presented earlier both events are external events (as opposed to internal events). Given the above, for the simulation to operate correctly, sufficient time had to be allowed for the mouse agent of interest to create its mental map and identify a nest site before the danger or other mouse intrusion event occurred.

In both cases the broad objectives of the evaluation were to:

1. Confirm that the implementation of primary activities was such that the mouse agents behaved as anticipated.
2. To compare the behaviour of the mouse agent with known mouse behaviour as observed by animal behaviourists.

The evaluation of each of the above primary activities, response to danger and response to mouse intrusion, are considered with respect to the above two broad objectives, in the following two sub-sections where the objectives are more specifically defined in each case.

7.6 Mouse Responding To Danger - Case Study Category 3

As noted in Chapter 6 the third case study category was concerned with the two primary activities: (i) where a danger event occurs and the mouse agent needs to return to its nest site and (ii) where another agent appears in the vicinity of its nest site and it is expected to defend its nest site. In the first case the nature of the danger is not specified other than that it necessitates the mouse agent to seek shelter in its identified nest site. In the second case the agent needs to return to its nest site to defend it. In both cases the mouse agent has to get to its nest site as quickly as possible using its mental map. In other words the mouse agent has to be able to identify a safe travel route from its mental map. Note that, in terms of the event categorisation presented earlier, both events are external events (as opposed to internal events). Given the above, for the simulation to operate correctly, sufficient time had to be allowed for the mouse agent of interest to create its mental map and identify a nest site before the danger or other mouse intrusion event can occur.

In both cases the broad objectives of the evaluation were to:

1. Confirm that the implementation of primary activities was such that the mouse agents behaved as anticipated (consistency checking).

2. To compare the behaviour of the mouse agent with known mouse behaviour as observed by animal behaviourists (corroboration checking).

The evaluation of each of the above primary activities, response to danger and response to mouse intrusion, are considered with respect to the above two broad objectives, in the following two sub-sections where the objectives are more specifically defined in each case.

7.6.1 Response to Danger

As noted above, response to danger entails safe travel route identification. How this was achieved in the context of the MBMABS framework was presented in Chapter 6. As also noted above, it should be recalled that safe travel route identification utilises the mental map created while mouse agents are exploring their environment.

For the evaluation a number of environments, with and without obstructions were used. More specifically the environments considered were: (i) a box with no obstructions, (ii) H-Box, (iii) O-Box and (iv) Tunnel Box (Figures 7.17, 7.18, 7.19, 7.20). With respect to each of these environments Figure 7.10 show the locations of waypoints that are expected to be included in the metal maps built up by individual agents.

Unfortunately there was no video data available for the response to danger case study. Instead, to evaluate the response to danger primary activity the paths travelled by mouse agents, given a danger event, were recorded and analysed. The expectation was that the safe routes travelled by mouse agents would be along walls and through waypoints on each agent's mental map.

For the evaluation the desire for walls d_w was set to 0.5, because it replicated similar behaviour to that displayed by real life mice as discussed in Subsection 7.3.1. On start up the desire to explore d_e was set to 0.5. The desire to defend the nest site d_g and the desire for safety d_s were both set to 0.0. Recall that when a danger event occurred the desire for safety, d_s increased to 1.0, hence d_s became the dominant desire.

Figure 7.17 and Figures 7.18, 7.19, and 7.20 show examples of recorded safe travel routes recoded with respect to each of the above environments. Starting with Figure 7.17 this shows an environment with no obstructions. The yellow dotted track shows the path followed while the mouse agent was exploring, creating its mental map and identifying an appropriate nest site, The identified nest site in Figure 7.17 is in the bottom left-hand corner. The blue solid line indicates that safe travel route followed by the mouse agent. When the danger event occurs the mouse agent is in “open space”, it moves to the nearest travel line in its mental map which item then follows to a waypoint. Once the first waypoint has been reached the next follow on waypoint is selected using the algorithm presented in Algorithms 5 and 6. Similar diagrams are presented in Figures

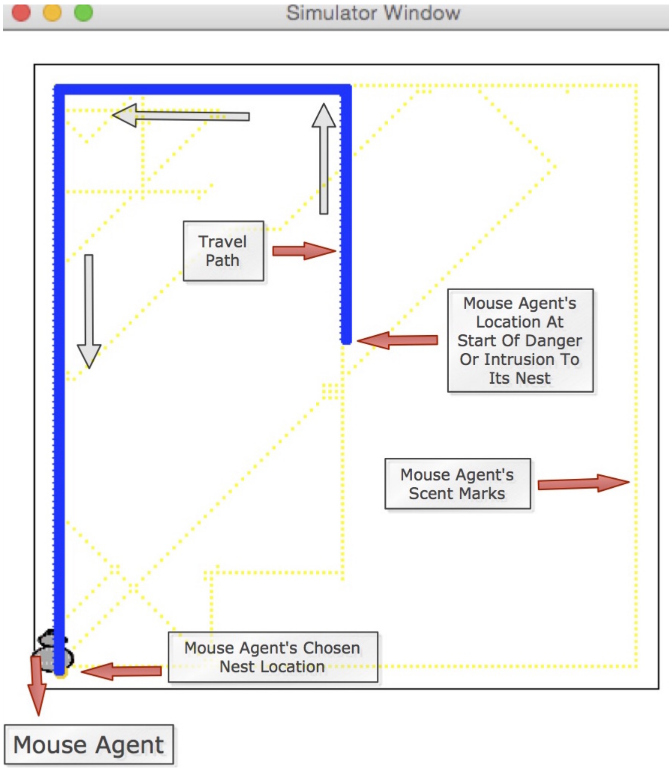


FIGURE 7.17: Use Of Mental Map/Safe Travel Route For Responding To Danger(Box without obstructions)

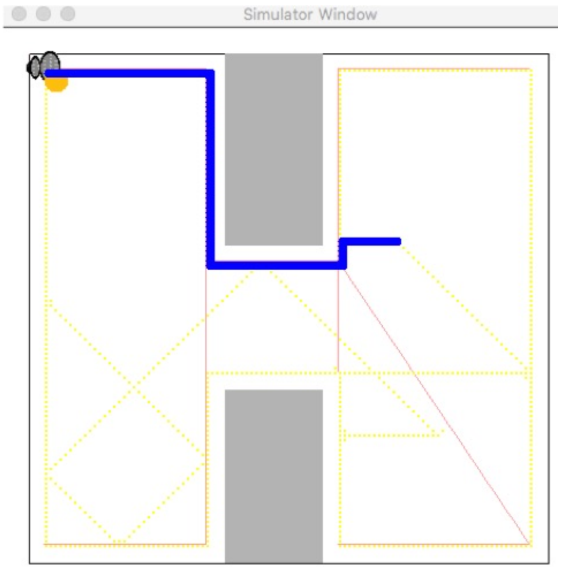


FIGURE 7.18: Response To Danger - H-Box Environment

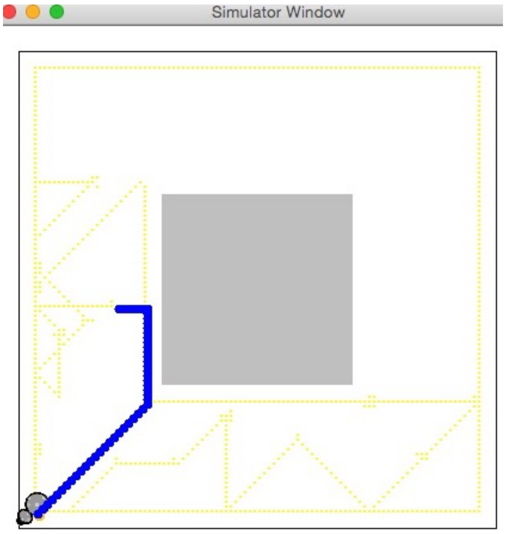


FIGURE 7.19: Response To Danger - O-Box Environment

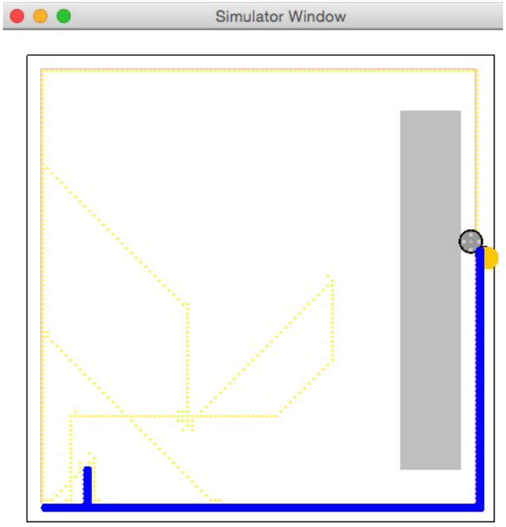


FIGURE 7.20: Response To Danger- Tunnel Environment

7.18, 7.19, and 7.20. Note that in in Figures 7.20, the mouse moves to the nearest point on a travel line, then chooses the closest waypoint, before travelling to the nest site.

From the diagrams, and other route traces not presented here, it was established that the simulation, with respect to the Response to Danger (safe travel route identification) primary activity, operated in a manner that was both correct and true to life.

7.6.2 Nest Defence

As noted above nest defence entails the primary activity where a mouse is expected to defend its nest site from intrusion from another mouse, which appears within the vicinity of its nest site. The objectives of the nest site evaluation have already been established above. As in the case of the Mouse Responding To Danger primary activity the nest site defence evaluation was conducted by using movement traces. For the evaluation simulations were run repeatedly to check for consistency in the mouse agent's behaviour.

The experimental settings were as before but in this case when the presence of another mouse agent is detected the desire to guard nest, d_g would increase to 1.0. Two mouse agents were used for the evaluations, both with identical desire configurations. The movement of each mouse agent was tracked. Experiments were also conducted with d_g disabled so as to provide a control. A number of example visualisations are presented in Figures 7.21, 7.22, 7.23, 7.24, and 7.25. In the figures the yellow movement trace is associated with mouse agent 1, and the purple trace with mouse agent 2. The orange nest is that associated with mouse agent 1, and the cyan nest with mouse agent 2. Figure 7.21 shows a visualisation where $d_g = 0$. From the figure it can be seen that neither mouse agent makes any attempt to guard its nest.

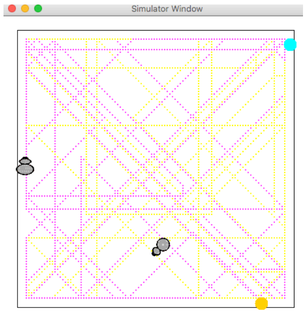


FIGURE 7.21: Absence of Nest Site Defence Desire ($d_g = 0$)

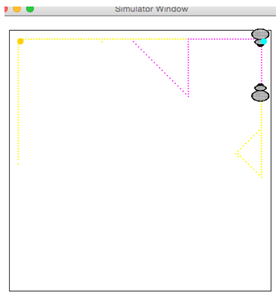


FIGURE 7.22: Nest Site Defence Sample Experiment 1

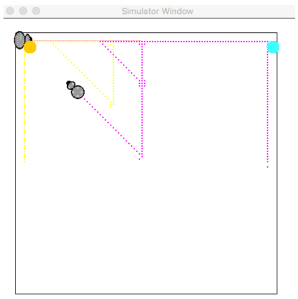


FIGURE 7.23: Nest Site Defence Sample Experiment 2

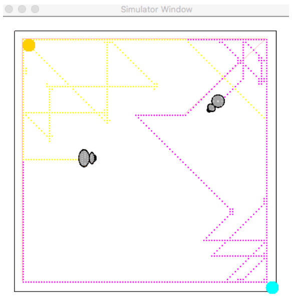


FIGURE 7.24: Nest Site Defence Sample Experiment 3

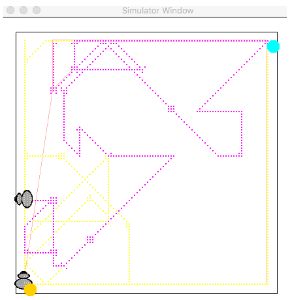


FIGURE 7.25: Nest Site Defence Sample Experiment 4

Figure 7.22 shows mouse agent 2, defending its nest from mouse agent 1. Figure 7.23 shows mouse agent 1 defending its nest from mouse agent 2. Figure 7.24 shows the areas visited by mouse agent 1 and mouse agent 2; from the yellow movement trace linked to mouse agent 1 and purple movement trace linked to mouse agent 2, it can be seen that agent 1 has been preventing intrusion to its nest area by staying close to its nest. Figure 7.25 shows how mouse agent 2 has moved significantly close to the nest area of mouse agent 1; the yellow movement trace and position of mouse agent 1 illustrates how mouse agent 1 has positioned itself to continue defending its nest from the intrusion by mouse agent 2. Overall analysis of the movement traces indicated that simulation of the nest defence primary activity operated as expected.

TABLE 7.15: Evaluation in the context of "Realism of Simulations"

Name	**Reference to Domain Experts**	**Visualisation**	**Usage of Video Data**
Movement	Domain experts confirmed that the expected behaviour was exhibited by MBMABS simulations with respect to movement. The mouse agent visited more wall and corner locations in comparison to open spaces (as expected).	Visualisation showed a degree of randomness in the movement of the mouse agents and that the mouse agents behaved as expected.	It was observed that the number of wall and corner (non-space) locations visited was similar to the number of non space locations visited in the video.
Exploration	Domain experts expect mice to explore a new environment to discover interesting locations and create a mental map of the environment to be used for safe travel and nesting. Domain experts confirmed that, as expected, enclosed spaces such as corners were particularly interesting to mouse agents.	Visualisation again showed that there was a degree of randomness in the manner of exploration by the mouse agents, and that the simulation was again as expected.	Figure 7.7 showed a comparison between the video data and simulation data, indicating similarities in the way exploration was conducted by the mouse agent and real life mice.

