ORPHEUS: Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation

Mihai Polceanu

▶ **To cite this version:**

Mihai Polceanu. ORPHEUS: Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation. Artificial Intelligence [cs.AI]. Université de Bretagne Occidentale (UBO), 2015. English. NNT : . tel-01203388

HAL Id: tel-01203388

https://theses.hal.science/tel-01203388

Submitted on 22 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thesis / University of Western Brittany
under the seal of European University of Brittany

for obtaining the title of
Doctor of the University of Western Brittany
Mention : Computer Science
Doctoral school Sicma

presented by
Mihai POLCEANU
prepared at the European Center
for Virtual Reality
Lab-STICC Laboratory

O.R.P.H.E.U.S.: Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation

Defended on September 4th 2015
before the jury composed of :

- **Nicolas Maudet** (reviewer)
  Research Director, UPMC, France
- **Ubbo Visser** (reviewer)
  Research Director, University of Miami, USA
- **Fred Charles** (examiner)
  Research Director, Teesside University, UK
- **Dorin-Mircea Popovici** (examiner)
  Professor, OVIDIUS, Romania
- **Vincent Rodin** (examiner)
  Professor, UBO, France
- **Cédric Buche** (supervisor)
  Associate Professor HDR, ENIB, France

Thesis / University of Western Brittany
under the seal of European University of Brittany

for obtaining the title of
Doctor of the University of Western Brittany
Mention : Computer Science
Doctoral school Sicma

presented by
Mihai POLCEANU
prepared at the European Center for Virtual Reality
Lab-STICC Laboratory

O.R.P.H.E.U.S.: Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation

Defended on September 4th 2015
before the jury composed of :

- **Nicolas Maudet** (reviewer)
  Research Director, UPMC, France
- **Ubbo Visser** (reviewer)
  Research Director, University of Miami, USA
- **Fred Charles** (examiner)
  Research Director, Teesside University, UK
- **Dorin-Mircea Popovici** (examiner)
  Professor, OVIDIUS, Romania
- **Vincent Rodin** (examiner)
  Professor, UBO, France
- **Cédric Buche** (supervisor)
  Associate Professor HDR, ENIB, France

Abstract

Interactive virtual environments pose a wide variety of challenges for intelligent agents, especially to make decisions in order to reach their goals. The difficulty of decision making tasks rises quickly when introducing continuous space and real time. It also becomes increasingly harder to build intelligent agents that can meaningfully interpret and act in unknown situations. In this thesis, we take inspiration from cognitive science, specifically from how humans perform mental simulation to anticipate events in the world around them, with the aim of obtaining an autonomous agent that makes decisions and adapts itself to novel situations. The mental simulation paradigm enjoys significant interest from the cognitive science community, but computational approaches to mental simulation rely on specialized simulators for a given task and thus are limited to specific scenarios. Our contribution is a generic agent architecture (ORPHEUS) which supports decision-making based on the simulation of functional models of the world ahead of time, inspired from how humans imagine the outer world and the outcomes of their actions based on the state of the real environment. The novelty of our approach consists in its ability to integrate both physical and behavioural predictions into the same framework, based on heterogeneous mental models which are used to evolve internal, imaginary scenarios within the agent. We apply our generic architecture to different contexts, including artificial intelligence competitions, which require the agent using our approach to perform physical and behavioural anticipation in continuous space and time. We evaluate the applicability of our approach to realistic conditions such as noisy perception, decision time constraints and imperfect world models. Results demonstrate the genericness of our approach from one scenario to another without modifying the agent architecture, and highlight the possible uses of the proposed mental simulation framework.

Contents

Abstract v

Contents vii

List of Figures ix

List of Tables xiii

1 Introduction 1
- 1.1 Cognitive Approach Towards Computational Decision Making . 2
  - 1.1.1 Mental Simulation: a Common Framework 2
  - 1.1.2 Assessing the Applicability of Mental Simulation 2
- 1.2 Objectives 4
- 1.3 Manuscript Organization 5

2 Related Work 7
- 2.1 Environment Type 10
  - 2.1.1 Virtual World 10
  - 2.1.2 Real World 11
- 2.2 Mental Simulation Targets 12
  - 2.2.1 Environmental Aspects 12
  - 2.2.2 Behavioural Aspects 13
- 2.3 Cognitive Function 19
  - 2.3.1 Prospection 19
  - 2.3.2 Navigation 20
  - 2.3.3 Theory of Mind 20
  - 2.3.4 Counterfactual Reasoning 21
- 2.4 Computational Implementation 22
- 2.5 Discussion 23

3 Proposition: ORPHEUS 29
- 3.1 Generic Agent Architecture 30
  - 3.1.1 Perception 32
  - 3.1.2 Imaginary World 32

3.1.3 Abstract World . . . 41
3.2 Implementation . . . 44
3.2.1 Software Architecture . . . 44
3.2.2 Distributed Processing and Visualization . . . 55
3.3 Discussion . . . 57

4 Applications **61**
4.1 Introduction . . . 62
4.2 Angry Birds Agent . . . 63
4.2.1 Perception . . . 64
4.2.2 Abstract world . . . 66
4.2.3 Imaginary World . . . 66
4.2.4 Decision and Control . . . 67
4.2.5 Preliminary Results . . . 67
4.2.6 Competition Results . . . 69
4.3 Orphy the Cat . . . 71
4.3.1 Perception . . . 72
4.3.2 Abstract world . . . 73
4.3.3 Imaginary World . . . 73
4.3.4 Decision and Control . . . 74
4.3.5 Results . . . 74
4.4 Nao Robot Goalkeeper . . . 79
4.4.1 Perception . . . 81
4.4.2 Abstract world . . . 82
4.4.3 Imaginary World . . . 83
4.4.4 Decision and Control . . . 85
4.4.5 Results . . . 86
4.5 Discussion . . . 88

5 Conclusion **91**
5.1 Discussion . . . 93
5.1.1 Contributions of Orpheus . . . 94
5.1.2 Limitations of Orpheus . . . 95
5.2 Future Work . . . 96
5.2.1 Nao Soccer Lab . . . 96
5.2.2 Learning . . . 97
5.2.3 Planning . . . 98
5.2.4 Agent's Goals . . . 99
5.2.5 Emotions . . . 99
5.3 Publications . . . 101

Bibliography **103**

List of Figures

2.1 Excerpts from works using mental simulation in understanding similar agents. . . . 14
2.2 Excerpts from works performing imitation learning with the aid of mental simulation. . . . 15
2.3 Excerpts from works performing mental simulation based behaviour prediction on agents with different structure. . . . 16
2.4 Excerpts from works using mental simulation for collaboration with humans or other robots. . . . 17
2.5 Excerpts from works performing mental simulation of the agent's own self. . . . 18
2.6 Birthday present example: in order to arrive in the position to hand the gift, the agent was required to (a) reason on which type of gift would be most suitable, (b) find which way through town would be best to go to the store, (c) have an understanding of what the other wishes for a gift and (d) imagine that not bringing a present would have had undesirable results. . . . 19

3.1 Conceptual overview of the "simulation within the simulation" framework [Buche et al. (2010); Buche and De Loor (2013)]. . . . 30
3.2 Functionality of our architecture at an arbitrary time t: the state of the real environment $r(t)$ is transformed into a mental image $s(t+1)$ through perception, from which alternative futures ($s(t+n)$, $s'(t+n)$) can be obtained, past events $s(t-m)$ can be reenacted or changed to produce "what if" scenarios (s'). Decisions can be based on evaluating alternative futures. . . . 31
3.3 Visual perception of the real world evolution – $r(t)$ – as a series of timestamped images. . . . 32
3.4 Belief representation using entities with properties approach. . . . 33
3.5 Uncontrollable alternative futures. . . . 35
3.6 Controllable alternative futures. . . . 35
3.7 The mechanism through which a mental image is evolved in one step ($i \in [0, n-1]$), by interweaving effects from models. . . . 37

3.8 Mental model seen as a black box, controllable through an action interface. . . . 38
3.9 Action synthesis from multiple mental simulation results. . . . 40
3.10 Functionality expected from a mental model within our agent architecture. . . . 42
3.11 Packages of the ORPHEUS agent architecture and their interdependency. . . . 45
3.12 Class diagram - principal components of the `SimulationEngine` package. . . . 45
3.13 Class diagram - action synthesis in the `SimulationEngine` package. 47
3.14 Class diagram - creating system goals. . . . 49
3.15 Class diagram - the `Metamodel` package: mental images and their structure. . . . 50
3.16 Class diagram - mental model abstraction and on-demand creation. 51
3.17 Mental model abstraction - integration of a heterogeneous (both structurally and functionally) collection of models at runtime. . . . 53
3.18 Class diagram - mental model training and caching. . . . 53
3.19 Class diagram - `Communication` package. . . . 55
3.20 Resource management for mental simulations over a network of agents. Letters A-F signify simulated scenarios for which computation can be done by other agents in the network. Example: the orange agent computes scenario E for the blue agent. . . . 56
3.21 Network-based Graphical User Interface (GUI) for the agent's decision-making process. **Left:** ORPHEUS agent in virtual environment (Roboviz renderer by Stoecker and Visser (2012)); **Right:** visualization of decision-making mental simulations of the ORPHEUS agent which controls the goalkeeper. **Context:** Virtual Nao goalkeeper for the RoboCup 3D Simulation League, in collaboration with the RoboCanes team[1]. . . . 57

4.1 Annotated in-game view of Angry Birds. . . . 63
4.2 Conceptual overview for the Angry Birds bot. . . . 64
4.3 Obtaining correct object positions: (a) original scene image, (b) default bounding boxes, (c) improved algorithm, (d) obtained virtual reconstruction. . . . 65
4.4 Mental simulation tree with two levels: trajectories and tap times. 66
4.5 Best solution for Level 2 of "Poached Eggs" episode, found by our implementation (overlaid original state and highlighted trajectory) 69
4.6 Instantiation of the "real world" (3D environment) featuring the cat and autonomous birds (left), and the "imaginary world" of the cat with examples of mental simulations where it evaluates different action sequences (right). . . . 71
4.7 Conceptual overview for Orphy the cat. . . . 72

4.8 Models assigned to each entity in the imaginary world and action controller for the system's self model (cat). All models contribute to the evolution of the mental simulation at each step. 73
4.9 Agent behaviour in scenarios with cuboid obstacles (four black squares). Case (1) corresponds with 0 *units* view distance for birds, case (2) with 50 *units* and case (3) with 100 *units*. Subfigures (#.a) illustrate cat (grey) and caught birds (black) traces. Distance histograms are shown for the cat (#.b) and catches (#.c) relative to the closest obstacle. 76
4.10 Agent behavior in scenarios with cylindrical obstacles (four black circles). Case (4) corresponds with 0 *units* view distance for birds, case (5) with 50 *units* and case (6) with 100 *units*. Subfigures (#.a) illustrate cat (grey) and caught birds (black) traces. Distance histograms are shown for the cat (#.b) and catches (#.c) relative to the closest obstacle. 77
4.11 Examples of solutions found by the system for prey catching: (a) simple chase, (b) ambush, (c) hide and chase, (d) blockade. Cat traces shown with grey arrows, and catch locations with directed circles based on bird orientation. 78
4.12 Annotated RoboViz [Stoecker and Visser (2012)] visualization of a RoboCup 3D Simulation League world championship final between the teams *UT Austin Villa* and *RoboCanes*, held in João Pessoa, Brazil, in 2014. 80
4.13 Conceptual overview for the Nao goalkeeper. 81
4.14 Comparison of ball velocity values for X (left) and Y (right) axes over time, since the kick. 82
4.15 Long step action (left) with associated learned mental model (right). 83
4.16 Outcome prediction accuracy, with ball Estimated Time of Arrival (ETA). 84
4.17 The decision process of the ORPHEUS goalkeeper. 85

5.1 Nao Robot Soccer Lab, Centre Européen de Réalité Virtuelle (CERV), Brest, France. **Left**: initial tests; **Right**: goalkeeper evaluation session. 96
5.2 Nao goalkeeper evaluation interface by graduate students at the CERV. 97
5.3 Preliminary real time behaviour learning results in the "Orphy the Cat" application presented in this thesis. Traces show mental simulations of the learned trajectory of a bird, after 5 to 10 seconds of observation, in which three KRLS mental models collectively learn different aspects of the bird's behaviour. 98

List of Tables

2.1 Overview of areas of interest covered by relevant existing approaches. 24
2.2 Pros and Cons of approaches to computational mental simulation. 24

4.1 Challenges in each proposed evaluation scenario of Orpheus. . . . 62
4.2 Scores of the default naive agent. 68
4.3 Scores of the mental simulation agent 68
4.4 AIBirds 2013 competition results of top 10 entries (from 20 total), including our agent (IHSEV). 70
4.5 AIBirds 2014 competition results, including our agent (IHSEV). . 70
4.6 Goalkeeper success rates for 100 random kicks from different distances. The kick angles are chosen randomly from -45 to 45 degrees. Each rate is averaged over 30 repetitions of the 100 kicks (standard deviation shown in parentheses). 87
4.7 Average goalkeeper success rates for 100 random kicks as in Table 4.6, here grouped by angle (using the distance 5-7m). 87
4.8 Challenges identified in our study on related work, that were overcome by the proposed agent architecture, Orpheus. 88

List of Algorithms

1 Mental simulation 46
2 Estimating computation time 47
3 Action synthesis 48
4 Applying the default *SynthesizerPolicy* 49
5 Training session 54

Contents of Chapter 1

1 Introduction **1**
1.1 Cognitive Approach Towards Computational Decision Making . 2
1.1.1 Mental Simulation: a Common Framework 2
1.1.2 Assessing the Applicability of Mental Simulation 2
1.2 Objectives 4
1.3 Manuscript Organization 5

Chapter 1

Introduction

Summary of Chapter 1

This chapter introduces the goals and objectives of this thesis and the approach we take in order to achieve them. Within this work, we propose to model the decision mechanisms of an artificial agent by drawing inspiration from studies in cognitive science on mental simulation. Our focus is on the human's anticipation ability and capability to adapt while interacting. Herein, the reader is provided with insight on the dynamic context to which we apply our approach and the efforts that have been made by cognitive science research in studying the mechanisms behind human decision making. We discuss mental simulation and its applicability to intelligent agents, and provide an overview of what can be achieved via mental simulation, therefore presenting our motivation to implement an agent that functions based on this paradigm. Extrapolating on what is expected to be achievable through mental simulation, the objectives to be achieved in this work are outlined. Finally, the structure of the remainder of this thesis is introduced.

The research conducted in this thesis is situated within the field of artificial intelligence and focuses on autonomous agent cognition in dynamic environments. Our work aims to bridge concepts from cognitive science with computational techniques towards building an intelligent agent that is capable of more than it has been programmed to do.

The objective is to obtain an agent that can **adapt** to the changing conditions in its environment and that can do so in a timely manner. More precisely, our agent should **anticipate** events and interactions that happen in its world, and take the necessary actions depending on the context.

Our approach to this problem consists in taking inspiration from cognitive science studies on **mental simulation**, a mechanism central to decision making in humans, and formulating a computational approach to this concept.

1.1 Cognitive Approach Towards Computational Decision Making

To affirm that an artificial intelligence system reasons like a human, we are required to know how humans think by acquiring insight of the mechanisms of the mind. There are several ways to achieve this: by introspection, psychological experiments on behaviour and brain imaging. This work can help to formalise a mental simulation architecture that is clear enough to make it possible to be expressed in the form of a computer program. The interdisciplinary field of cognitive science advances in this direction by combining experimental psychology techniques and computer models of computational intelligence. In this context, mental simulation is a mechanism that appears to be a particularly interesting approach to modelling the process of decision making [Hesslow (2002); Berthoz (1997, 2003)].

1.1.1 Mental Simulation: a Common Framework

Evidence of mental simulation has been studied in humans [Buckner and Carroll (2007); Decety and Grèzes (2006)] where similar brain regions have been observed to become activated when individuals perform remembering, prospection and Theory of Mind [Premack and Woodruff (1978)] tasks. Humans have the ability to remember past events in a rich, contextual way, that allows them to revisit those memories and apply gained knowledge to novel scenarios. Planning generally takes the form of prospection in humans, where future scenarios are visualised in one's mind, enabling the individual to choose what actions to perform based on their imagined utility. When in a social context, theorising and reasoning about what others think and feel – *i.e.* Theory of Mind – are crucial skills for successful interaction with other people.

Research also extends into how people use this type of simulation to mentally represent mechanical systems [Hegarty (2004)] and to reason on how these will evolve in various conditions. It was suggested that the prediction ability of this approach can be limited by the perceptual capability of the individual and that, in complex situations, subjects cannot accurately predict the precise motion trajectories, but can evaluate within a reasonable margin of error what the effects would be.

From a computational perspective, studies on mental simulation in humans suggest that this paradigm offers a common framework for reasoning about one's environment as a whole and therefore aid the creation of an architecture that would enable an agent to reach a higher level of adaptability.