TABLE 7.16: Evaluations in the context of "Realism of Simulations" - Continued

Name	**Reference to Domain Experts**	**Visualisation**	**Usage of Video Data**
Nest Site Discovery	Domain experts identified suitable nest locations in the given environments, and confirmed suitable nest locations were selected during MBMABS simulations.	Visualisation of the simulations (Figures 7.12, 7.13, 7.15 7.14) indicated that the mouse agents chose suitable locations with respect to each environment.	The results from the video data analysis showed similarities in the number of visits to suitable nest location areas in both the video and simulation data (see Table 7.8).
Safe Travel Identification	The domain experts confirmed that, as anticipated, when there was danger introduced into a scenario, the mouse agent used the identified safe travel routes to return to its nest to its nest.	Visualisation (Figures 7.18, 7.19, 7.20, and 7.17) showed that the use of identified safe travel routes to escape danger in different environments resulted in realistic simulations.	No video data available.
Nest Site Defence	In the case of nest site defence, domain experts described the behaviour of mice when defending their territory. Mice will defend their nests by preventing intrusion around their nests. The domain experts confirmed that the simulated behaviour was as expected.	Visualisation of simulations (Figure 7.21) indicated that the movement traces of mouse agents resulted in realistic simulations.	No video data available

7.7 Summary

This chapter has provided an evaluation for the MBMABS Framework presented in the previous chapter. The evaluation was conducted in terms of the three case study categories introduced in Chapter 6, (i) mouse in a box without obstructions, (i) mouse in a box without obstructions, and (iii) response to danger. Recall that each category incorporated one or more primary activities (in turn comprised of secondary activities represented by states in a behaviour graph). The primary activities considered were: (i) movement, (ii) exploration, (iii) nest site discovery, (iv) safe travel route identification which was used to respond to danger and (v) nest defence. The evaluation was conducted using a number of different mechanisms: (i) comparison with video data, (ii) analysis of movement traces and (iii) analysis of statistical (numeric) outputs from simulation runs. The results indicated that the simulations operated in a manner consistent with real life "mouse-in-a-box" experimentation, and that the mechanism used to realise the simulations (the behaviour graph concept, desires and so on) functioned in an appropriate manner. Tables 7.15 and 7.16 present a summary of how realism was determined from the evaluations conducted. In Chapter 8 this thesis is concluded with a summary and some ideas for future work.

Chapter 8

Conclusion

8.1 Introduction

This thesis set out to investigate the nature of a MABS Framework to support the simulation of animal behaviour, with a focus on mice behaviour. Mechanisms whereby a Mammalian (Mouse) Behaviour Multi-Agent Based Simulation (MBMABS) could be realised were therefore investigated and reported on. The central idea was the concept of the behaviour graph, a form of finite state machine, where vertices are states and edges represent state changes, specifically directed at mammalian behaviour study including the concept of randomness. The behaviour graph operated using the notion of desires and events. With respect to desires the behaviour graph concept had some parallels with BDI frameworks which have also, on occasion, been used for simulation purposes. The reason for the work being directed at the usage of MABS was because such systems are particularly well suited to situations which involve the modelling of self-deterministic entities such as animals or humans. The key question that the research presented in this thesis sought to address was:

How can multi-agent based technology best be used to simulate animal, especially rodent, behaviour in as realistic a manner as possible?

This chapter provides a summary of the solutions proffered in this thesis with respect to the above research question. The structure of this chapter is as follows. In Section 8.2 an overall summary of the research presented in this thesis is provided. In Section 8.3 the main findings and contributions of the thesis are presented in terms of the research question and research issues identified earlier in the thesis. The chapter, and the thesis, is then concluded with Section 8.4 which itemises some directions for further research.

8.2 Summary

In the thesis a MBMABS framework is proposed, discussed and evaluated. Central to the framework was the concept of the behaviour graph comprised of vertices representing states and edges representing possible state changes. The basic idea of the MBMABS framework was that each "player" (animal) is represented by an agent. The behaviour for each agent is encapsulated in terms of a set of desires (D) and a behaviour graph (B). The development of the behaviour graph concept was presented by first considering the concept in the abstract, without committing to a particular category of animal or particular scenarios. To this end the concept was discussed in generic terms in Chapter 4. This chapter also presented an evaluation of the generic framework. This evaluation indicated that:

- With respect to the scalability of the behaviour graph, as the number of agents interacting with the behaviour graph increased, the performance of the behaviour graph was affected in the sense that it required more resources to operate optimally.
- Increasing the number of vertices also leads to a requirement for improved system resources.
- The proposed generic framework successfully allows a collection of agents to operate within a simulation environment.
- The usage of the desire concept could be effectively used to select follow on states.
- The usage of the behaviour graph concept could be effectively used to model the behaviour of agents (at least in the abstract context).

Therefore the evaluation indicated that a sound foundation had been established for the later work directed at more domain specific kinds of MABS.

Chapter 4 then considered the extension (development) of the generic MABS framework presented in Chapter 4 by considering its application in terms of mammalian behaviour. To this end four categories of agent were identified: (i) character agents (ii) static agents (iii) housekeeping agents and (iv) utility agents; of which the character agent was the most significant.

Character agents were defined as having six main attributes:

1. A set of desires, D.
2. A location within the environment expressed in terms of x-y coordinates referenced to the origin on the environment (tile world) in which it will operate.
3. A current state, defined by a vertex in the behaviour graph associated with the character agent.
4. A "stateTime".
5. A direction in which it is facing or travelling in (expressed in terms of the four cardinal (north, south, east, west) and four inter-cardinal (north west, north east, south west and south east) directions.
6. An identification number.

The environments in which the simulations operated, represented by environment agents (one per simulation), were defined in terms of a bounded tile world E measuring $x \times y$ tiles. To indicate a particular tile in E located at x-coordinate i and y-coordinate j the notation e_{ij} was used. Each tile e, except at the boundary of the environment, had eight neighbours. Thus, in a single simulation iteration, and in the absence of any obstructions, an agent at some location e_{ij} was free to move to any of its neighbouring tiles. In other words, agents had eight degrees of movement. The tiles in an environment represented a variety of "ground types" each indicated by what was termed a Ground Type Identifier (GTI); a label drawn from a set of labels L. For the case studies the set L comprised "wall", "corner" (also known as choice points), "tunnel", "open space".

The MBMABS framework presented in Chapter 5 was then further developed in Chapter 6 by considering a number of case studies. In total three case study categories were considered: (i) mouse in a box without obstructions, (ii) mouse in a box with obstructions and (iii) mouse in a box responding to danger. The activities featured in the case studies were defined in terms of primary activities and secondary activities. Primary activities comprise secondary activities which in turn were defined by states within the behaviour graph. In total five primary activities were considered:

1. Movement
2. Exploration
3. Nest site discovery
4. Safe travel route identification
5. Nest site defence

The evaluation of the proposed MBMABS Framework was then presented in Chapter 7. The evaluation was conducted in terms of the case studies and primary activities identified in the previous chapter. The case studies were modelled around real life "mouse in a box experiments" conducted by the Mammalian Behaviour and Evolution (MBE) group, Institute of Integrative Biology at the University of Liverpool. This in turn meant that the simulations could be referenced to real life experiments. This was done by: (i) comparing the operation of simulated experiments with video data of the same experiment conducted in real life, (ii) demonstrations to domain experts (from the MBE group), (iii) recording of the behaviour of mouse agents in simulations by storing paths followed and interpreting this in terms of "correctness". The effect of various parameter settings was also considered. The main findings from this reported evaluation are considered in the next section.

8.3 Contribution and Main Findings

In this section the main findings of the work considered in this thesis are presented with reference to the research question and research issues identified in Chapter 1. The section is organised as follows. We commence by considering each of the previously itemised research issues in turn and then return to the central research question. Each of the research issues is itemised and discussed below. In each case the relevant research issue is first presented in the same way as it was presented in Chapter 1 and then discussed.

1. *Given that each agent (entity) within a MABS will posses a particular set of features and traits, a suitable mechanism whereby these features and traits can be represented, in a well structured manner, was required. The central idea here, as noted above, is the usage of a behaviour graph, although the nature of this graph was unclear at commencement of the programme of research.*

   The nature of the proposed behaviour graph was fully investigated and defined in the thesis. The behaviour graph comprises a set of states s. Each sate has a set of possible state changes associated with it. The idea presented in this thesis was that state changes were affected by desires and events where events, in turn, could be internal or external (self directed by individual character agents or out of the immediate control of such agents). Internal events occur as a result of an agent completing some self appointed task, for example changing the direction movement, or when the current state "times out". Timing out is concerned with the duration whereby an agent may remain in a particular state; agents are assumed to be unable to remain in any one state indefinitely. With respect to agent features and traits the idea formulated in the thesis was that features of agents, within the proposed MBMABS, would be represented by attributes and traits by desires. To this end two types of desire were identified, static desires and dynamic desires.

2. *As noted in the introduction to this chapter an important element of the proposed behaviour graph structure is the concept of desires. The idea here is that desires will affect the operation of the behaviour graph when invoking "state changes", although how this would operate was a matter for the research.*

   The nature of desires was investigated by considering a set of case studies. The idea being to use these case studies to drive the derivation of the nature of the desires to be incorporated into the desired MBMABS framework so that individual case studies could be realised. To this end 4 desires were identified: (i) Desire for walls (d_w), (ii) desire to explore (d_e), (iii)desire for safety (d_s) and (iv) desire to guard nest (d_g).

   The fundamental idea was that desires would influence the selection of follow on states. However, from consideration of the primary activities associated with the various case studies it was clear that some follow states were associated with particular desires while others had no relevance with respect to particular desires. The mechanism identified for taking this into consideration was to weight follow on states with respect to particular desires. Thus each follow on state would be allocated a set of weightings W such that there was a one to one correspondence between each element in W and each element in D. Using the contents of the set

D associated within a particular character agent and the sets W associated with two or more follow on states a preference value could be calculated for each follow on state. However, a requirement for the proposed MBMABS was a degree of randomness, the simulation should not operate in exactly the same manner on each simulation run. Consequently a random probabilistic mechanism was derived for selecting follow on states which the conducted evaluation demonstrated produced realistic simulations.

3. *Following on from (2) above it was also unclear how desires would be encapsulated and how they would change with time as a simulation progressed.*

   From the above each character agent possessed a set of desires D. From investigation of the case studies considered, and the associated primary activities, it was clear that these desires could be static or dynamic. Each desire had a desire "strength" associated with it, a number between 0.0 and 1.0. In the case of static desires, the value would remain constant. In the case of dynamic desires this would change. From investigation with respect to the primary activities associated with the identified case studies it was found that the strengths associated with desires could change in a variety of ways. They could increase or decrease gradually, or they could jump. The distinction can be illustrated by considering the desire to explore; this desire gradually decrease if no new "interesting locations" (waypoints) were found, but would jump back to its start level if a new point was found. In other words it was found that dynamic desires should be influenced by both internal and external events. Recall that internal events are controlled by the character agents themselves, while external events are the opposite. An agent's desire(s) influenced its state changes.

4. *Individual agents will need to be able to make autonomous decisions based on their surroundings and desires; some appropriate mechanism for doing this would therefore also need to be incorporated.*

   The level of autonomy exhibited by the agents in a MABS of any kind is an important one. From the foregoing it was found that the most appropriate mechanism for allowing individual agents to make autonomous decisions was through the concept of static and dynamic desires to influence state changes. The exception to this was with respect to the direction of movement selection primary activity. This was found to be such a common activity that, although it could have been implemented in terms of states, a specific mechanism was derived for achieving this. An agent will have n directions to move in where n is defined by the number of immediate neighbour tiles into which the agent can legally move; a maximum number of 8 possible tiles. Using the agent's desires and the individual tile GTIs a set of preferences were derived which in turn were used to direct movement.

5. *The agents will exist in an environment, possibly an agent in its own right, which will also have certain features associated with it; appropriate techniques would be*

required to represent such environments, and the interface with the activities of other kinds of agents.

In tune with the spirit of MABS it was deemed appropriate to consider all elements of the proposed MBMABS to be agents even if in some cases the agents would be "dumb" agents in the sense that they did not do anything other than exist (and have the ability to be queried by other agents). Thus the simulation environments to be considered were modelled as agents, only one environment agent per simulation. Environment agents could be queried by character agents with respect to the GTI at a particular location. Similarly obstructions were defined in terms of agents linked to environment agents.

6. *The desired animal behaviour MABS, unlike the kind of problem solving usually conducted with respect to more standard forms of MAS, needed to feature a degree of randomness; the MABS agents should sometimes behave in an unexpected manner because this is what animals do in real life. Some mechanism for achieving this would thus also be required so that such randomness could be built into the MABS.*

   Randomness was an important requirement for the desired MBMABS, character agents should, on occasion behave in an unexpected manner because this was a feature of real-life experiments. This was realised as described above in terms of the random probabilistic mechanism derived for follow on state selection. The significance of this randomness was that simulations featured a degree of serendipity without which their operation would be unrealistic. For example with respect to the desire to explore if an agent always selects wall locations at the edges of an environment, because of the thigmotaxis desire, obstructions located in the middle of the environment would never be found.