1.1.2 Assessing the Applicability of Mental Simulation

Advances in cognitive science suggest the need of a unified framework of human cognition, while mental simulation and the theory of mind receive much

attention as fundamental human traits for social reasoning. In the field of artificial intelligence, a variety of cognitive architectures have been proposed [Langley et al. (2009)], but only few feature mental simulation. Moreover, those that do include mental simulation make use of it within localised contexts and by using models of the environment and human psychology which are constructed by human experts, and not by the agent itself.

Mental models allow an individual to anticipate certain aspects of the future state of the environment. From the way humans can, in some way, visualise scenarios through mental imagery, there is reason to believe that such models should be executable, so that an agent can apply them to the current state of the environment and obtain the next state as a result. This simulation process provides a version of the future that can be used to make decisions. Executable knowledge has been tackled in some cognitive architecture implementations [Kennedy et al. (2009)] that use sets of production rules that apply to certain preconditions and yield a new state of the system. However, these approaches require world models which must be developed beforehand making their usability cumbersome and restricted to specific problems.

Through the means of mental simulation, an agent can have access to an image of the future. Hence, the agent can compute a strategy to cope with the imagined situation before the future state is reached and therefore act in a timely manner when the event occurs. Deciding when to apply the strategy may depend on time, if the anticipated event is time-dependent (such as a moving object) or precondition-dependent (an expected event that is known to happen if certain cues occur in the environment). An example of such precondition-dependent anticipation would be avoiding passing cars while crossing a street, where it is expected that cars will pass but the precondition for avoidance is the recognition of a car. Clearly, after a car has been recognised, its trajectory can be anticipated.

Aiming to obtain an intelligent agent which can act autonomously in a complex environment, this thesis consists in a novel mental simulation based approach, without relying classical production rule based paradigms which generally require human intervention to formally describe problems which need to be solved. This does not imply however to completely remove the human from the process, but to allow the design of an agent that can function without intervention in an environment where humans can participate.

1.2 Objectives

The main purpose of this thesis is to obtain an agent that makes decisions in various dynamic environments, based on the paradigm of mental simulation. An intermediary aim is to minimise the set of assumptions that are used to achieve the agent's behaviour, in order to maintain genericness. Therefore, the objectives of this work can be summarised as follows:

- Develop a **generic** approach which can be used in a variety of environments. Most implementations of cognitive architectures are susceptible to difficulties when the application scenario is modified or radically changed.
- Design for **autonomy**. More generic implementations tend to be highly difficult to use in scenarios that are more complex, requiring much contribution from domain experts who must model the problem for the agent to reason on. Therefore minimum expert intervention is highly desirable in such approaches.
- Focus on **online** anticipation and learning. To be able to autonomously function within a more complex and dynamic environment, the agent must be able to **adapt** to new conditions and learn while acting.
- Finally, achieve behaviour **believability**. The property of being believable grants the agent a higher level of interaction potential with human users. While the concept of believability is a subjective matter, similar-to-human behavioural traits have been shown to favour the perception of the agent as a human, by other humans.

1.3 Manuscript Organization

This section aims to give an overview of what is discussed and the reason behind each study.

Chapter 2 is dedicated to the **current state of the art**. It studies how concepts from psychology, cognitive science and neuroscience have been implemented into agents and what the capabilities of these agents are. This survey is important to identify both the shortcomings and strong points of these approaches, in an attempt to obtain an improved architecture that can allow an agent to anticipate and reason within its environment.

A framework proposal is described in **Chapter 3** which aims to satisfy the previously stated objectives. Due to the complexity of human cognition, it may not be possible to achieve a computational replica without reconstructing all mechanisms that make it possible. However, by identifying the fundamental paradigms which enable such high level of cognition, attempts can be made to create a generic architecture which would allow a wide range of similar functionality.

Within **Chapter 4** the proposed framework is analysed and tested within different scenarios in order to evaluate its **applicability**. The choice for scenarios aims to target relevant situations in which the capabilities of the approach can be assessed, while avoiding over-specialization.

Finally, our work is **concluded in Chapter 5** by stating what the approach can provide, what it cannot and how it may be improved in further work.

Contents of Chapter 2

2 Related Work **7**

2.1 Environment Type . . . 10
2.1.1 Virtual World . . . 10
2.1.2 Real World . . . 11
2.2 Mental Simulation Targets . . . 12
2.2.1 Environmental Aspects . . . 12
2.2.2 Behavioural Aspects . . . 13
2.2.2.1 Agents with Similar Structure . . . 13
2.2.2.2 Agents with Different Structure . . . 15
2.2.2.3 The Self . . . 17
2.3 Cognitive Function . . . 19
2.3.1 Prospection . . . 19
2.3.2 Navigation . . . 20
2.3.3 Theory of Mind . . . 20
2.3.4 Counterfactual Reasoning . . . 21
2.4 Computational Implementation . . . 22
2.5 Discussion . . . 23

Chapter 2

Related Work

Summary of Chapter 2

This chapter is dedicated to the study of existing approaches that explicitly use mental simulation to achieve anticipation and decision making. Current implementations of the mental simulation paradigm in related works, taken together, computationally address many aspects suggested by cognitive science research. Agents are able to find solutions to non-trivial scenarios in virtual or physical environments. Existing systems also learn new behaviour by imitation of others similar to them and model the behaviour of different others with the help of specialised models, culminating with the collaboration between agents and humans. Approaches that use self models are able to mentally simulate interaction and to learn about their own physical properties. Multiple mental simulations are used to find solutions to tasks, for truth maintenance and contradiction detection. However, individual approaches do not cover all of the contexts of mental simulation and they rely on techniques which are only suitable for subsets of obtainable functionality. This paper spans through four perspectives on the functionality of state of the art artificial intelligence applications, while linking them to cognitive science research results. Finally, an overview identifies the main gaps in existing literature on computational mental simulation.

Interactive virtual environments pose a wide variety of challenges for intelligent agents, especially to make decisions in order to reach their goals. The difficulty of decision-making tasks rises quickly by introducing continuous space and real time into question [Doya (2000)]. It also becomes increasingly harder to build intelligent agents that can meaningfully interpret and act in unknown situations.

Classical approaches to decision making, such as production systems [Anderson (1993); Laird (2012)], semantic networks and other formal knowledge representation frameworks [Negnevitsky (2005)] require domain experts to provide descriptions of the agents' environments, the types of objects to be used and entities to interact with [Castillo (1997)]. Upon these approaches, learning algorithms have been used to learn new rules [Fürnkranz (1999); Kavšek and Lavrač (2006)] and policies [Sutton and Barto (1998)] that tell the agent how to behave in various situations.

However, their application is limited when dynamic environments are considered [Brooks (1990)], where agents must assess multiple interactions between entities and their environment, such as the effects of collision, object shape, action timing and visual occlusion on behaviour. Research efforts have been made to address the issues posed by dynamic environments and have yielded important results, such as in robotics [Visser and Burkhard (2007)], but challenges still remain that span over several research fields. Concurrently, cognitive architectures have been proposed to integrate multiple techniques into autonomous agents that are better suited for dynamic environments, but some significant aspects such as anticipation and adaptation are still weakly integrated [Vernon et al. (2007)] in existing approaches.

For intelligent agents to interact with their environment and other agents, they must have an understanding of how the world around them functions, what effects their actions will have and what decisions are optimal in given scenarios. With the aim to achieve the implementation of such an agent, the approach taken in this thesis relies on mental simulation based anticipation. This paradigm has received moderate attention by other researchers, but insufficiencies exist and a generic approach that can function in both simple and complex scenarios has not yet been proposed.

The mental simulation paradigm enjoys significant interest from the cognitive science community [Kahneman and Tversky (1981); Berthoz (1997); Grezes and Decety (2001)]. It is used to explain how humans make certain predictions for making decisions, imagining "what if" scenarios (multiple worlds) and revisiting past events in novel ways [Moulton and Kosslyn (2009)]. Moreover, there exists evidence that mental simulation is not strictly a human capability, but that some animals may also be able to perform it for goal-oriented decision making [Chersi et al. (2013)]. The principle of mental simulation consists in constructing an imaginary world that can function on its own, based on which various future states of the environment can be inferred and decided upon, resulting in an individual's behaviour.

This chapter aims to identify and classify existing approaches in literature that rely on the mental simulation paradigm or that make use of it to enhance the functionality of an agent. The technique of using computational mental simulation in decision making can be applied in various contexts depending on the type of environment, whether simulation targets are objects or intelligently behaving entities and the level of realism the simulations reflect.