7. *As also already noted, any solution to the above issues must be scalable; scalability was thus also identified, in its own right, as a research issue requiring investigation in the context of this thesis.*

   Scalability is of course a desirable feature of any MABS. This was explored early on in the thesis with respect to the abstract (generic) version of the eventually established MBMABS framework. The abstract behaviour graph was evaluated in terms of its scalability and it was observed that up to 9000 agents and a behaviour graph comprising 900 vertices did not significantly impact on performance. By extension this finding is therefore also applicable to the MBMABS framework derived from the abstract framework.

8.4 Limitations and Future Work

The work presented in this thesis has demonstrated that the proposed mechanisms and processes can be used to realise a MBMABS in a manner that leads to realistic simulations. However, the identified MBMABS framework featured a number of deficiencies and simplifications which remain to be addressed so that the proposed framework can be more beneficial and wide ranging. These are considered in this section as follows.

1. **Direction of movement selection**: The current MBMABS framework incorporates the simplification that character agents only move in the cardinal and inter-cardinal directions. In the context of the mouse in a box case studies, provided the box is aligned in the cardinal directions, this produces reasonably realistic simulations. However, a more realistic simulation would clearly result if character agents had 360° of movement. Investigation of how this might be implemented is thus a desirable avenue for future work.

2. **Velocity of travel**: Currently mouse agents travel at a constant velocity (or are stopped). As in the case of direction of travel selection, better simulations would result if the velocity of travel was not constant. For example in the case of a danger event the mouse agent would be expected to travel back to its nest at greater speed than when (say) it was simply exploring. Mechanisms for incorporating a variety of velocities of travel are thus also a desirable area for future investigation.

3. **More extensive evaluation using video data**: To date evaluation has been conducted using video data from mouse in a box case studies without obstructions. Video data for mouse in a box case studies with obstructions was not available because of the resource required to obtain this data. However, more wide ranging evaluation, using further video data would clearly be desirable in general, and specifically to support larger scale evaluation involving two or more mice for each case study.

4. **Further Primary activities**: Only a number of primary activities were considered within the context of the thesis. Although the activities considered sufficed to support the development of the various techniques and mechanism to support the implementation of such activities, the study of further primary activity would add further benefit. Examples of additional primary activities that night be considered include hunting for food (driven by a dynamic hunger desire) and sleeping (driven by a dynamic sleep desire). Investigation of the implementation of such further activities would be beneficial as it would allow for a more comprehensive range of simulations to be undertaken.

5. **Scent marking**: Early on in the thesis it was noted that mice mark their territory with scent. Male mice also attempt to obliterate the scent markings of competitor mice. Primary activities involving scent were not explored in the thesis (although the use of scent marking was included for selecting new locations during the exploration primary activity), yet this is an important element of mouse behaviour. Further work on how such activities can be incorporated into the MBMABS framework would be beneficial.

6. **Female Mouse Agents**: Only male mouse behaviour was considered in this thesis. It will be desirable to include female mice into the study, by studying and adding female mouse agents to integrate real life female mice attributes and behaviour. Although this will add additional complexity to the MBMABS framework. it will lead to better simulation activities.

7. **Comparison to other related work**: It would have been desirable to also implement the MBMABS framework by extending an existing framework, for example NetLogo, and then compare the results obtained. Additionally, with respect to agent behaviour, it would also be desirable to investigate an extended BDI mechanism which can be compared to the MBMABS framework. Pure BDI agents assume agent behaviour to be rational. The pure BDI framework was thus not considered to be best suited animal behaviour simulation, because animal behaviour includes irrational components, most significantly, emotions. An interesting avenue for future work would thus be to investigate the usage of these alternative mechanisms.

8. **Scalability**: The abstract behaviour graph was evaluated in terms of its scalability and it was observed that up to 9000 agents and a behaviour graph comprising 900 vertices did not significantly impact on performance. It would however be desirable to increase the number of agents and vertices to better identify scalability constraints.

In conclusion it is suggested that the proposed MBMABS framework, has provided a useful proof of concept approach to MBMABS that is likely to have wide ranging benefits. Domain experts have confirmed that the MBMABS framework provides a useful mechanism whereby computational agents can be utilised to support the prediction of the behaviour of rodents. A supporting letter from the domain experts consulted as the work in this thesis progressed is included in Appendix F.

Appendix A

Towards Large-Scale Multi-Agent Based Rodent Simulation: The "Mice In A Box" Scenario

Towards Large-Scale Multi-Agent Based Rodent Simulation: The "Mice In A Box" Scenario

E. Agiriga, F. Coenen, J. Hurst, R. Beynon, D. Kowalski

Abstract Some initial research concerning the provision of a Multi-Agent Based Simulation (MABS) frameworks, to support rodent simulation, is presented. The issues discussed include the representation of: (i) the environment and the characters that interact with the environment, (ii) the nature of the "intelligence" that these characters might posses and (iii) the mechanisms whereby characters interact with environments and each other. Two categories of character are identified: "dumb characters" and "smart characters", the obvious distinction being that the first posses no intelligence while the second have at least some sort of reasoning capability. The focus of the discussion is the provision of a simple "mice in a box" scenario simulation.

1 Introduction

Multi-Agent Based Simulation (MABS) is concerned with the harnessing of Multi-Agent System (MAS) technology to enable large scale simulations. The challenge is the mechanisms and representations required to build frameworks to support the desired simulation. Using MABS the *characters* that play a part in the simulation, and the environment(s) in which they exist, are conceptualised as agents. MABS has been applied in many domains such as: the monitoring and control of intelligent

E. Agiriga, F. Coenen, D. Kowalski
Dept. of Computer Science, University of Liverpool, Liverpool L69 3BX, UK. e-mail: {grigs,coenen,darek}@liverpool.ac.uk

R. Beynon
Institute for Biocomplexity, University of Liverpool, Liverpool L69 3BX, UK. e-mail: R.Beynon@liverpool.ac.uk

J. Hurst
Mammalian Behaviour and Evolution Group, University of Liverpool, Liverpool L69 3BX, UK. e-mail: G.E.Hurst@liverpool.ac.uk

buildings [2], transport chains [3], malaria re-emergence in the south of France [6], and urban population growth [7], to give just a few examples. To the best knowledge of the authors there is no work on MABS frameworks to study rodent behaviour. This paper describes some early research regarding issues concerned with the provision of MABS frameworks for rodent control. The focus of this report is a simple "mice in a box" scenario. However, the intention is to develop the framework so that it can be used to support large scale mouse simulations comprising some thousand agents.

2 The Mouse in a Box Scenario

The scenario at which the discussion presented in this paper is directed is that of a number of mice contained in a box. The scenario is founded on the sort of experiments conducted by rodent behaviourists who wish to observe the way that mice interact when placed in a closed environment, namely a $1.22 \times 1.22m$ box[1]. The fundamental idea is that one, two or more mice are placed in a box in which they can "run around". Mice have an affinity to walls [1] (they are *thigmotaxic*) and thus tend to moves along walls (although not exclusively so), thus in the absence of any obstructions a mouse's movements tend to be limited to the edges of the box. The mouse can move round the box in either a clockwise or ant-clockwise direction. It can also stop or turn around, occasionally it may venture into the space in the middle of the box. Mice are also interested in exploring their surroundings, the ultimate goal is the find and maintain an optimum nest location. The stronger Male mice have the best territory (nest locations). Females look for males with the best territory. Males mark their territory with scent, the stronger the male the stronger the scent. In the scenarios considered in this paper only male mice are considered. They are driven by the following desires:

1. A preference for wall locations as opposed to open space locations (in open space they are liable to attack by predators).
2. A desire to explore their environment.
3. A desire to avoid locations which feature the scent of other mice (unless that scent is significantly weaker than the mouse's own scent).
4. A requirement to avoid other mice that come into close proximity.

The above provides for some motivation for a mouse agent to move (to explore its locality), although there is no specific goal (reward). Whether the mouse moves or does not move, how long it moves for (or does not move) and which direction it should take, is a decision influenced partly by the above desires and partly by a random element.

[1] The value 1.22 is a result of the fact that the board from which the boxes are typically fashioned comes in $2.44 \times 1.22m$ sheets

3 The MABS Framework

The MABS framework is conceptualised in terms of a “cloud” in which a number of agents exist (Figure 1). From the figure we have three types (classes) of agent: (i) environment agents, (ii) obstruction agents and (iii) mouse (character) agents. The first two are characterised as “dumb” agents in that they do not display any intelligence, while the last has some “thinking” capability. From the figure it can be observed that we have only one environment agent and any number of obstruction and mouse agents (in fact we can have zero obstruction agents, but it would not make any sense to have zero mouse agents). In Figure 1 the arcs indicate communication lines; so the vision is that mouse agents can communicate with one another and the environment agent, while obstruction agents only communicate with the environment. Inspection of Figure 1 indicates that we also have some: (i) house keeping agents to facilitate the operation of the framework, (ii) a simulation interface with which an end user can interact so as to set up individual simulations and (iii) a visualisation unit that allows the end user to observe simulations. Each of the individual classes of agent are described in detail in the following three sections.

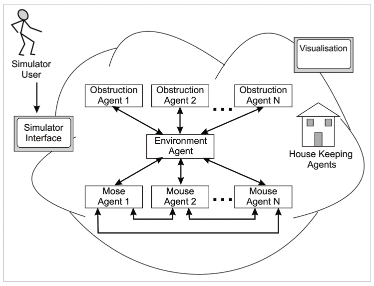


Fig. 1 Proposed MABS Framework

4 The Environment Agent

In the context of the proposed MABS framework an environment agent describes the *playing area*. In the case of the mouse in a box scenario this will be the box. A significant research issue with respect to the desired MABS is how best to represent this playing area. The simplest approach is to represent the playing area as a 2-D grid. However, this may not scale up for large simulations and features the irritation that the centroids of the neighbouring squares of a current square are not equidistant (neighbouring squares on the diagonal are further away than the immediately adjacent squares). Alternative representations include hexagonal grids, vector maps and graphs. However, because of its simplicity, the 2-D grid representation was adopted with respect to the framework described here.

The environment agent thus represents a playing area comprised of a 2-D grid. The dimensions of the environment were defined in terms 1cm units. A mouse was assumed to measures 7cm in all directions (not true, but the assumption can be upheld for the purpose of the simple mouse in a box scenario). A mouse was deemed to move at the rate of one 1cm per 50 mili-seconds. Each grid square (location) was given a numeric code, a Ground Type Identifier (GTI), indicating the nature of the square. The currently available codes were in the range $|0\ldots4|$ where: 0 indicated a "no-go" square, 1 a "wall" square, 2 a "space" square (non-wall square), 3 a "choice point" and 4 an obstruction (serving to hide the location of other mouse agents). The mouse cannot move into no-go or obstruction locations.

A mouse agent's location is described by its centroid; thus a mouse cannot get closer to a wall or obstruction than 3cm. Therefore all squares within three units of a wall or obstruction were encoded as no-go squares (0), squares exactly four units away from a wall or obstruction were labelled as wall squares (1), and squares more than four units away from walls as space squares (2). Choice points, at their simplest, are then wall squares that coincide with *obtuse* corners; where the mouse might wish to change direction (or stop); or squares where current movement may proceed in more than one wall direction. The corners of the boxes could also have been marked as choice points; however the movement of a mouse agent entering into these locations will be blocked thus, in effect, the location acts as a choice point without actually being marked as such

The current implementation features six types of environment agent: (i) Box, (ii) H-box, (iii) O-box, (iv) Four Box, (v) Four Nest and (vi) Maze. The first represents the simplest scenario. The H-box introduces the concept of obstructions (agents in their own right) into the box scenario, obstructions can be thought of as "bricks" placed into the box environment so as to impede a mouse agent's progress. The four box scenario comprises four occurrences of the box scenario running simultaneously, but described as a single environment with obstructions placed so as to achieve four boxes. The four nest box was used to simulate the interaction of four mouse agents. The maze scenario comprises a box scenario with a set of obstructions arranged to form a "maze", the objective here was to test whether a mouse object could find its way through this environment. Every environment agent has the following fields:

1. *widthX*, the width of the environment, in terms of grid squares, in the X (East-West) direction.
2. *widthY*, the width of the environment, in terms of grid squares, in the Y (North-South) direction.
3. *groundArea*, the two dimensional grid describing the *locations* that make up the ground area (as described above).
4. *gateCoords*, one or more gates where characters can enter the environment (start points).
5. *obstructionList*, a list of zero, one or more obstruction agents that the environment needs to know about.

Each location within the environment has a GTI and a record of any scent at the location, together with the ID for the mouse agent to which the scent sample belongs. Scent is defined in terms of an integer. Scent typically lasts for 8 to 24 hours depending on the dominance of the mouse. We degrade the mouse scent on each iteration of the simulation. To speed up the simulation we can enhance the degradation factor. Currently the maximum scent strength is 255 and it is degraded by 0.25 on each iteration (a more realistic simulation would require a much lower degradation factor).

5 The Obstruction Agent

Obstruction agents are simple agents that, as noted above, can be conceptualised as "bricks" that may be located within an environment. The bricks may be placed in the box as the scenario progresses, hence obstructions are considered to be agents in their own right. The H-box environment contains two obstruction agents so that the environment, when observed in plan view, formed an "H" shape. The O-box contained a single obstruction in the middle of the box so that the environment, when observed in plan view, resembled an "O" shape. The four box and four nest environment also contained two obstruction agents, but arranged to form an intersecting cross so as to divide the environment into four sub-boxes (Our "bricks" can intersect) and to form four "nest area" respectively. The maze environment had eighteen obstruction agents arranged in a "maze" formation. Similar to an environment agent, obstruction agents are dumb agents. The significance of obstruction agents is that mouse agents cannot "see" behind them; they obstruct a mouse agent's "field of view".

6 The Mouse Agent

A mouse agent is the central character in our mouse simulator. Mouse agents have the following fields:

1. $state$, the current state of the mouse agent, either $moving$, $stopped$ or $turning$.
2. $stateTime$, the time spent in the current state.
3. $coordX$, the mouse agent's current X location with respect to the environment agent.
4. $coordY$, the mouse agent's current Y location with respect to the environment agent.
5. $direction$, the direction the mouse agent is facing, a number in the range of $|0\ldots7|$ representing N, NE, E, SE, S, SW, W or NW respectively.
6. $goalDirection$, the direction the agent wishes to face (only applicable when turning).
7. $turnDirection$, the "turning direction", either $clockwise$ or $anticlockwise$ (also only applicable when turning).
8. $scentStrength$, the strength of its scent.

9. *visionMap*, a disc of locations, with radius v, representing the part of the environment which a mouse agent can "see". Thus a mouse agent's field of vision is equivalent to v.

Table 1 Action Table

Current State	Event	Action	Comments	New State
stopped	None	Agent decides to move in direction faced	$sateTime = 0$	*moving*
stopped	None	Agent decides to move another direction	$sateTime = 0$	*turning*
moving	At choice point	Agent decides to move in new direction		*turning*
moving	Obstruction	Agent decides to move in new direction		*turning*
moving	Obstruction	Agent decides to stop	$sateTime = 0$	*stopped*
moving	None	Agent decides to stop	$sateTime = 0$	*stopped*
turning	Completed turn	None		*moving*

Mouse agents are dynamic agents in that they can change their location, direction, goal direction, turn direction and state. At the same time they are "intelligent" agents in that they can make decisions about which way to face and where to go.The operation of our mouse agent is founded on the well established concept of a Finite State Machine (FSM) [5, 8]. FSM are used to model processes in terms of a finite set of *states*. A change from one state to another is called a *transition*. Transitions are caused by *events* or *actions* (something happening to the agent or the agent doing something). The possible transitions to a new state, caused by an event or action, are typically described using a *transition table* (*state diagram* or *state table*). FSMs can be conceptualised as graphs (state models) where the vertexes represent states and the edges transitions caused by events or actions. An alternative approach would be to use the Belief-Desire-Intention (BDI) model [4]. This offers the advantage that it is supported by existing logic models. However, planning is typically outside the scope of the model. Given that in our model we think of mouse agents being in a certain state; and that changes from one state to another with an element of randomness as well as intention (expressed in the form of preferences), a finite state machine mechanism of operation was adopted.

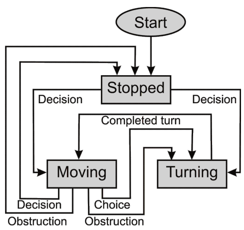


Fig. 2 State Model

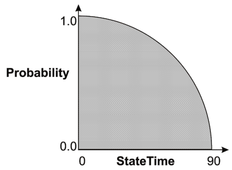


Fig. 3 Cosine Probability

The transition table for the mouse object is given in Table 1, which should be interpreted with respect to the state model presented in Figure 2 . There are seven different possible transitions. At the start of each simulation the default state for a mouse object is *stopped*. Eventually the mouse will decide to move (how this is determined is discussed below). The mouse object can either move in the direction it is currently facing or turn to face another direction and then move (how this is determined is also discussed below).Thus there are two possible state transitions associated with the *stopped* state.

There are four possible state transitions associated with the *moving* state. The first is when the mouse reaches a choice point. From the above, mice are "wall huggers". A choice point is a location where there are more than one possible next wall locations (as in the case of the maze environment) or the next possible wall location requires a change in direction. In the first case the mouse may decide to continue to move in the current direction, in which case there will be no state change; alternatively the mouse may decide to turn and move in a new direction, thus adopt a *turning* state. The second and third movement state changes are where the mouse's movements are blocked (for example at the corner of a box environment). In this case the mouse can decide to stop (adopt a *stopped* state) or head off in a new direction (change to a *turning* state). Note that in the case of a choice point, in the current implementation, the mouse does not have an option to stop. Finally a mouse in a *moving* state may simply decide to stop (how this is determined is discussed below).

The final state transition in Table 1 occurs when a mouse agent completes a turn, in which case the mouse will move in the direction it is now facing (i.e. adopt a *moving* state). The assumption here is that the only reason for a mouse to turn is to move in a new direction.

At the start of a simulation the state of the mouse agent is always *stopped*. Conceptually the mouse agent can only stay stopped for a finite period of time T. The probability that the mouse will stay stopped decreases as the the current $stateTime$ increases (i.e. as the time the mouse agent has spent in its stopped state increases). When $stateTime \equiv T$ the probability that the mouse will stay stopped is 0.0 (definitely decide to move), when $stateTime \equiv 0$ the probability is 1.0 (definitely stay stopped). This probability distribution was modelled using a cosine probability curve (Figure 3); we could have used a linear probability, or some other alternative, however the cosine probability has the feature that the likelihood of the mouse agent staying stopped remains high at low $stateTime$ values, and becomes negligible (reducing to 0.0) as $stateTime$ approaches T. On each simulation iteration, when the mouse object is stopped, a random number r is generated. A state transition will then occur when:

$$r < cosin\left(\frac{90 \times stateTime}{T}\right) \tag{1}$$

A similar process was applied where a mouse agent's state is *moving*. The assumption is again that the mouse will continue to move for a finite period of time, but in this case the time period was assumed to be $2T$. Thus, on each iteration, when the mouse object is moving a state transition will occur when:

$$r < cosin\left(\frac{90 \times stateTime/2}{T}\right) \tag{2}$$

7 Selecting a Direction of Travel

When a mouse agent reaches a choice point or discovers an obstruction (i.e. it cannot or may not proceed any further in the current direction) the agent must make a decision. Where an obstruction is reached the mouse has the option to stop or proceed in a new direction (see Figure 2); the decision whether to stop or not is determined using identity 2. Where a change of direction is indicated a mouse agent has between 0 and 8 potential directions it can choose from. A mouse agent cannot enter no-go locations ($GTL = 0$); thus, depending on the mouse agent's current location, some directions will not be permissible. It is possible for a mouse agents movement to be entirely blocked by obstructions and/or the presence of other mouse agents. in which case the mouse will adopt a *stopped* state. Assuming a mouse agent has one or more potential directions it can move in each potential direction has a preference value p of between $|0.0\ldots1.0|$. The complete set of preference values, P, is then defined as:

$$P = \{p_0, p_1, \ldots, p_n\} \tag{3}$$

such that:

$$\sum_{i=0}^{i=n} p_i = 1.0 \tag{4}$$

(where n is the number of available directions/locations).

Preference values are made up of a number of components $C = \{c_1, c_2, \ldots, c_m\}$, where m is the number of components. Each component describes some factor of the decision making process. A specific component j associated with a specific direction i is indicated as c_{ij}. Each component has a value of $|0.0\ldots1.0|$. Such that $\sum_{i=0}^{i=n} c_{ij} = 1.0$ (i.e. the set of values describing a particular component across the set of potential directions is equivalent to 1.0). Some components may be considered to have greater significance than others, thus the components are weighted[2]. The weighting associated with a component c_j is indicated by w_j. The preference (p) for a particular location (i) is then calculated as follows:

$$p_i = \frac{\sum_{j=0}^{j=m} w_j p_{ij}}{\sum_{j=0}^{j=m} w_j} \tag{5}$$

In the current simulation implementation four components are considered ($m = 4$) as follows:

[2] Although not a feature of the current implementation, these weighting mat be dynamic (i.e. they may be changed according to circumstances).

c_1 Preference according to GTL (desire for wall locations over space locations).
c_2 Preference for locations not recently or never visited (desire to explore).
c_3 Preference for avoiding locations where the scent of another mouse is significant compared with a mouse agent's own scent strength (desire to avoid the scent trails of other mice).
c_4 Preference for directions that tend to move way from other mouse agents if within sight (desire to avoid other mice).

Algorithm 1 Determination of preference for wall location component (c_1)

```
L = Set of potential locations
N_n = Number of nonspace locations in L
N_s = Number of space locations in L
if N_s ≡ 0 then
    p_n = 1.0/N_n
else
    p_n = P_n/N_n
    p_s = P_s/N_s
end if
for i = 0 → |L| do
    if L_i.groundType ≡ space location then
        L_i.c_1 = p_s
    else
        L_i.c_1 = n_s
    end if
end for
```

7.1 Desire for Wall Locations over Space Locations (c_1)

As noted above mice prefer to move along walls, thus a preference should be given to directions (next locations) adjacent to walls. A mouse agent will have potentially N_n wall locations and N_s space locations to choose from, where N_n and N_s are whole numbers in the range of $|0\ldots 8|$. Except in the special case where a mouse agent is blocked in, $1 \leq (N_n + N_s) \leq 8$. Of these directions zero, one or more will be space locations, and one or more will be non-space (wall or choice point) locations. The overall probability that a non-space location, L_n, is selected is given by P_n; and the overall probability that a space location, L_s, is selected by P_s, where P_n is assumed to be significantly greater than P_s. If $N_s \equiv 0$ then $P_n = 1.0$. Thus the probability of selecting a specific non space location is given by P_n/N_n, and the probability of selecting a specific space location (if such locations exist) is given by P_s/N_s. The process of determining the values for the preference component that reflects a desire for wall locations is given in algorithm 1.

7.2 Desire to Explore (c_2)

The desire to explore is expressed according to where a mouse agent has been recently, which in turn is expressed according to the scent strength of the mouse agent's own scent strength found at neighbouring locations. A mouse agent prefers locations (directions) where its own scent is not present, or at least weak. Thus the preference for new locations is expressed as a fraction of the inverse of the mouse agent's own scent strength (s_{inv_i}) at a given location i. If no scent is present $s_{inv} = 1.0$. The process for calculating the desire to explore preference component is given in algorithm 2. The c_2 component at a particular candidate location q is given by:

$$c_{q2} = \frac{s_{inv_q}}{\sum_{i=0}^{i=n} s_{inv_i}} \tag{6}$$

In algorithm 2 the factor k is used to reduce the influence of the scent strength at recently visited locations. The current maximum scent strength is 255, and thus the k value has been set to 10; if we simply used the inverse of the scent strength the influence of very recent directions will be negligible, 0.004 ($1/255$) as compared to 0.039 ($10/255$).

Algorithm 2 Determination of desire to explore component (c_2)

```
L = Set of potential locations
S = Set of inverses scent strengths
total = 0.0
for i = 0 → |L| do
    if L_i.ownScentStrength ≡ 0 then
        S_i = 1
    else
        S_i = k/L_i.ownScentStrength
        total = total + S_i
    end if
end for
for i = 0 → |L| do
    L_i.c_2 = S_i/total
end for
```

7.3 Desire to Avoid Scent Trails of other Mice (c_3)

The desire to avoid the scent trails of other mice is encapsulated in a similar manner to the desire to explore new locations. We use the inverse of the strength of the strongest scent belonging to another mouse agent, or 1.0 if there is no such scent. The process is presented in algorithm 3 where *maxScentStrength* is the scent strength associated with the scent strengths at a location belonging to other mice, 0 if there is no such scent strength.The constant K is again used.

7.4 Desire to Avoid other Mice (c_4)

A mouse agent knows nothing about the locations of other mice until they appear on its vision map. In the current simulation the radius of the vision map (v) is set to 20, however if the location of another mouse agent is obscured by an obstruction the current mouse agent will not know anything about this other mouse. To ensure the mouse agents do not actually crash into each other a buffer region of ten units is place round other mouse agents. Our mouse agents are currently programmed to avoid other mouse agents that are on its vision map. The values for this preference component are calculated according to the distance d from each candidate location to the nearest other mouse (if any). The c_4 component at a particular candidate location q is the distance d from the given candidate location q divided by the sum of the distances from all of the locations. Thus:

$$c_{q4} = \frac{d}{\sum_{i=0}^{i=n} d} \tag{7}$$

Algorithm 3 Determination of desire to avoid scent trails of other mice (c_3)

$L = Set\ of\ potential\ locations$
$S = Set\ of\ inverses\ scent\ strengths$
$total = 0.0$
for $i = 0 \rightarrow |L|$ **do**
  if $L_i.maxScentStrength \equiv 0$ **then**
    $S_i = 1$
  else
    $S_i = k/L_i.maxScentStrength$
    $total = total + S_i$
  end if
end for
for $i = 0 \rightarrow |L|$ **do**
  $L_i.c_3 = S_i/total$
end for

Algorithm 4 Next Location Algorithm

$L = Set\ of\ potential\ locations$
$Prob = 0.0$
$R = randomNumberGenerator()$
$L_{final} = -1$
for $i = 0 \rightarrow |L|$ **do**
  $Prob = Prob + L_i.prob$
  if $R < Prob$ **then**
    $L_{final} = L_i$
    $break$
  end if
end for
$return(L_{final})$

7.5 Decision making process

From the above each location has four components which are used to calculate a preference value for the location. Experiments indicated that the weighting that should be associated with c_1 and c_3 should be higher than those associated with the other components, w_1 and w_3 were therefore set to 2, while the remaining weightings were set to 1. The total preference for a particular location q was this given b:

$$p_q = \frac{2c_1 + c_2 + 2c_3 + c_4}{5} \tag{8}$$

The selection of a new direction was then determined using algorithm 4. The weightings can of course be adjusted as desired by the end user.