Consequently, this study is structured based on four questions about the **application of mental simulation in an artificial intelligence system**:

- **Where** is the mental simulation based system used ?
  (Section 2.1)
- **What** is mental simulation used for within these systems ?
  (Section 2.2)
- **Why** is mental simulation useful in a given case ?
  (Section 2.3)
- **How** do these systems implement computational mental simulation ?
  (Section 2.4)

To answer these questions, we analyse the properties of existing approaches from each point of view. Hence, to answer **where** these systems are used, we compare approaches that are applied to virtual reality and those which have a robotic implementation. Thereafter we investigate **what** these systems are able to anticipate by making use of their mental simulation capabilities, by surveying their ability to predict physical phenomena and behavioural traits of other agents. From the point of view of cognitive science, we are also interested in **why** they require mental simulation – *i.e.* which of the cognitive functions, namely prospection, navigation, Theory of Mind and counterfactual reasoning, are accomplished through this mechanism. Finally, we look into **how** these systems are built, where we classify approaches based on three major schools of thought in artificial intelligence: symbolic, sub-symbolic and analogical representation systems.

This structure allows a clearer analysis of existing approaches that use the simulation paradigm and shows how various implementations fit into a generic view of using mental simulation for decision making. Moreover, to the best of our knowledge, there exists no comprehensive survey on computational approaches to mental simulation. The remainder of this chapter is structured into these four aspects of the computational use of mental simulation and their more specific sub-cases.

Finally, the chapter concludes with an overview (section 2.5) of existing approaches and how they cover the multiple contexts of using simulation in decision-making. The techniques used for each context are summarised and the objectives of this thesis are motivated.

2.1 Environment Type

Artificial intelligence applications, since their advent in the second half of the twentieth century, have diversified to tackle many areas of human intelligence. Research in this field has led to optimal algorithms on a number of problems and super-human performance on others such as in the 90s, when the Deep Blue computer [Hsu (2002)] won against a chess world champion. However, humans still excel in many quotidian tasks such as vision, physical interaction, spoken language, environmental and behavioural anticipation or adapting ourselves in the constantly changing conditions of the natural world in which we live. To this end, hard problems have often been idealised in computer simulations (virtual reality) where research could focus on the essentials of the artificial agents' intelligence without the need to solve low-level problems like noisy perception, motor fatigue or failure to name a few. Once matured in virtual reality, such agents would be ready for embodiment into a robotic implementations where, only few prove to be feasible. From this point of view, we can categorise existing approaches through the prism of environment complexity, namely those that have been implemented in virtual environments (Subsection 2.1.1) and those that have a robotic embodiment (Subsection 2.1.2)

2.1.1 Virtual World

The challenge for an intelligent agent in a virtual world is to cope with potentially complex behaviour, but in an accessible sandbox context. The virtual world is a controlled environment where observing events and object properties is simplified so that agent development can focus on behaviour while neglecting problems that arise from interfacing with the world. An example of such simplicity is given by the trajectory of an object moving under the effects of gravity, whose exact coordinates can be directly sampled by the agent without requiring to capture, segment and analyse an image. The main characteristic that describes this environment type is the focus on behaviour, but this brings the drawback of possible poor scaling of developed methods towards the real environment due to noise, uncertainty and interface issues.

Regarding computational approaches to mental simulation, most existing works have been evaluated in virtual environments of varying complexity. Discrete environments provide a simple but informative view of the behaviour of an agent [Ustun and Smith (2008)], under controllable circumstances. As complexity rises, namely the transition from discrete to continuous space, the challenge for intelligent agents to perform tasks increases significantly, but it also enables a wider range of behaviour. Only now does the use of **mental simulation** begin to find its applications, and **advantages over traditional methods**, in agent decision-making. Literature provides mental simulation approaches to constrained 2-dimensional continuous space [Kennedy et al.

(2008, 2009); Svensson et al. (2009); Buche et al. (2010)] which focus on developing models to cope with the increased complexity of the environment. Other works go even further, to **continuous 3-dimensional space** where trajectories are more dynamic as human users intervene [Buche and De Loor (2013)] and collisions [Kunze et al. (2011a)] or occlusions [Gray and Breazeal (2005)] take place.

A recent trend that relates to the paradigm of mental simulation is the application of Monte Carlo Tree Search (MCTS) to real time video games, as an enhancement to its history of success in turn-based games. Succinctly, if given the capability to simulate the outcomes of its actions, an agent can rely on MCTS planning algorithms such as UCT [Kocsis and Szepesvári (2006)] to perform more efficiently in real-time strategy (RTS) games [Balla and Fern (2009); Churchill and Buro (2013)] or similar scenarios. We note that MCTS consists in planning algorithms **reliant on a simulator** – *i.e.* a way to obtain the effects of performed actions – while **mental simulation encompasses the mechanisms for constructing such a simulator**, which could eventually be used together with heuristic planning techniques.

2.1.2 Real World

In the real world, agents require a physical embodiment (or interface) in order to interact with the environment. The challenge of performing mental simulation within a real setup is to anticipate the behaviour of real entities which are perceived through noisy and fault-prone sensory input. In addition to issues that exist in virtual worlds, reality poses further obstacles to object detection and recognition, and therefore can be viewed as a **significantly more complex** version of a continuous 3-dimensional virtual world. Systems that aim to achieve functionality in the real world must also solve interface problems in order to exhibit their behaviour. Interface issues include acquiring adequate information from sensors and effectors, the possibility of externally caused damage to the system and environment noise.

Several systems using mental simulation have been developed as **dually-compatible** with both virtual environments and robotic embodiments. This allowed the authors to evaluate the cognitive process of their approach [Gray and Breazeal (2005); Breazeal et al. (2009)] in virtual reality where the agent can perform more dexterous actions that its robotic counterpart. Computer simulations were also used by Kennedy et al. (2008, 2009) to evaluate their approach to improving a robot's performance within a team.

Other researchers have directly approached reality with robots that use mental simulation to support their natural **language skills** [Roy et al. (2004)], **reasoning** [Cassimatis et al. (2004)] and **resilience** [Bongard et al. (2006)]. We note that these are **difficult problems in robotics**, and it is interesting that **mental simulation** is able in these cases to **decrease complexity** of the original tasks and allow robots to perform better in the real world.

2.2 Mental Simulation Targets

Depending on what the agent encounters in its environment, the use of mental simulation in existing research can be divided into two categories: inanimate objects and entities which exhibit some form of behaviour. In the following, we explore **what** existing approaches use mental simulation for, namely the environmental (Subsection 2.2.1) and behavioural (Subsection 2.2.2) aspects of their environment.

2.2.1 Environmental Aspects

One aspect of mental simulation is represented by anticipating how insentient systems evolve based on a model of the laws that govern their behaviours. Such systems can be composed objects that move according to the laws of physics or deterministic mechanisms such as, for example, a light switch that can be used to turn a light bulb on and off. Such systems are "simple" in the sense that the underlying rules are deterministic and exhibit little or no change over time, for example applying a force to an object will always trigger a mass-dependent acceleration on that object; this does not exclude the potential complexity of such system.

Possessing a mental model of physical phenomena allows humans to anticipate the consequences of actions that are performed in the environment [Hegarty (2004)]. Having a representation of properties such as mass, gravity, elasticity and friction are necessary in successful predictions of mechanical outcomes. Humans tend to construct a mental image of a given scenario, as it would visually appear in reality, in order to reason in certain contexts [Bergen (2005)]. The ability of humans to analyse environmental information has been linked to their capability of focusing on relatively small sets of data, through the process of attention management [Gross et al. (2004)], due to not being able to process the entire depth of the observable world. Nevertheless, humans are proficient at high precision tasks such as anticipating and counterbalancing weights using body movements [Dufossé et al. (1985)].

A wide range of approaches have been proposed to control an agent's behaviour in complex physical environments, with arguably one of the most successful being those based on Reinforcement Learning (RL) [Sutton and Barto (1998); Kormushev et al. (2013)]. While particularly well suited for noisy, real world data, these approaches usually assume a direct function between the agent's sensors and effectors, which can ultimately lead to limited scalability of their adaptiveness [Atkeson and Santamaria (1997)] in novel scenarios. In this sense, our focus turns to adaptability using **internal model** based approaches Pezzulo et al. (2013).