7.6 Change of Direction

Having selected a new location it may be necessary to change direction, if so a state transition from moving to turning will occur. Where a turn is initiated the mouse agents *goalDirection* and *turnDirection* fields must be reset. The value for the *turnDirection* field is calculated as follows as shown in algorithm 2 (recall that directions are specified as integers within the range $|0\ldots7|$).

Algorithm 5 Direction of Turn Algorithm

```
diff = absolute(direction − goalDirection)
if if(goalDirection > direction) then
   if diff ≤ 4 then
      return("clockwise")
   else
      return("anticlockwise")
   end if
else
   if diff ≤ 4 then
      return("anticlockwise")
   else
      return("clockwise")
   end if
end if
```

8 Operation

The operation of the simulator was controlled by a Loop which iterated every 50 milliseconds. Thus, given that the mouse agent (when in a moving state) moves at a rate of one grid square per iteration and a grid square measures 1cm, the mouse agent travels at 1200cm per minute (or 72km per hour). Experiments were conducted using a number of different environments with a turn rate of 45 degrees per iteration, $T = 90$, $P_n = 0.95$, $P_s = 0.05$ and $k = 10$. The Box experiment was intended to establish that the mouse agent behaved in a reasonably realistic manner, as confirmed by domain experts. The H-box was intended to establish that the mouse agent could react to obstructions, the O-box was intended to observe the mouse agent's behaviour should it cross the open space between the outer wall of the box and the obstruction, the Maze experiment was used to evaluate the mouse agent's ability to negotiate choice points and the 4-box to demonstrate that mouse agents did not behave in the same way given four identical spaces. Finally the four nest box simulation was used observe how a group of mouse agents might interact given a hypothetical situation that they each might want to guard their own nest site.

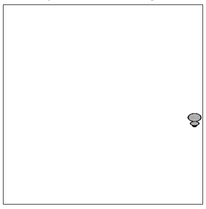


Fig. 4 Box Simulation

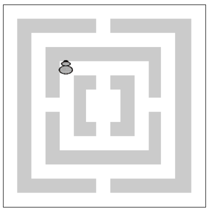


Fig. 5 Maze Simulation

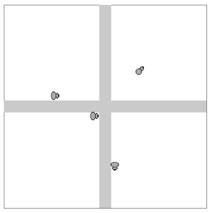


Fig. 6 4 Box Simulation

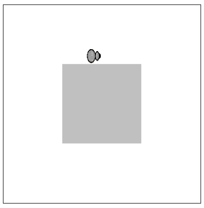


Fig. 7 O Box Simulation

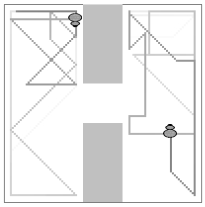


Fig. 8 H Box Simulation (with scent traces)

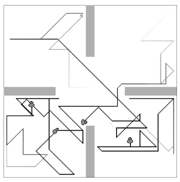


Fig. 9 Four Nest Box Simulation (with scent traces)

Figures 4 to 10 illustrating the simulations. Inspection of the figures indicates how mouse agents, when not influenced by the presence of other mouse agents, tend to follow walls. In the case of the O-box environment (Figure 7) the mouse agent has crossed the open space and is now hugging the wall of the obstruction. Figure 8 shows the H-box environment with two mouse agents and Figure 9 the four nest environment with four mouse agents. Both figures include scent trails. The objective in both cases was to observe how mice agents might define there own space. For the benefit of the simulation, and to allow easy observation, the "lifespan" of the scent deposits was kept deliberately shirt. Better results would be achieved by increasing the longevity of the scent trails however in this case the simulation has to be run over a much longer (and more realistic) time period. The experiment demonstrated in Figure 10 was designed to demonstrate that the simulator could function with a reasonable number of mouse agents.

9 Discussion and Conclusions

In this paper we have described a simple Multi-Agent Based Simulation (MABS) framework to describe the mouse in a box scenario. The intention was to provide a simple start point for the development of large scale rodent simulations. Features of the framework are: (i) that it can be used to create sophisticated environments using the concept of obstruction agents, (ii) several mice can operate in these environments and (iii) the mice operate in a sufficiently realistic manner. Experiments indicated that environments were easy to create and that simulations were easy to run and observe. The authors therefore believe that they have established a sound foundation on which to build. Current work is directed at techniques to support more sophisticated scenarios and to allow mouse agents to learn about their environments.

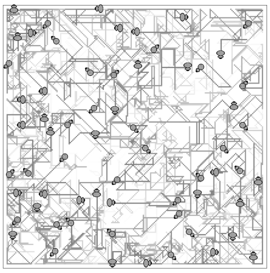


Fig. 10 Large (64 mouse agent) box simulation (with scent trails)

References

1. P. Crowcroft. *Mice All Over*. The Chicago Zoological Society, 1973.
2. P. Davidsson. Multi agent based simulation: Beyond social simulation. In *Proc. Workshop on Multi Agent Based Simulation (MABS) 2000*, pages 141–155. Springer-Verlag, LNCS 1979, 2001.
3. P. Davidsson, J. Holmgren, J.A. Persson, and L. Ramstedt. Multi agent based simulation of transport chains. In *Proc. 7th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS), Volume 2*, pages 1153–1160. Springer-Verlag, LNAI 3415, 2008.
4. B. Georgeff, M.P. andPell, Pollack M.E., M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. In *Proc. 5th Int. Workshop on Intelligent Agents: Agent Theories, Architectures and Languages (ATAL'98)*, pages 1–10. Springer-Verlag, London, 1998.
5. A. Gill. *Introduction to the Theory of Finite-state Machines*. McGraw-Hill, 1962.
6. C. Linarda, N. Pononb, D. Fontenilleb, and E.F. Lambin. A multi-agent simulation to assess the risk of malaria re-emergence in southern france. *Ecology Modelling*, 220(2):160–174, 2009.
7. X. Pan, C.S. Han, K. Dauber, and K.H. Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI and Society*, 22(2):113–132, 2007.
8. F. Wagner. *Modeling Software with Finite State Machines: A Practical Approach*. Auerbach Publicationsl, 2006.

Appendix B

A Multiagent Based Framework for the Simulation of Mammalian Behaviour

A Multiagent Based Framework for the Simulation of Mammalian Behaviour

E. Agiriga, F. Coenen, J. Hurst and D. Kowalski

Abstract A Mammalian Behaviour Multi-Agent Based Simulation (MBMABS) framework is proposed. Central to the framework is the concept of a behaviour lattice comprised of vertices representing states and edges representing possible state changes. State changes occur as a result of an agent completing some self-appointed task or as a result of some external event. Each state has one one or more predefined potential follow on states. Where there is more than one follow on state selection is made according to a weighted random selection process. The weightings are derived dynamically according to individual agent desires. The elements of the MBMABS framework are described in detail. The operation of the framework is illustrated using a case study.

1 Introduction

Computer simulations are used widely with respect to all kinds of applications [2, 5, 6]. A growing area of interest for computer simulation is animal behaviour. Animal behaviour can be perceived of as the way in which animals react to an environment as typically exhibited through movement [3, 4]. Simulations of animal behaviour are seen as desirable for a variety of reasons, the most significant of which are: (i) once established they are inexpensive to operate, (ii) they can be used for what if style experiments without causing any permanent damage, (iii) they provide a simple mechanism for experiments to be repeated using the same set of

Emmanuel Agiriga, Frans Coenen, Darek Kowalski
Department of Computer Science, University of Liverpool, Liverpool, L69 3BX e-mail: grigs, coenen, darek@liverpool.ac.uk

Jane Hurst
Mammalian Behaviour and Evolution Group, University of Liverpool, Liverpool, L69 3BX e-mail: g.e.hurst@liverpool.ac.uk

parameters or by changing only one parameter, and (iv) they provide an excellent tool to enhance understanding of animal behaviour. The primary purpose of animal behaviour simulation is to allow behaviourologists to extend their current knowledge without needing to resort to expensive real life experimentationThe work described in this paper proposes the Mammalian Behaviour Multi-Agent Based Simulation (MBMABS) framework, a framework to support computer simulation of mammalian behaviour. The framework is founded on ideas first proposed by the authors in [1]. The basic idea is that each animal is represented by an agent. The behaviour for each agent is encapsulated in terms of a set of desires (D) and a behaviour lattice (B). These are described in Sections 2 and 3 respectively. Each agent has five main attributes: (i) a location within some environment (described by an x-y coordinate pair, (ii) a direction in which it is facing, (iii) a velocity (which may be zero indicating that it is not moving), (iv) a *state* defined by a vertex in a behaviour lattice, and (v) a set of desires. A third element of the MBMABS framework is the concept of environments (landscape) in which the agents are intended to operate. The MBMABS framework has been designed to provide a generic simulation facility that allows the inclusion of a range of desires and behaviours. The simulation operates on a iterative basis. On each iteration agents either perform some action according to their current "state" or undertake a state change. To provide for a full understanding of the proposed environment a case study is presented in Section 4. Some conclusions are presented in Section 5.

2 Desires

An agent can have any number of desires (k), goals or objectives that the agent wishes to adhere to, $D = \{d_1, d_2, \ldots, d_k\}$. Each desire has a "strength" associated with it, a number between 0.0 and 1.0. Desires are characterized as being either: (i) constant or (ii) dynamic. A constant desire is one whose strength remains fixed throughout a simulation, while a dynamic desire is one whose strength changes (i.e. increases,reduces or remains static) with time. We model the changing strengths associated with dynamic desires using a cosine curve. A change in the character of a dynamic desire is usually associated with a state change. Typically an agent has several competing desires at a given time point in a simulation. A simple application of desires is in the selection of a direction for an agent that has decided to adopt a moving state. When an agent has decided to move it will have n directions to move in where n is defined by the number of immediate neighbour tiles into which the agent can move although some of these may feature obstructions. Thus we can identify a set of T possible tiles $T = \{t_1, t_2, \ldots, t_n\}$ where $0 \leq n \leq 8$. Note that the set T can be empty (the agent is unable to move). Each tile in T will also have a weighting associated with it calculated as the simulation progresses. Thus we have a set weightings $W = \{w_1, w_2, \ldots, w_n\}$ associated with each possible location in T indicating there desirability with respect to D.

3 Behaviour Lattice

A central feature of the MBMABS framework is the behaviour lattice. The behaviour lattice comprises: (i) a set of vertices each describing a "state" and (ii) a set of directed edges describing permitted state changes. Only certain states follow on from other states (have edges between them). Throughout a simulation each agent in the simulation is associated with one and only one vertex in the behaviour lattice at any discrete time point. State changes occur as a consequence of some event. With respect to some states there may be a number of alternative independent events that can trigger a state change. Events may be either: (i) external or (ii) internal. An external event is associated with some occurrence resulting as a consequence of the agent moving around its environment, for example encountering an obstruction or another agent. An internal event is concerned with an agent completing some self appointed task, for example changing the direction in which it is facing or timing out. Timing out is concerned with the duration whereby an agent may remain in some states; agents are assumed to be unable to remain in any one particular state indefinitely.

Although not applicable to all states, timing out is implemented using a value p (a field in each agents definition) that is set to 1.0 when the agent moves into a relevant state (vertex in the behaviour lattice). This value is then decreased according to the definition of a cosine curve, on each iteration of the simulation. With each iteration a random number r $(0.0 \leq r \leq 1.0)$ is generated. If r is greater than p a state change is triggered. Thus at time 0 the probability that an agent will remain in its current state is 1.0 (definitely remain), at time N the probability that an agent will remain in its current state is 0.0 (definitely not remain); thus, as time progresses, the likelihood of a state change increases. The value of N will depend on the nature of the state under consideration.

Each vertex in the behaviour lattice will have at least two methods associated with it: (i) an action method and (ii) a state change method. The action method is used to process the current action of the agent. Three standard action methods are: moving, stopped and turning. The state change method is used to identify a follow on state and undertake any preparatory processing required before the follow on state can be commenced. Follow on states are selected in either a fixed manner or a probabilistic manner. Fixed selection occurs where, as a result of some event, there is only one possible follow on state. Probabilistic state changes occur where there are a number of competing alternative follow on states, in which case one is chosen in a probability influenced random manner whereby a weighting mechanism is used to influence follow on state selections according to current desire strengths.

4 Case Study, A Mouse in a Box Simulation

This section describes the operation of the MBMABS framework by considering a case study directed at mouse behaviour. The authors have used the proposed MBMABS framework to implement a mouse behaviour simulator. More specifically the case study considers the situation where a mouse is placed in a new environment which it is then expected to explore. The exploring is directed by a desire to explore. The environment for the mouse behaviour simulation was a simple box. This was adopted because identical boxes are used with respect to laboratory based experiments using real mice; hence the operation of simulated scenarios could be compared with similar scenarios run in ”real-life”. The environment E in this case comprised a set of tiles labelled using the the set $\{0,1,2,3,4,5\}$ indicating (respectively): wall locations,corner locations, tunnel locations, choice locations (a location where we wish to enforce consideration for change) and open space. These all have significance with respect to mouse behaviour. The rest of this section is organised as follows; in Sections 4.1 and 4.2 we consider the mouse desires and the behaviour lattice for the case study. Then in 4.3 we discuss the operation of the simulation.