From the perspective of anticipating environmental aspects, existing research on computational mental simulation makes use of this paradigm to predict object trajectories [Buche and De Loor (2013)] and eventual collision

between them [Kunze et al. (2011b)]. Mental models can also serve to represent objects that exit the perception field Roy et al. (2004); Cassimatis et al. (2004) and enable an agent to maintain a consistent world view in the absence of direct input. Representing collisions between objects allows agents to evaluate their actions [Kunze et al. (2011b)], determine their own appearance [Bongard et al. (2006)] or interact with objects of interest [Buche et al. (2010)].

2.2.2 Behavioural Aspects

The second use case of mental simulation is represented by anticipating behaviours of more complex, autonomous entities. This class of entities is comprised of systems (both artificial and natural) that exhibit some form of non-determinism, free will or high behaviour complexity. Characteristics of environmental mental simulation are inherited and extended in anticipating complex entities and if the assumption is made that these entities are also able to make inferences on others, several levels of anticipation arise. For example, in the scenario where John knows that Mary falsely thinks that Joe has a toy car, John has a two-level mental image of his friends which allows him to know that Joe does not own a toy car (level 1) but Mary thinks he does (level 2).

Three types of behavioural mental simulation can be distinguished based on which entity is considered for such reasoning, namely entities which have a high degree of resemblance (Subsection 2.2.2.1), those which do not (Subsection 2.2.2.2) and one's own self (Subsection 2.2.2.3).

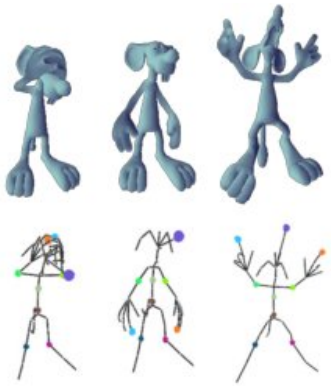
2.2.2.1 Agents with Similar Structure

The context of anticipating and learning from the behaviour of other entities which are similar in structure is considered a special case due to the possibility of using one's own internal structure to achieve such inference. In its simple form, one can consider for example two identical robots (with identical internal states) that when put in exactly the same conditions will behave in the same manner.

One of the requirements for understanding similar others is recognizing what they intend to do. The challenge in this context is to anticipate goals and actions based on the fact that the other entity is similar or identical in structure and reasoning mechanisms.

Humans are not only able to reproduce behaviour [Rochat and Passos-Ferreira (2009)] but can also understand the underlying goals of another's actions [Rao et al. (2004)]. Experiments showed that intention recognition can be achieved from both successful and failed attempts to perform an action, given the performer was a human and not a mechanical device [Meltzoff (1995); Meltzoff et al. (2002)].

To mentally simulate the actions of other entities that closely resemble themselves, agents in existing approaches use their own mechanisms to infer useful information about chosen targets (examples in Figure 2.1). Body mapping techniques are used, generally based on simplified skeleton structures [Buchsbaum et al. (2005)], to create the gestural information link between agents. Once mapped, motion patterns are matched against the agent's inventory of actions in order to infer intentions [Gray and Breazeal (2005); Breazeal et al. (2009)].



Max T Mouse described in Buchsbaum et al. (2005), using a skeleton to map movements of other mouse on its own.

Leonardo from Gray and Breazeal (2005), using its own body model to infer human intentions.

Figure 2.1: Excerpts from works using mental simulation in understanding similar agents.

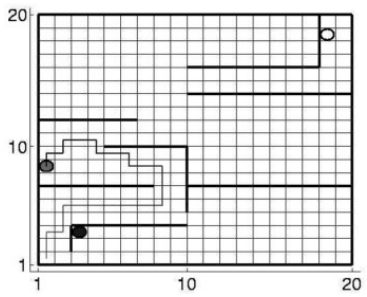
Due to the fact that mental simulation targets have **similar** behaviour mechanisms and embodiments, the need for additional specialised models for others' behaviour is avoided. Unfortunately, when differences increase, this technique leads to the occurrence of the **correspondence problem** [Dautenhahn and Nehaniv (2002); Alissandrakis et al. (2002)]. Moreover, current implementations are limited to a **predefined** inventory of actions, based on a specific model of self and others, restricting them from more **flexible** behaviour.

Subsequent to identifying the intentions of a similar other, the challenge is to extend one's knowledge by learning from observed actions. Learning complexity in this context is still relatively reduced because, in this case, novelty is expressed as different utilizations of the same underlying mechanisms.

In a close relation with understanding intention, humans are able to learn novel methods of achieving the intended goal. The phenomenon of imitation in humans and animals is covered in a wide range of research, clustered into

two main contexts by Rizzolatti (2005): the capacity to replicate an observed action and to learn new behaviour through observation.

Depending on their implementation, current approaches have mechanisms to extend their knowledge base by observing other agents or human-experts performing actions (examples in Figure 2.2). An example of such mechanism are behaviour trees, branches of which can be extended with new nodes [Buchsbaum et al. (2005)]. Prototypic fuzzy cognitive maps have been employed to learn by imitation without modifying the structure of the prototypes themselves [Buche et al. (2002, 2010)]. Software images [Costa and Botelho (2013)] have been proposed as a framework for enabling agents to identify similar others and to learn by imitation.



Imitation in a maze environment from Rao et al. (2004).

Virtual agent in Buchsbaum et al. (2005) can learn new behaviour from its peer.

Figure 2.2: Excerpts from works performing imitation learning with the aid of mental simulation.

The use of one's own mechanisms to reason about others provides a fast and convenient way to **learn** new behaviour. However, the occurrence of the correspondence problem still imposes **limitations** on agents. Knowledge **extension** is not performed when the internal structure does not correspond between agents.

2.2.2.2 Agents with Different Structure

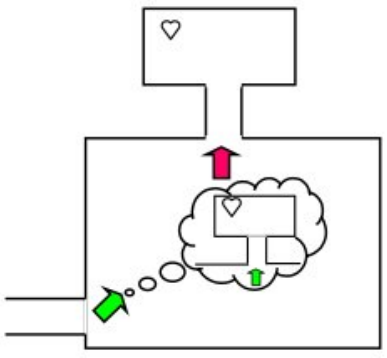
The general case of understanding other entities, involving anticipation and learning, requires the capability to formulate theories about how structurally-different others behave. Using one's own mechanisms for this purpose is fundamentally limited and therefore the challenge is to obtain a generic method of representing others.

The general challenge in the interaction with a heterogeneous agent environment is to anticipate complex entities which may be dissimilar to oneself and to predict what they believe and how they will act. In this case, predic-

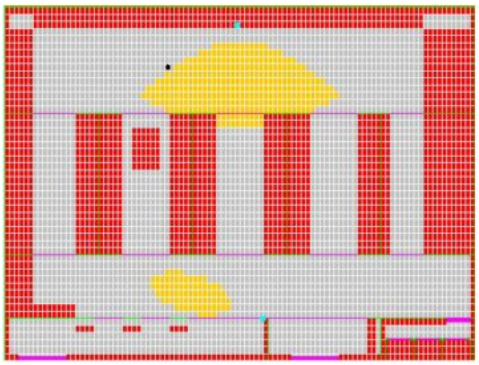
tion of environmental changes should also be taken into account, as behaviour generally depends on changes in the agent's surroundings.

Whether implicit or explicit, a certain level of anticipation is required by any autonomous entity for it to function in its environment. The phenomenon of anticipation has been studied in a variety of domains such as biology, neuroscience, cognitive science and artificial intelligence [Poli (2010)].

The predominant approach in existing implementations is represented by the use of custom theories or models of other entities (examples in Figure 2.3). Agents place themselves "in the shoes of others" and mentally simulate what their intentions and beliefs are [Ustun and Smith (2008)] based on the information they have available.



Quakebot in Laird (2001) which can simulate what other players will do.



Mental simulation of intruder from Ustun and Smith (2008), within a physical security scenario.

Figure 2.3: Excerpts from works performing mental simulation based behaviour prediction on agents with different structure.

Existing approaches use **simplified** models to predict essential aspects of the behaviour of other entities. Some agents use feedback to **improve** their models, in order to more accurately anticipate behaviour. The drawback of current methods is that a general-purpose approach for anticipation has not been proposed. Models are tailored by domain-experts for **specific** scenarios, which fail when faced with novel contexts.

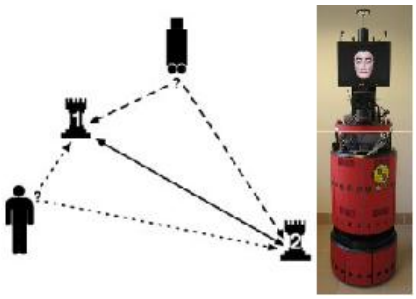
Understanding other entities leads to the challenge of being able to collaborate and form teams. Anticipating team members becomes important towards the achievement of common goals. As with other forms of complex anticipation, this context includes both environmental and behavioural simulation.