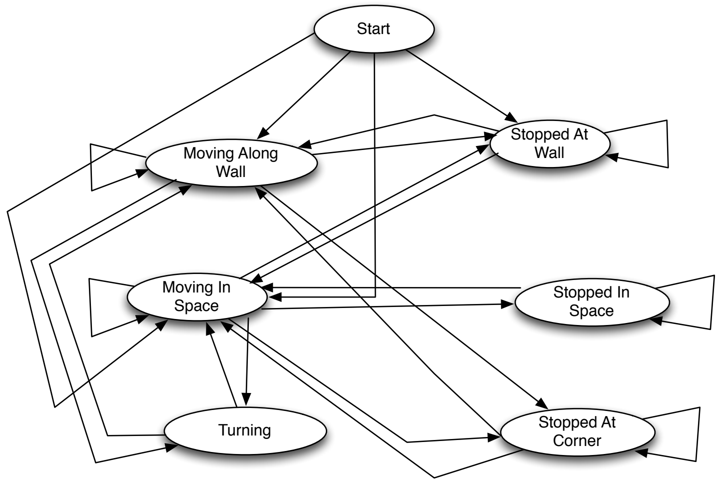


Fig. 1: Fragment of behaviour lattice for mouse case study

4.1 Mouse Agent Desires

For the purpose of the case study presented in this section it is assumed that our mice agents have only two desires: (i) a constant desire to stay close to walls and (ii) a dynamic desire to explore their environment. The preference for wall locations is a behaviour exhibited by mice called *thigmotaxis*. The desire to explore is a feature of many mammalian behaviours. In the case of mice they “know” the best (fastest and/or safest) route back to their nest site. With respect to our mice behaviour simulation the desire to explore is expressed as the desire to create a mental map of their environment (which can later be utilised). This map comprises a set of vertices and edges (and should not be confused with a behaviour lattice). The vertices are *waypoints* and the edges represent *travel lines*. Waypoints are significant locations and are currently defined as corners or choice points. The desire to explore is a dynamic desire, initially set to 1, that will decrease until a new waypoint is found. If no waypoint is found the desire to explore reaches 0 it will remain at zero for the remainder of the simulation or until such time as a new waypoint is found when the desire to explore will jump back to 1 before starting to decrease again.

4.2 Mouse Agent Behaviour Lattice

The behaviour lattice for the mouse behaviour case study is given in the figure 1. From the figure it can be observed that the behaviour lattice features 7 vertices (states), they include: (i) Start, (ii) Moving Along Wall, (iii) Stopped At Wall, (iv) Moving In Space, (v) Turning, (vi) Stopped In Space and (vii) Stopped At Corner. Each representing a particular activity which the mouse agent may be performing at a particular time T in the simulation. The meaning of each state can be derived from its nomenclature. For instance the Start state is the *current state* at the beginning of the simulation. Each of these States have one or more permissible *follow on states* (states which may be adopt whenever a relevant events occurs). The directed edges of the behaviour lattice (Figure 1) indicate a transition from a *current state* to a *follow on state* as indicated by the direction of the arrow. Some states have several possible *follow on state*.

The mouse agent assumes the Start state as the *current state* at the beginning of the simulation (the Start state cannot exist as a follow on state). At the start of the simulation, the mouse agent will immediately select one of the three permissible follow on states (the follow on state is selected using the process described above in Section 3). As noted above, at the start of the simulation the dominant desire is the dynamic desire for the agent to explore its surroundings. There is also a constant desire for wall locations, so the agent is more likely to choose the Moving Along Wall state.

4.3 Operation

The simulation thus mimics the conjectured process whereby a mouse agent might build up a mental map of its environment describing *interest locations* (corners locations in this case study). At the same time the mouse agent creates links (paths) between interest locations. The simulation operates as follows. The agent starts at a predetermined gate location next to a wall. It then has to make decision whether to start exploring or simply move around the environment in a random manner. The desire to explore will be strong so the likelihood is that the mouse agent will adopt a moving state. Since it also has a desire for proximity to walls it is most likely to adopt a moving along wall state. As it proceeds the desire to explore will start to decrease. The mouse will continue in this moving state until either it finds an interest location or the state “times out”. In the first case the location will be mapped (with a node in the mental map), the desire to explore will jump back up to 1.0 and the mouse agent will continue. In the second case the mouse agent may decide to resume moving along the wall or move back along the wall or move away from the wall or assume a stopped state. The mouse agent will continue moving around its environment in this manner. At some point its desire to explore will drop to zero, this will happen when after some time t no further, previously undiscovered, interest locations are found. While the desire to explore is strong the mouse agent will try to make decisions about where to go next (which states to adopt) influenced by the current state of its map.

5 Discussion and Conclusions

The MBMABS Multiagent Based Simulation framework for modelling animal behaviour has been presented. The central features of the framework are a set of desires and a behaviour lattice. The operation of the framework was illustrated using a mouse behaviour case study. Creation of the case study, and others not reported here, has demonstrated that the proposed framework also indicated that the framework readily supports the creation of such simulations. It was also easy to observe the behaviour of the simulated entities. The evaluation of the simulations was conducted with the help of animal behaviourists by comparing simulation behaviour with real behaviour. The evaluation also indicated that the MBMABS framework readily supports the addition of states and desires. It is however the case that as the number of states increase, the behaviour lattice becomes more complex and difficult to understand because the number of vertices and directed edges will also increase. For future work the research team intend to investigate more challenging animal behaviour scenarios such as nest site selection, territory guarding and threat avoidance. Experiments have been conducted using 4 mouse agents, however more testing environments with up to 64 mouse agents is being considered for future work.

References

1. E. Agiriga, F. Coenen, J. Hurst, R. Beynon, and D. Kowalski. *Towards Large-Scale Multi-Agent Based Rodent Simulation: The "Mice In A Box" Scenario*, pages 369–382. Research and Development in Intelligent Systems XXVIII. Springer, 2011.
2. MP Anderson, DJ Srolovitz, GS Grest, and PS Sahni. Computer simulation of grain growthi. kinetics. *Acta metallurgica*, 32(5):783–791, 1984.
3. Mohan P. Arora and Chander Kanta. *Animal behaviour [electronic book] Mohan P. Arora ; edited by Chander Kanta*. Mumbai India] : Himalaya Pub. House, 2009; Rev. ed, 2009.
4. Reena Mathur. *Animal behaviour [electronic book] Reena Mathur*. Meerut, India : Rastogi Publications, 2009, 2009.
5. Allen Newell and Herbert Alexander Simon. *Computer simulation of human thinking*. Rand Corporation, 1961.
6. Y. Zhang, BE Hobbs, A. Ord, and HB Mhlhaus. Computer simulation of single-layer buckling. *Journal of Structural Geology*, 18(5):643–655, 1996.

Appendix C

Behaviour Matrix For Case Study Category 1

C.1

Table C.1: Behaviour Matrix For Case Study Category 1

Current State	Follow On State	Event	Selection
Start	Moving Along Wall	Internal: start	Probabilistic
	Moving In Space	Internal: start	Probabilistic
	Stopped At Wall	Internal: start	Probabilistic
Moving Along Wall	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Turning	Internal: (i) timing out (ii) looking for interesting location. External (i) obstruction encountered	Fixed
	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
Turning	Moving Along Wall	Internal: completed turn	Fixed
	Moving In Space	Internal: completed turn	Fixed

C.2

Table C.2: Behaviour Matrix For Case Study Category 1 - Continued

Moving In Space	Moving In Space	Internal: timing out (ii) looking for interesting location	Probabilistic
	Turning	Internal: (i) change in direction required (ii) seeking interesting location. External (i) obstruction encountered	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped In Space	Internal: (i) timing out	Probabilistic
	Stopped At Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
Stopped At Wall	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Moving Along Wall	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out (ii)looking for interesting location	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic

C.3

Table C.3: Behaviour Matrix For Case Study Category 1 - Continued

Stopped In Space	Stopped In Space	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out	Probabilistic
Stopped At Corner	Stopped At Corner	Internal: (i) timing out	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic
Creating Nest	Creating Nest	Internal (i) timing out	Probabilistic
	Stopped At Wall	Internal (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped At Corner	Internal:(i) timing out (ii) looking for interesting location	Probabilistic

Appendix D

Behaviour Matrix For Case Study Category 2

D.1

TABLE D.1: Behaviour Matrix For Case Study Category 2

Current State	Follow On State	Event	Selection
Start	Moving Along Wall	Internal: start	Probabilistic
	Moving In Space	Internal: start	Probabilistic
	Stopped At Wall	Internal: start	Probabilistic
Moving Along Wall	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Moving In Tunnel	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Turning	Internal: (i) timing out (ii) looking for interesting location. External: (i) obstruction encountered	Fixed
	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic

D.2

Table D.2: Behaviour Matrix For Case Study Category 2 - Continued 2

Moving In Tunnel	Moving In Tunnel	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out	Probabilistic
	Turning	Internal: (i) timing out (ii) looking for interesting location. External: (i) obstruction encountered	Fixed
Moving In Space	Moving In Space	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Turning	Internal: (i) change in direction required (ii) looking for interesting location. External: (i) obstruction encountered	Fixed
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped In Space	Internal: (i) timing out	Probabilistic
	Stopped At Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
Stopped At Wall	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Moving Along Wall	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic

D.3

Table D.3: Behaviour Matrix For Case Study Category 2 - Continued 3

Turning	Moving Along Wall	Internal: (i) completed turn	Fixed
	Moving In Space	Internal: (i) completed turn	Fixed
	Moving In Tunnel	Internal (i) completed turn	Fixed
Stopped In Space	Stopped In Space	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out	Probabilistic
Stopped In Tunnel	Stopped In Tunnel	Internal: (i) timing out	Probabilistic
	Moving In Tunnel	Internal: (i) timing out	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic

D.4

Table D.4: Behaviour Matrix For Case Study Category 2 - Continued 4

Stopped At Corner	Stopped At Corner	Internal: (i) timing out	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic
Creating Nest	Creating Nest	Internal: (i) timing out	Probabilistic
	Stopped At Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out (ii) looking for interesting location	Probabilistic

Appendix E

Behaviour Matrix For Case Study Category 3

E.1

Table E.1: Behaviour Matrix For Case Study Category 3

Current State	Follow On State	Event	Selection
Start	Moving Along Wall	Internal: Start	Probabilistic
	Moving In Space	Internal: Start	Probabilistic
	Stopped At Wall	Internal: Start	Probabilistic
Moving Along Wall	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location.	Probabilistic
	Moving In Tunnel	Internal: (i) timing out (ii) looking for interesting location.	Probabilistic
	Turning	Internal: (i) timing out (ii) looking for interesting location. External: (i) obstruction encountered	Fixed
	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped At Nest Site	Internal: (i) timing out	Probabilistic

E.2

Table E.2: Behaviour Matrix For Case Study Category 3 - Continued

Moving In Tunnel	Moving In Tunnel	Internal: (i) timing out (ii) looking for interesting location.	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) looking for interesting location.	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out	Probabilistic
	Turning	Internal: (i) timing out (ii) looking for interesting location	Fixed
Moving In Space	Moving In Space	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Turning	Internal: (i) timing out (ii) change in direction required (iii) looking for interesting location. External: (i) obstruction encountered	Fixed
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped In Space	Internal: (i) timing out	Probabilistic
	Stopped At Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic

E.3

Table E.3: Behaviour Matrix For Case Study Category 3 - Continued

Moving Along Travel Lines	Moving Along Travel Lines	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out. External: (i) seeking next waypoint on mental map or nest to escape danger	Probabilistic
	Moving Along Wall	Internal: (i) timing out. External: (i) seeking waypoint on mental map or nest to escape danger	Probabilistic
	Moving In Tunnel	Internal: (i) timing out. External: (i) going to nest through known paths to escape danger	Probabilistic
	Turning	Internal: (i) timing out. External: (i) obstruction encountered on way to nest	Fixed
	Stopped At Wall	Internal: (i) timing out. External: (i) looking for next waypoint (using safe travel route to escape danger)	Probabilistic
	Stopped At Corner	Internal: (i) timing out. External: (i) seeking waypoint (using safe travel route to escape danger)	Probabilistic
	Stopped At Nest Site	Internal: (i) timing out. External: (i) found nest in danger	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out. External: (i) seeking waypoint (using safe travel route to escape danger)	Probabilistic

E.4

Table E.4: Behaviour Matrix For Case Study Category 3 - Continued

Moving To Nearest Safe Location	Moving To Nearest Safe Location	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out. External: (i) seeking nearest waypoint in danger	Probabilistic
	Turning	Internal: (i) timing out. External: (i) obstruction encountered (ii) change in direction required to find nearest waypoint in danger	Fixed
	Stopped At Wall	Internal: (i) timing out. External: (i) seeking first waypoint in danger to identify safe route to nest in danger	Probabilistic
	Stopped At Corner	Internal: (i) timing out. External: (i) seeking first waypoint in danger to identify safe route to nest in danger	Probabilistic
	Stopped At Nest Site	Internal: (i) timing out. External: (i) found nest in danger	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out. External: (i) seeking first waypoint in danger to identify safe route to nest in danger	Probabilistic