Human interaction within a collaborative context relies on several aspects of social cognition such as role-taking and empathy. Being able to understand others, thereby developing a theory of mind, enriches the cognitive ability of

individuals to perform social interaction Meltzoff (2002).

By adding models of humans in a collaborative context, existing approaches (examples in Figure 2.4) make simulations to determine the intentions of other members to improve team performance [Kennedy et al. (2008)], by adopting others' point of view [Roy et al. (2004)]. Research using simulation techniques in this context is focused on collaboration within human teams [Kennedy et al. (2009)], while the agents have the role of helpful companions. There exist, however, implementations that allow agent-to-agent collaboration [Hanratty et al. (2007)].

Robot in Roy et al. (2004) takes human's perspective to communicate.



Team robot from Kennedy et al. (2009) uses cognitive model to anticipate human decisions.

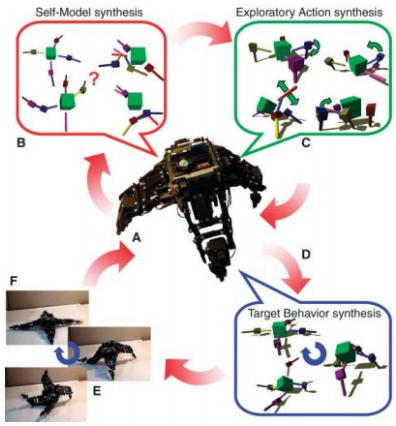
Figure 2.4: Excerpts from works using mental simulation for collaboration with humans or other robots.

Team-oriented approaches can make **decisions** based on simulations of others' behaviour. They are capable of taking the **perspective** of other team members in order to understand requests and act accordingly, but this is currently done using **expert-tailored** cognitive models of the team-mates (either human or artificial). These implementations function in relatively **simplified** scenarios, and require further configuration in case of scenario changes. Moreover, the agent's **emotional system**, seen as responses to an examination of self, others and environment, is not approached.

2.2.2.3 The Self

Once capable of simulating the evolution of the environment and other entities, the challenge is to achieve introspection and include the self into these simulations. This context requires a functional model of self and information about possible interactions with the surroundings.

Recent studies suggest that common mechanisms are responsible for the accounts of theory of mind and awareness of self mental states or "theory of *own* mind" [Frith and Happé (1999); Happé (2003); Williams (2010)].



Robot described in Bongard et al. (2006) that uses continuous self-modelling to recover from damage.

Approach from Kunze et al. (2011b) using a self model to simulate actions.

Figure 2.5: Excerpts from works performing mental simulation of the agent's own self.

Existing implementations approach this case using either preassigned physical selves or by creating their own models based on interactions with the environment (examples in Figure 2.5). Physical-engine based approaches enable agents to emulate themselves carrying out an action [Kunze et al. (2011b)] and leverage mental simulation results to make decisions in reality. Such mental simulations allow agents to evaluate strategies in advance and choose the optimal course of action [Buche et al. (2010)]. This can be done without a predefined model of the agent, by using primitives to automatically construct simplistic models of themselves, using sensory data [Bongard et al. (2006)].

Virtual physics models help agents to **interact** with the environment. Updating the self model during interaction makes such systems more **robust** to changes such as unexpected damage. By automatically creating the models, **expert intervention** is minimised. Even though virtual selves may conflict with reality, **errors** can be used to improve the current model. However, a trade-off occurs between model **simplicity and accuracy** as faster simple models may lack details while slower complex ones lead to erroneous mental simulation. Currently, self models do not have **adaptable** levels of detail and approaches are generally limited to the physical self. Moreover, few approaches consider the need for a detailed representation of the **self**.

2.3 Cognitive Function

From the point of view of cognitive science, we investigate **why** existing approaches use mental simulation. Research [Buckner and Carroll (2007)] suggests that mental simulation can be regarded as a common mechanism – *i.e.* overlapping areas of brain activity – in the brain for remembering events, performing prospection (Subsection 2.3.1), navigation (Subsection 2.3.2) and accounting for a Theory of Mind (Subsection 2.3.3). More recent studies [Van Hoeck et al. (2012)] also connect counterfactual reasoning (Subsection 2.3.4) – thinking about *"what could have been"* – to related brain regions used by the previously mentioned cognitive functions. Each of the four cognitive functions discussed in this section is illustrated in Figure 2.6.

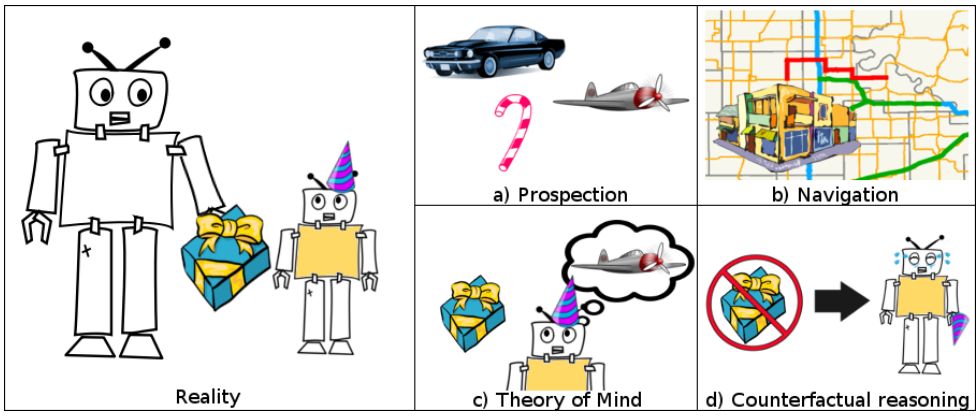


Figure 2.6: Birthday present example: in order to arrive in the position to hand the gift, the agent was required to (a) reason on which type of gift would be most suitable, (b) find which way through town would be best to go to the store, (c) have an understanding of what the other wishes for a gift and (d) imagine that not bringing a present would have had undesirable results.

2.3.1 Prospection

Prospection is a form of self-projection into the future, through which one anticipates future events to find actions which are most favourable towards achieving a certain goal. Great emphasis is placed on the role of mental simulation on this ability in cognitive science research [Taylor et al. (1998)], while computational approaches have recently begun to make use of the paradigm in artificial intelligence systems [Kunze et al. (2011a); Buche and De Loor (2013)].

Planning has been one of the first research directions since the advent of artificial intelligence (for an in-depth review see [Wilkins (2014)]). With the development of novel Monte Carlo Tree Search (MCTS) variants (for survey

see [Browne et al. (2012)]), efficient algorithms have been proposed for more complex environment conditions such as real time strategy games [Churchill and Buro (2013)]. Existing approaches using MCTS show promise in providing an efficient way to make decisions in complex scenarios, but all such approaches **rely on the assumption that a simulator is provided** which can compute future states of the environment on demand. For simpler scenarios like games, this simulator can be developed as the game mechanics are known and accessible, but generally this is not the case for agents that interact with the real world, where such a simulator does not exist.

In this sense, mental simulation is a candidate mechanism for enabling the prospection ability required by general purpose planners. Existing approaches make use of the mental simulation paradigm to make **predictions of the behaviour** of other entities [Buche et al. (2010)] in order to **reach their goals**, to **evaluate their own actions** [Kunze et al. (2011b)] and also as a medium for estimating **which actions** can provide more information about an unknown state of the agent's embodiment in the world [Bongard et al. (2006)].

2.3.2 Navigation

The ability of humans to recall places for navigation purposes differs from map-like artificial systems in that it also elicits contextual information based on previous visits of the location in question. Intuitively, it is not difficult to imagine why prospection would be beneficial to our ability as humans to navigate efficiently: when performing navigation, memories [Byrne et al. (2007)] and emotional information associated to certain places that we visited in the past can make us prefer or avoid them. If our memory had not provided us with a location related context, we would not be able to perform goal-oriented navigation.

Navigation itself does not refer only to large scale movement. Within the meaning of navigation, we identify **multiple nuances** that depend primarily on scale and the individual who is performing it. Using **mental simulation**, an agent can move its arm towards an object using different perspectives than its own [Roy et al. (2004)]. Similarly, one can take the **perspective of another** to infer a navigation plan [Laird (2001); Ustun and Smith (2008); Kennedy et al. (2008, 2009)]. **Larger scale** self **navigation** – *i.e.* where the agent itself moves from one location to another – can also be **evaluated** [Buche et al. (2010)] and **anticipated** [Svensson et al. (2009)] through mental simulation based prospection.