E.5

Table E.5: Behaviour Matrix For Case Study Category 3 - Continued

Stopped At Nest Site	Stopped At Nest Site	Internal: (i) timing out	Probabilistic
	Turning	Internal: (i) timing out (ii) seeking interesting location	Fixed
	Moving In Space	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
Turning	Moving Along Wall	Internal: (i) completed turn	Fixed
	Moving In Space	Internal: (i) completed turn	Fixed
	Moving In Tunnel	Internal: (i) completed turn	Fixed
	Moving To Nearest Safe Location	Internal: (i) completed turn	Fixed
	Moving Along Travel Lines	Internal: (i) completed turn	Fixed
	Stopped At Nest Site	Internal: (i) completed turn	Fixed

E.6

Table E.6: Behaviour Matrix For Case Study Category 3 - Continued

Stopped In Space	Stopped In Space	Internal: (i) timing out	Probabilistic
	Moving In Space	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
Stopped In Tunnel	Stopped In Tunnel	Internal: (i) timing out	Probabilistic
	Moving In Tunnel	Internal: (i) timing out	Probabilistic
	Create Nest Site	Internal: (i) creating nest in identified location	Probabilistic
Stopped At Corner	Stopped At Corner	Internal: (i) timing out	Probabilistic
	Moving Along Wall	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
	Moving Along Travel Lines	External: (i) seeking waypoint (using safe travel route to escape danger)	Probabilistic
	Creating Nest	Internal: (i) creating nest in identified location	Probabilistic
	Avoid Nest Site	External: (i) avoiding radius around a foreign nest	Probabilistic
Create Nest Site	Creating Nest Site	Internal: (i) timing out	Fixed
	Stopped At Wall	Internal: (i) timing out (ii) looking for interesting location	Probabilistic
	Stopped At Corner	Internal: (i) timing out (ii) looking for interesting location	Probabilistic

E.7

Table E.7: Behaviour Matrix For Case Study Category 3 - Continued

Resting	Resting	Internal: (i) timing out. External (i) hiding at nest from danger	Probabilistic
	Stopped At Nest Site	Internal: (i) timing out.	Probabilistic
Guard Nest Site	Guard Nest Site	Internal: (i) timing out	Probabilistic
	Stopped At Nest Site	Internal: (i) timing out. External (i) defending nest from intruder	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out. External (i) defending nest from intruder	Probabilistic
	Stopped In Space	Internal: (i) timing out. External (i) defending nest from intruder	Probabilistic
	Stopped At Wall	Internal: (i) timing out. External (i) defending nest from intruder	Probabilistic
	Stopped At Corner	Internal: (i) timing out. External (i) defending nest from intruder	Probabilistic

E.8

Table E.8: Behaviour Matrix For Case Study Category 3 - Continued

Avoid Nest Site	Avoid Nest Site	Internal: (i) timing out	Probabilistic
	Stopped At Corner	Internal: (i) timing out External (i) seeking locations outside radius of foreign nest	Probabilistic
	Stopped At Wall	Internal: (i) timing out External (i) seeking locations outside radius of foreign nest	Probabilistic
	Stopped In Tunnel	Internal: (i) timing out External (i) seeking locations outside radius of foreign nest	Probabilistic
	Stopped In Space	Internal: (i) timing out External (i) seeking locations outside radius of foreign nest	Probabilistic

E.9

Table E.9: Behaviour Matrix For Case Study Category 3 - Continued

Stopped At Wall	Stopped At Wall	Internal: (i) timing out	Probabilistic
	Moving At Wall	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
	Moving In Space	Internal: (i) timing out (ii) seeking interesting location	Probabilistic
	Moving To Nearest Safe Location	External: (i) seeking first waypoint in danger to identify safe route to nest in danger	Probabilistic
	Moving Along Travel Lines	Internal: (i) timing out External: (i) seeking next waypoint on map to escape danger	Probabilistic
	Create Nest Site	Internal: (i) creating nest in identified location	Probabilistic
	Avoid Nest Site	External: (i) seeking locations outside radius, r around a foreign nest	Probabilistic

Appendix F

Supporting Letter From Domain Experts

Professor Jane Hurst BSc PhD
William Prescott Chair of Animal Science

Mammalian Behaviour & Evolution Group
Institute of Integrative Biology
University of Liverpool
Leahurst Campus
Neston, CH64 7TE, UK

T +44 151 794 6100
E jane.hurst@liv.ac.uk
W http://www.liv.ac.uk/mbe

To whom it may concern,

RE: Emmanuel Agiriga

In view of the above named PhD student's up-coming viva, I thought it may be helpful to provide some additional commentary on the relevance of the work undertaken by the student in the context of the study of mammalian behaviour and especially rodent behaviour. I am the William Prescott Professor of Animal Science and head the Mammalian Behaviour and Evolution research group in the University's Institute of Integrative Biology. I was Emanuel's third PhD supervisor with a particular remit to provide domain knowledge in support of Emanuel's programme of study (what I understand is referred to as a "domain expert" in computer science circles).

Simulation tools, of the kind proposed and realised in the thesis, have wide ranging benefits in the context of the prediction of behaviour of rodents. In particular, the ability to predict the behaviour of pest species such as the house mouse in complex environments has considerable potential benefit for the development of more effective rodent control strategies. As yet, there is little understanding of how pest rodents interact with physical infrastructure in complex habitats, although this is essential for the most effective design and deployment of traps and baiting points. Thus, simulation tools that allow much better prediction of behaviour will have significant benefits in terms of food security, human and livestock health, the economic impact of infrastructure damage, and the potential for reduced ecological and other damage to non-target species. At present no such simulation tools exist, despite the global importance of mouse and rat pests and the failure of current baiting and trapping strategies to provide effective control.

The "pilot work" reported on in the thesis demonstrate that computational agents can indeed be used to define and predict the behaviour of large numbers of rodents, in complex environments, to inform and shape rodent pest control strategies. The work presented in the thesis thus provides an **excellent foundation for further work** directed at the simulation of rodent behaviour in the context of pest control (and mammalian behaviour in the wider context).

Yours sincerely

Prof Jane Hurst

References

[1] Matt Tuttle. Finite state machines for game developers, 2011. URL `http://matttuttle.com/2011/02/finite-state-machines-for-game-developers`.

[2] Paul Davidsson, Johan Holmgren, Hans Kyhlbäck, Dawit Mengistu, and Marie Persson. *Applications of agent based simulation*, pages 15–27. Springer, 2007.

[3] Stefania Bandini, Sara Manzoni, and Giuseppe Vizzari. Agent based modeling and simulation: an informatics perspective. volume 12, page 4. 2009.

[4] Daniele Benedettelli. Finite state machines. *Creating Cool MINDSTORMS® NXT Robots*, pages 47–58, 2008.

[5] Stephen C Pratt, David J Sumpter, Eamonn B Mallon, and Nigel R Franks. An agent-based model of collective nest choice by the ant temnothorax albipennis. *Animal Behaviour*, 70(5):1023–1036, 2005.

[6] Budhitama Subagdja, Liz Sonenberg, and Iyad Rahwan. Intentional learning agent architecture. *Autonomous Agents and Multi-Agent Systems*, 18(3):417–470, 2009.

[7] Weiming Shen, Qi Hao, Hyun Joong Yoon, and Douglas H Norrie. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced engineering INFORMATICS*, 20(4):415–431, 2006.

[8] Derek W Bunn and Fernardo S Oliveira. Agent-based simulation-an application to the new electricity trading arrangements of england and wales. *IEEE transactions on Evolutionary Computation*, 5(5):493–503, 2001.

[9] Paul Davidsson, Lawrence Henesey, Linda Ramstedt, Johanna Törnquist, and Fredrik Wernstedt. An analysis of agent-based approaches to transport logistics. *Transportation Research part C: emerging technologies*, 13(4):255–271, 2005.

[10] Weiming Shen, Lihui Wang, and Qi Hao. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(4):563–577, 2006.

[11] Nick Malleson, Linda See, Andrew Evans, and Alison Heppenstall. Implementing comprehensive offender behaviour in a realistic agent-based model of burglary. *Simulation*, pages 50–71, 2010.

[12] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3): 7280–7287, 2002.

[13] Xiaoshan Pan, Charles S Han, Ken Dauber, and Kincho H Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2):113–132, 2007.

[14] Bertrand Dumont and David R C Hill. Multi-agent simulation of group foraging in sheep: effects of spatial memory, conspecific attraction and plot size. *Ecological Modelling*, 141(1):201–215, 2001.

[15] Catholijn M Jonker and Jan Treur. Agent-based simulation of animal behaviour. *Applied Intelligence*, 15(2):83–115, 2001.

[16] Thomas P Monath, Verne F Newhouse, Graham E Kemp, Henry W Setzer, and Anthony Cacciapuoti. Lassa virus isolation from mastomys natalensis rodents during an epidemic in sierra leone. *Science*, 185(4147):263–265, 1974.

[17] Grant R Singleton, Steven Belmain, Peter R Brown, Ken Aplin, and Nyo Me Htwe. Impacts of rodent outbreaks on food security in asia. *Wildlife Research*, 37 (5):355–359, 2010.

[18] Jane Hurst. Mammalian behaviour and evolution, University of Liverpool, Liverpool, 2016. URL `www.liverpool.ac.uk/mbe/page34/page34.html`.

[19] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Mathematical model of self-organizing hierarchies in animal societies. *Bulletin of mathematical biology*, 58(4):661–717, 1996.

[20] Peter M Kappeler and Nils Anthes. *Animal behaviour*. Springer, 2010.

[21] Peter Crowcroft. Mice all over. *The Chicago Zoological Society*, 1966.

[22] Pascalle Leonie Petra Van Loo, Linda Van Zutphen, and Vera Baumans. Male management: coping with aggression problems in male laboratory mice. *Laboratory animals*, 37(4):300–313, 2003.

[23] Gordon C Rausser and Stanley R Johnson. On the limitations of simulation in model evaluation and decision analysis. *Simulation and Games*, 6(2):115–150, 1975.

[24] Muhammad Tufail, Frans Coenen, Tintin Mu, and Saqib Jamshid Rind. Mining movement patterns from video data to inform multi-agent based simulation. In *International Workshop on Agents and Data Mining Interaction*, pages 38–51. Springer, 2014.

[25] Dirk Helbing. *Social self-organization: Agent-based simulations and experiments to study emergent social behavior*. Springer, 2012.

[26] Tuncer I Ören. Artificial intelligence in simulation. *Annals of operations research*, 53(1):287–319, 1994.

[27] John S Carson. Introduction to modeling and simulation. In *Proceedings of the 36th conference on Winter simulation*, pages 9–16. Winter Simulation Conference, 2004.

[28] Anu Maria. Introduction to modeling and simulation. In *Proceedings of the 29th conference on Winter simulation*, pages 7–13. IEEE Computer Society, 1997.

[29] Feirouz Ksontini, René Mandiau, Zahia Guessoum, and Stéphane Espié. Affordance-based agent model for road traffic simulation. *Autonomous Agents and Multi-Agent Systems*, 29(5):821–849, 2015.

[30] Joachim W Schmidt. Introduction to systems analysis, modeling and simulation. In *Proceedings of the 17th conference on Winter simulation*, pages 3–14. ACM, 1985.

[31] Thomas P Engel. Computer simulation techniques. *Journal of clinical monitoring and computing*, 17(1):3–9, 2002.

[32] Emmet Thurman Hooper. *A systematic review of the harvest mice (genus Reithrodontomys) of Latin America*. Number 77. University of Michigan Press, 1952.

[33] Oliver P Pearson. Habits of harvest mice revealed by automatic photographic recorders. *Journal of Mammalogy*, 41(1):58–74, 1960.

[34] Kazuaki Iwata, Ma Onosato, Ka Teramoto, and Suguru Osaki. A modelling and simulation architecture for virtual manufacturing systems. *CIRP Annals-Manufacturing Technology*, 44(1):399–402, 1995.

[35] Reiko Ishiwaka and Takayuki Mori. Early development of climbing skills in harvest mice. *Animal behaviour*, 58(1):203–209, 1999.

[36] Norbert Noury and Tareq Hadidi. Computer simulation of the activity of the elderly person living independently in a health smart home. *Computer methods and programs in biomedicine*, 108(3):1216–1228, 2012.

[37] Fritz Rohrlich. Computer simulation in the physical sciences. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, pages 507–518. JSTOR, 1990.

[38] Tim Baynes and Scott Heckbert. Micro-scale simulation of the macro urban form: Opportunities for exploring urban change and adaptation. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 14–24. Springer, 2009.

[39] Islem Henane, Lamjed Ben Said, Sameh Hadouaj, and Nasr Ragged. Multi-agent based simulation of animal food selective behavior in a pastoral system. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 283–292. Springer, 2010.

[40] Claudio O Stöckle, Marcello Donatelli, and Roger Nelson. Cropsyst, a cropping systems simulation model. *European journal of agronomy*, 18(3):289–307, 2003.

[41] Roy L Nersesian. *Computer simulation in financial risk management: a guide for business planners and strategists*. Praeger Pub Text, 1991.

[42] Mansoor Dailami, Ilya Lipkovich, and John Van Dyck. Infrisk: a computer simulation approach to risk management in infrastructure project finance transactions. *World Bank Policy Research Working Paper*, (2083), 1999.

[43] Nicholas H G Holford, Hui C Kimko, Jon P R Monteleone, and Carl C Peck. Simulation of clinical trials. *Annual review of pharmacology and toxicology*, 40(1): 209–234, 2000.

[44] Barry W Brook, Julian J O'Grady, Andrew P Chapman, Mark A Burgman, Resit Akçakaya, and Richard Frankham. Predictive accuracy of population viability analysis in conservation biology. *Nature*, 404(6776):385–387, 2000.