2.3.3 Theory of Mind

The capability of an individual to assign mental states to oneself and others is known in the literature as Theory of Mind (ToM) [Premack and Woodruff

(1978)]. Functionally, possessing a model of others' decision process which takes observations as input enables the individual to determine goals, anticipate actions and infer the causes of certain observed behaviours. The two predominant approaches to how the decision process is represented are the Theory-Theory [Carruthers (1996)], which relies on a "folk psychology" that is used to reason about others in a detached way, and the Simulation Theory [Goldman (2005)] which claims that the individual's own decision mechanism is used for inference (simulation), using pretend input based on observations.

Simulation Theory has received enthusiasm from artificial intelligence researchers, arguably because it provides an interesting and **computationally feasible mechanism** for reasoning about other individuals. The "like me" approach [Meltzoff (2007)] has been adopted in various **social agents** that interact with others in virtual [Buchsbaum et al. (2005)] and real [Gray and Breazeal (2005); Breazeal et al. (2009)] environments. It has also been used in **team scenarios** to enable a robot to take the **predicted behaviour** of its team mates into account to **improve its own plan** [Kennedy et al. (2008, 2009)]. Theory Theory based approaches are less common, but show that "objectively" reasoning on the behaviour of others can lead to comparable results [Laird (2001); Ustun and Smith (2008)].

2.3.4 Counterfactual Reasoning

Whether expressing the regret that things could have been better if only actions were taken, feeling relieved that the worst scenario did not happen or simply imagining what would have happened if past events were different, humans often think in a counterfactual manner Roese (1997). Counterfactual reasoning has been thoroughly documented in psychology [Roese and Olson (1995)] throughout the development of children and adults. This type of inference has been studied within the more general problem of causation [Pearl (2009)].

Interestingly, works that implement forms of counterfactual reasoning into autonomous agents are **scarce**. Examples include making use of this mechanism for **minimizing regret** in games with incomplete information [Risk and Szafron (2010)]. Mental simulation has also been discussed as a mechanism for counterfactual reasoning [Markman et al. (1993)]. However, computational approaches to inference about **what could have been**, via mental simulation, are **limited**, focusing on relatively simple cases of object continuity [Cassimatis et al. (2004)] under the form of integrating new knowledge into past events.

2.4 Computational Implementation

Based on **how** computational mental simulation has been approached in literature, we identify three main perspectives on modelling the paradigm.

The first type of approach is given by traditional **symbolic systems** which model the process of mental simulation through sets of **rules and logical inference**. This category is characterised by a **coarse** level of **granularity in prediction**, as the rules they use are abstracted away from low-level information such as the detailed geometry of the environment, collisions or location of obstacles. The **majority** of existing computational approaches to mental simulation are, at least to some extent, symbolic systems. Some of the most relevant examples of such agents [Laird (2001); Cassimatis et al. (2004); Kennedy et al. (2008, 2009)] are constructed on top of well known **cognitive architectures** such as ACT-R [Anderson (1983)] and Soar [Laird et al. (1987)]. Focus is placed on goal recognition through mentally simulating others' actions using the agent's own reasoning model [Buchsbaum et al. (2005); Gray and Breazeal (2005); Breazeal et al. (2009)] and on **evaluating the outcomes** of actions [Kunze et al. (2011a)]. Using a symbolic approach, an agent would be able to perform mental simulation on **high level** information like beliefs and annotated actions, but would require other, low-level tasks to be modelled separately and abstracted so that it can be used as a black box.

The second category consists in using **sub-symbolic** techniques to make predictions in the form of mental simulations. Granularity of the mental simulations is, in this case, **finer** as trajectories are continuously taken into account [Buche et al. (2010)]. Likewise, sensory information can also be predicted [Svensson et al. (2009)] which leads to a plausible approach to **implicit anticipation**. However, using only low-level controllers **narrows the use of mental simulation to specific targets**.

Finally, the category of systems based on **analogical representations** [Shepard and Cooper (1986); Kosslyn (1980)] consists in approaches which model their environment as an **internal virtual world** [Buche and De Loor (2013)]. This allows agents to change perspective [Roy et al. (2004)], **anticipate behaviour** [Ustun and Smith (2008)] and **generate scenarios** that help them obtain useful information about the world and themselves [Bongard et al. (2006)]. Using analogical representations [Rumelhart and Norman (1983)] provides a natural approach on mental simulation, similar to mental imagery in humans [Grezes and Decety (2001)]. Nonetheless, **existing computational approaches use specific models** to perform this type of mental simulation.

2.5 Discussion

Within this chapter, we examined related works in various contexts of using mental simulation for decision-making by anticipating events in the environment and behaviours of entities that populate it. The study was divided into four main sections, each reviewing the state of the art through the prism of **where**, for **what**, **why** and **how** mental simulation was used. Hence, relevant approaches from literature have been discussed and analysed, to identify their strong and feeble characteristics, within the context of the type of environment they aim for, which phenomena they are able to predict, which cognitive functions they achieve and finally, the techniques used for their implementation. The approaches that explicitly use mental simulation are compiled in Table 2.1 for a complete view of the level to which they address the aspects of this paradigm as suggested by the cognitive science community. Additionally, we investigate the way in which they achieved their functionality and also whether or not their authors provided testable proof-of-concepts or open source code.

Taken together, existing approaches cover all areas of interest in using mental simulation as an anticipation and decision-making technique, however there exists no implementation that addresses all of them on its own. Neither does any approach propose a **generic** way that makes it extensible over all examined features simultaneously.

The majority of implementations have begun in virtual reality, but fewer have taken the leap to robotic embodiments due to dependencies on specific information about their environment. Those that do however pass into the real world are either limited in the actions they can perform or rely heavily on repositories of actions that are abstracted for the use within a higher-level framework.

Focus is placed on solving only a subset of the cognitive functions associated with mental simulation, and this is done using specific models of the task at hand. Hence, not many elements are taken into account into mental simulation, for example anticipating trajectories but not collisions, or focusing on only one of the environmental and behavioural aspects of the environment, although they are generally interdependent. One of the cognitive abilities linked with the paradigm of mental simulation – counterfactual reasoning – has been scarcely approached, and only in a relatively simplistic fashion.

We have also discussed three trends in implementing computational mental simulation, namely those using symbolic, sub-symbolic and analogical representations. These approaches vary in mental simulation granularity, *i.e.* the space and time scale to which an agent can perform a mental simulation of its

[1]Virtual world categories in Table 2.1: **D**iscrete / **C**ontinuous, **2D**imensional / **3D**imensional

[2]Partial source code, such as providing only the main architecture sources but not the full implementation of the approach, is marked with $\sim$ in Table 2.1

	Computational use of mental simulation																
	Where?		(for) **What?** Env.			Beh.			**Why?**				**How?**			**Open?**	
	Virtual world **D**/**C**, 2D/3D [1]	Real world	Trajectories	Collisions	Occlusions	Similar	Different	Self	Prospection	Navigation	Theory of Mind	Counterfactuals	Symbolic	Sub-symbolic	Analogical	Source code or demo available [2]	
Laird (2001)	C_{3D}						✓	✓		✓	✓		✓			∼	
Roy et al. (2004)		✓	✓	✓	✓					✓					✓		
Cassimatis et al. (2004)		✓	✓		✓					✓		✓	✓				
Buchsbaum et al. (2005)	C_{3D}					✓		✓			✓		✓				
Gray and Breazeal (2005)	C_{3D}	✓	✓		✓	✓		✓			✓		✓				
Bongard et al. (2006)		✓		✓				✓	✓						✓	∼	
Ustun and Smith (2008)	D_{3D}		✓				✓			✓	✓				✓		
Kennedy et al. (2008)	C_{2D}	✓	✓				✓	✓		✓	✓		✓			∼	
Breazeal et al. (2009)	C_{3D}	✓	✓		✓	✓		✓			✓		✓				
Kennedy et al. (2009)	C_{2D}	✓	✓				✓	✓		✓	✓		✓			∼	
Svensson et al. (2009)	C_{2D}		✓							✓				✓			
Buche et al. (2010)	C_{2D}		✓			✓		✓	✓	✓				✓		✓	
Kunze et al. (2011b)	C_{3D}		✓	✓					✓				✓			✓	
Buche and De Loor (2013)	C_{3D}		✓	✓					✓					✓		✓	
Churchill and Buro (2013)	C_{2D}		✓						✓						✓	✓	

Table 2.1: Overview of areas of interest covered by relevant existing approaches.

environment. In essence, coarse granularity leads to faster computation of abstract knowledge, while fine granularity can cope with highly detailed models of reality. The advantages and drawbacks of these approaches are illustrated in Table 2.2.