[45] Robert C Lacy. Vortex: a computer simulation model for population viability analysis. *Wildlife research*, 20(1):45–65, 1993.

[46] Joanna J Bryson, Yasushi Ando, and Hagen Lehmann. Agent-based modelling as scientific method: a case study analysing primate social behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 362(1485): 1685–1699, 2007.

[47] Van Dyke Parunak, Robert Savit, and Rick L Riolo. Agent-based modeling vs. equation-based modeling: A case study and user guide. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 10–25. Springer, 1998.

[48] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

[49] Averill M Law and David W Kelton. *Simulation modeling and analysis*, volume 2. McGraw-Hill New York, 1991.

[50] Mark Broadie and Paul Glasserman. Pricing americanstyle securities using simulation. *Journal of Economic Dynamics and Control*, 21(8):1323-1352, 1997.

[51] Louis O Scott. Option pricing when the variance changes randomly: Theory, estimation, and an application. *Journal of Financial and Quantitative analysis*, 22 (04):419–438, 1987.

[52] David E Burmaster and Paul D Anderson. Principles of good practice for the use of monte carlo techniques in human health and ecological risk assessments. *Risk analysis*, 14(4):477–481, 1994.

[53] Authur W Warrick and William R Gardner. Crop yield as affected by spatial variations of soil and irrigation. *Water Resources Research*, 19(1):181–186, 1983.

[54] Charles R Standridge and Duane Steward. Using expert systems for simulation modeling of patient scheduling. *Simulation*, 75(3):148–156, 2000.

[55] Robert E Shannon, Richard Mayer, and Heimo H Adelsberger. Expert systems and simulation. *Simulation*, 44(6):275–284, 1985.

[56] David J Spiegelhalter, Philip A Dawid, Steffen L Lauritzen, and Robert G Cowell. Bayesian analysis in expert systems. *Statistical science*, pages 219–247, 1993.

[57] Anna Hart. Knowledge acquisition for expert systems. In *Knowledge, skill and artificial intelligence*, pages 103–111. Springer, 1988.

[58] Fred Collopy and Scott J Armstrong. Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations. *Management Science*, 38(10):1394–1414, 1992.

[59] Robert G Cowell. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media, 2006.

[60] Brandon Bennett, Anthony G Cohn, Frank Wolter, and Michael Zakharyaschev. Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence*, 17(3):239–251, 2002.

[61] Jacoby Carter and John T Finn. Moab: a spatially explicit, individual-based expert system for creating animal foraging models. *Ecological Modelling*, 119(1): 29–41, 1999.

[62] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[63] Paul Davidsson. Multi agent based simulation: beyond social simulation. In *Multi-Agent-Based Simulation*, pages 97–107. Springer, 2000.

[64] Thomas C Schelling. Dynamic models of segregation. *Journal of mathematical sociology*, 1(2):143–186, 1971.

[65] Blake LeBaron. A builder's guide to agent-based financial markets. *Quantitative Finance*, 1(2):254–261, 2001.

[66] Simon Hallé and Brahim Chaib-draa. A collaborative driving system based on multiagent modelling and simulations. *Transportation Research Part C: Emerging Technologies*, 13(4):320–345, 2005.

[67] Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.

[68] Charles M Macal and Michael J North. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, pages 2–15. Winter Simulation Conference, 2005.

[69] Adriano Melo, Mairon Belchior, and Vasco Furtado. Analyzing police patrol routes by simulating the physical reorganization of agents. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 99–114. Springer, 2005.

[70] Yuqing Tang, Simon Parsons, and Elizabeth Sklar. Modeling human education data: From equation-based modeling to agent-based modeling. In *Multi-Agent-Based Simulation VII*, pages 41–56. Springer, 2007.

[71] Maartje Spoelstra and Elizabeth Sklar. Agent-based simulation of group learning. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 69–83. Springer, 2007.

[72] Shah Jamal Alam, Ruth Meyer, and Emma Norling. A model for hiv spread in a south african village. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 33–45. Springer, 2008.

[73] Samer Hassan, Luis Antunes, and Juan Pavón. Mentat: a data-driven agent-based simulation of social values evolution. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 135–146. Springer, 2009.

[74] Lorenza Manenti, Sara Manzoni, Giuseppe Vizzari, Kazumichi Ohtsuka, and Kenichiro Shimura. An agent-based proxemic model for pedestrian and group dynamics: motivations and first experiments. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 74–89. Springer, 2012.

[75] Gennaro Di Tosto and Frank Dignum. Simulating social behaviour implementing agents endowed with values and drives. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 1–12. Springer, 2013.

[76] Pablo Werlang, Michel Q Fagundes, Diana F Adamatti, Karina S Machado, Andrea von Groll, Pedro E A da Silva, and Adriano V Werhli. Multi-agent-based simulation of mycobacterium tuberculosis growth. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 131–142. Springer, 2014.

[77] Vidit Bansal, Ramachandra Kota, and Kamalakar Karlapalem. System issues in multi-agent simulation of large crowds. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 8–19. Springer, 2008.

[78] Kazuo Miyashita. Asap: Agent-based simulator for amusement park. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 195–209. Springer, 2005.

[79] Nelson Minar, Roger Burkhart, Chris Langton, Manor Askenazi, et al. The swarm simulation system: A toolkit for building multi-agent simulations. 1996.

[80] Maksim Tsvetovat and Maciej Łatek. Dynamics of agent organizations: application to modeling irregular warfare. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 60–70. Springer, 2009.

[81] Mazlina Abdul Majid, Peer-Olaf Siebers, and Uwe Aickelin. Modelling reactive and proactive behaviour in simulation. *arxiv*, 2010.

[82] Colin M Henein and Tony White. Agent-based modelling of forces in crowds. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 173–184. Springer, 2005.

[83] Shu-Heng Chen, Chung-Ching Tai, and Shu G Wang. Does cognitive capacity matter when learning using genetic programming in double auction markets? In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 37–48. Springer, 2010.

[84] Chris J Topping, Tine S Hansen, Thomas S Jensen, Jane U Jepsen, Frank Nikolajsen, and Peter Odderskær. Almass, an agent-based model for animals in temperate european landscapes. *Ecological Modelling*, 167(1):65–82, 2003.

[85] Lu Tan, Mingyuan Hu, and Hui Lin. Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. *Information Sciences*, 295:53–66, 2015.

[86] Jake Pauls. Movement of people. *SFPE Handbook of Fire Protection Engineering*, 2:3, 1995.

[87] Gary M Jackson. *Predicting Malicious Behavior: Tools and Techniques for Ensuring Global Security*. John Wiley & Sons, 2012.

[88] David Lee and Mihalis Yannakakis. Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996.

[89] Mikael Fridenfalk. A simplified representation of finite state machines by generic states. In *Nicograph International 2014, Visby, Sweden, May 2014*, pages 163–164, 2014.

[90] Branislav Ulicny and Daniel Thalmann. Crowd simulation for interactive virtual environments and vr training systems. In *Computer Animation and Simulation 2001*, pages 163–170. Springer, 2001.

[91] Pengyu Hong, Matthew Turk, and Thomas S Huang. Gesture modeling and recognition using finite state machines. In *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 410–415. IEEE, 2000.

[92] Jacqueline Jarvis, Dennis Jarvis, and Ralph Rönnquist. *Holonic execution: a BDI approach*, volume 106. Springer Science & Business Media, 2008.

[93] Michael E Bratman, David J Israel, and Martha E Pollack. Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3):349–355, 1988.

[94] Emiliano Lorini and Michele Piunti. Introducing relevance awareness in bdi agents. In *International Workshop on Programming Multi-Agent Systems*, pages 219–236. Springer, 2009.

[95] Steven F Railsback, Steven L Lytinen, and Stephen K Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9): 609–623, 2006.

[96] Mark d'Inverno, David Kinny, Michael Luck, and Michael J Wooldridge. A formal specification of dmars. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 155–176. Springer, 1997.

[97] Gerhard Weir and Sandip Sen. Adaptation and learning in multi-agent systems. In *Proceedings of IJCAI'95 Workshop, Montreal, Canada*, 1996.

[98] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, and Henry Soldano. Learning in bdi multi-agent systems. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 218–233. Springer, 2005.

[99] Dhirendra Singh, Sebastian Sardina, Lin Padgham, and Stéphane Airiau. Learning context conditions for bdi plan selection. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 325–332. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[100] Michael Wooldridge. *Reasoning about rational agents.* MIT press, 2000.

[101] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Transactions of The Society for Modelling and Simulation International*, 81(7):517–527, 2005.

[102] Michael J North, Nicholson T Collier, and Jerry R Vos. Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(1):1–25, 2006.

[103] Edouard Amouroux, Thanh-Quang Chu, Alain Boucher, and Alexis Drogoul. Gama: an environment for implementing and running spatially explicit multi-agent simulations. In *Pacific Rim International Conference on Multi-Agents*, pages 359–371. Springer, 2007.

[104] Bernd Schmidt. *The modelling of human behaviour: The PECS reference models.* SCS-Europe BVBA, 2000.

[105] George Mason University USA. Mason: Multiagent based simulation toolkit, 2016. URL `http://www.cs.gmu.edu/~eclab/projects/mason/Download`.

[106] Nick Collier. Repast: An extensible framework for agent simulation. *The University of Chicago Social Science Research*, 36:2003, 2003.

[107] Pau Fonseca I. Casas. *Formal languages for computer simulation: transdisciplinary models and applications*. Igi Global, 2013.

[108] Seth Tisue and Uri Wilensky. Netlogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence, Chicago, IL*, pages 7–9, 2004.

[109] Dustin R Rubenstein and Keith A Hobson. From birds to butterflies: animal movement patterns and stable isotopes. *Trends in Ecology & Evolution*, 19(5): 256–263, 2004.

[110] Joyce H Poole, Peter L Tyack, Angela S Stoeger-Horwath, and Stephanie Watwood. Animal behaviour: elephants are capable of vocal learning. *Nature*, 434(7032):455–456, 2005.

[111] Shinichi Watanabe, Masako Izawa, Akiko Kato, Yan Ropert-Coudert, and Yasuhiko Naito. A new technique for monitoring the detailed behaviour of terrestrial animals: a case study with the domestic cat. *Applied Animal Behaviour Science*, 94(1):117–131, 2005.

[112] Neil J Vickers. Mechanisms of animal navigation in odor plumes. *The Biological Bulletin*, 198(2):203–212, 2000.

[113] Luca Börger, Benjamin D Dalziel, and John M Fryxell. Are there general mechanisms of animal home range behaviour? A review and prospects for future research. *Ecology letters*, 11(6):637–650, 2008.

[114] Michael H Dickinson, Claire T Farley, Robert J Full, Mimi A Koehl, Rodger Kram, and Steven Lehman. How animals move: an integrative view. *Science*, 288(5463): 100–106, 2000.

[115] Reena Mathur. *Animal Behaviour*. Global Media, 2010.

[116] Laura Betzig, Monique Borgerhoff Mulder, and Paul Turke. *Human reproductive behaviour: A Darwinian perspective*. CUP Archive, 1988.

[117] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1990.

[118] Johannes S Odendaal. Animal-assisted therapy-magic or medicine? *Journal of psychosomatic research*, 49(4):275–280, 2000.

[119] Eric M Leroy, Brice Kumulungui, Xavier Pourrut, Pierre Rouquet, Alexandre Hassanin, Philippe Yaba, André Délicat, Janusz T Paweska, Jean-Paul Gonzalez, and Robert Swanepoel. Fruit bats as reservoirs of ebola virus. *Nature*, 438(7068): 575–576, 2005.

[120] Bastiaan G Meerburg, Grant R Singleton, and Herwig Leirs. The year of the rat ends, time to fight hunger! *Pest Management Science*, 65(4):351–352, 2009.

[121] Christopher J Barnard. *Animal behaviour: mechanism, development, function and evolution.* Pearson Education, 2004.

[122] Pia K Leppänen, Béatrice M Ewalds-Kvist, and Ritva-Kajsa Selander. Mice selectively bred for open-field thigmotaxis: life span and stability of the selection trait. *The Journal of general psychology*, 132(2):187–204, 2005.

[123] Dallas Treit and Marian Fundytus. Thigmotaxis as a test for anxiolytic activity in rats. *Pharmacology Biochemistry and Behavior*, 31(4):959–962, 1988.

[124] Stephen V Kaye. Movements of harvest mice tagged with gold-198 wires. *Journal of Mammalogy*, 42(3):323–337, 1961.

[125] Oliver P Pearson. The prey of carnivores during one cycle of mouse abundance. *The Journal of Animal Ecology*, pages 217–233, 1966.

[126] Marco Perugini and Richard P Bagozzi. The distinction between desires and intentions. *European Journal of Social Psychology*, 34(1):69–84, 2004.

[127] Richard P Bagozzi. The self-regulation of attitudes, intentions, and behavior. *Social psychology quarterly*, pages 178–204, 1992.

[128] Richard P Bagozzi and Utpal M Dholakia. Intentional social action in virtual communities. *Journal of interactive marketing*, 16(2):2–21, 2002.

[129] David Fraser. Animal ethics and animal welfare science: bridging the two cultures. *Applied Animal Behaviour Science*, 65(3):171–189, 1999.

[130] Robert F Sproull. Using program transformations to derive line-drawing algorithms. *ACM Transactions on Graphics (TOG)*, 1(4):259–273, 1982.

[131] Michael L V Pitteway and Andrew J Green. Bresenham's algorithm with run line coding shortcut. *The Computer Journal*, 25(1):114–115, 1982.