Approach	**Pros**	**Cons**
"Like-me"	◊ Mechanism for goal inference	♦ Correspondence problem ♦ Specific
Different (unknown)	◊ Independent of other's structure ◊ No assumptions	♦ More difficult to learn
Symbolic	◊ High-level inference ◊ Direct rules	♦ Actions abstracted away ♦ Can miss details
Sub-symbolic	◊ Low-level control ◊ Precise movement	♦ Specific controllers ♦ Difficult to model interactions
Analogical	◊ Intrinsic ability to generate scenarios ◊ Multi-scale interaction	♦ Construction difficulty

Table 2.2: Pros and Cons of approaches to computational mental simulation.

Furthermore, as Table 2.2 also shows, the two major ways of addressing

anticipating the actions of other presumably intentional entities in the environment – "Like-me" and Different/unknown structure – stem from the two dominant theories of how Theory of Mind is performed by humans: Simulation Theory and Theory Theory, respectively. In effect, this led to implementations that are constructed to resemble humans (or other entities they may interact with) and use these models in mental simulation, while the others attempt to create abstract models of others' behaviour for prediction.

Overall, specific implementations are often preferred due to convenience. However, this leads to limited functionality when the environment and the behaviour of simulated agents change significantly. One example of technique which is only used in few implementations is automatic model creation. This enables the agent to infer its own body structure, but it is not currently used for other physical entities. Another example is represented by production rule systems which have relatively high expressive power, but are not used in complex scenarios due to the difficulty of problem formalization. Connectionist approaches also exhibit high performance in creating efficient controllers, but learning rate drastically decreases with the size of the network and are therefore only used for specific tasks.

Finally, we investigated the availability of functional software support for the computational approaches to mental simulation. Few researchers have published runnable demonstrations of their work (including videos, test cases and open source code). For some works, only the cognitive architecture used is provided, but not the actual extension of the architecture to the mental simulation paradigm.

To conclude the study of existing works that use the simulation paradigm for anticipation and decision-making, several shortcomings which motivate the objectives of this thesis are identified and listed in the following:

- Current approaches are constrained to function in relatively specific setups, with few exceptions which do not provide access to the details of their implementation. Such exceptions that do aim to be more **generic** rely heavily on models created by experts which tend to be difficult to obtain for contexts with higher degree of complexity.

- Existing implementations are generally used in controlled scenarios and are not designed to be fully **autonomous**.

- Although approaches exist that function in real time, **online** learning is used only for specific tasks. The mechanism of the simulation paradigm, such as continuous imagination-reality comparison in complex environments and imaginative behaviour, are not yet fully exploited.

- Due to either context simplicity or specific functionality, **believability** is only achieved to a relatively low level.

In the following chapter, we introduce and discuss our approach to computational mental simulation that uses an analogical representation system but with the possibility of heterogeneously modelling various entities in the environment within a multi-agent imaginary world. We thereby aim to obtain a generic agent architecture that enables mental simulation in real time continuous environments with dynamic interactions between entities.

Contents of Chapter 3

3 Proposition: ORPHEUS **29**

3.1 Generic Agent Architecture 30
3.1.1 Perception 32
3.1.2 Imaginary World 32
3.1.2.1 Mental Images 33
3.1.2.2 Mental Simulation 34
3.1.2.3 Interweaving Effects 36
3.1.2.4 Decisions and Control 38
3.1.2.5 Imagination 40
3.1.3 Abstract World 41
3.1.3.1 Structure 41
3.1.3.2 Model Training and Evaluation 43
3.2 Implementation 44
3.2.1 Software Architecture 44
3.2.2 Distributed Processing and Visualization 55
3.3 Discussion 57

Chapter 3

Proposition: ORPHEUS

Summary of Chapter 3

*This chapter introduces our contribution: a generic mental simulation agent architecture – ORPHEUS – which provides an answer to the shortcomings identified in the current state of the art. Herein, we provide a detailed conceptual description of our architecture, followed by its implementation in the `C++` programming language. The conceptual description is structured into three main aspects: perception, imaginary world for mental simulation and an abstract world for learning. It explains how the agent observes the environment and constructs its inner imaginary analogue of the world which functions based on a variety of abstract models. We also discuss how our architecture can enable an agent to perform mental simulation in virtual and real environments with the purpose of achieving the discussed cognitive functions and managing both environmental and behavioural anticipation simultaneously. The implementation section then delves into greater detail of how the conceptual framework is integrated within the software architecture, resulting in a **generic** approach to building **autonomous** agents capable of **online** learning and **believable** behaviour.*

3.1 Generic Agent Architecture

Our contribution is an agent architecture that provides a generic approach to the paradigm of mental simulation, as support for decision-making. According to this paradigm, an individual owns an imaginary space which is built by sensing reality and functions on its own to predict what outcomes are to be expected in reality in various situations. From the computational standpoint, we consider the "real world" as the environment from which our system takes its input and upon which it acts. In the general sense, the "real world" could be the physical reality. However, the "real world" can also be modelled as a virtual environment in which entities can behave autonomously and perceive each other. In addition, we refer to an "imaginary world" as a completely separate virtual space, which is proprietary to the agent(s) using the architecture, where mental simulations and system decision making are performed. Finally, for the imaginary world to function, the agent requires knowledge and the ability to acquire it from observing how itself and other entities interact within the environment, hence an "abstract world".

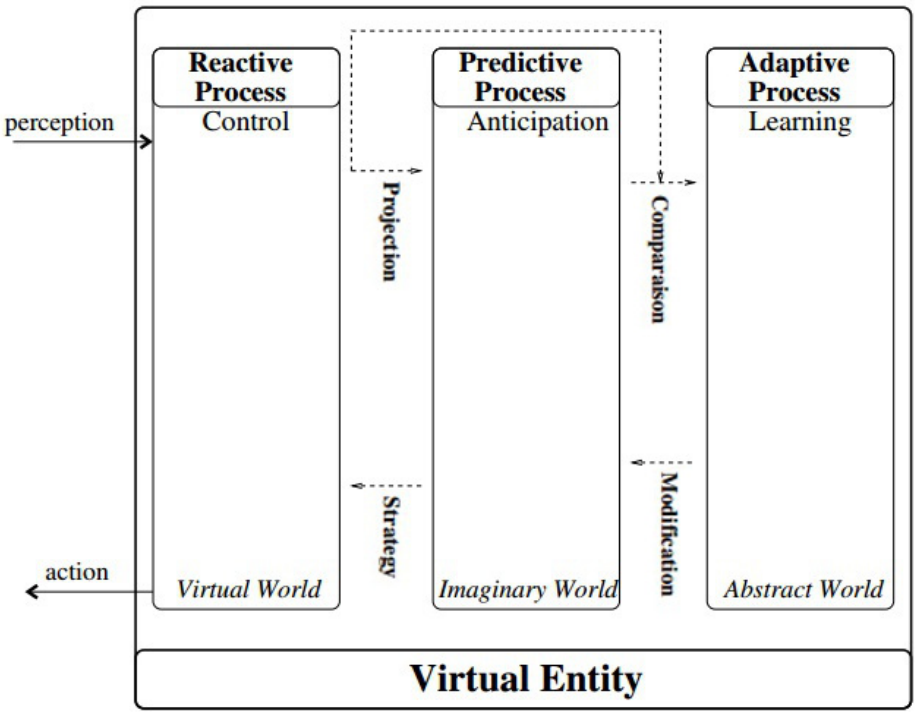


Figure 3.1: Conceptual overview of the "simulation within the simulation" framework [Buche et al. (2010); Buche and De Loor (2013)].

This approach was first suggested by Buche et al. (2010); Buche and De Loor (2013) as a novel means of creating adaptive virtual agents (Fig. 3.1).

However, the suggested conceptual framework does not describe how "simulation within the simulation" should be constructed, nor provides a generalization of this approach to multiple scenarios.

The agent architecture proposed in this chapter – ORPHEUS: *Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation* – provides a description of mechanisms that allow the application of the mental simulation paradigm into agents in various contexts.

Taking inspiration from cognitive science literature and building upon the three conceptual layers suggested before, we constructed a functional overview of the mental simulation paradigm (Fig. 3.2). The functionality of the architecture is illustrated with the present time as reference, stemming into the past and future. We note that as time passes, the same process can be applied: continuous perception, learning from and interpreting the past, predicting the future and acting based on predictions.

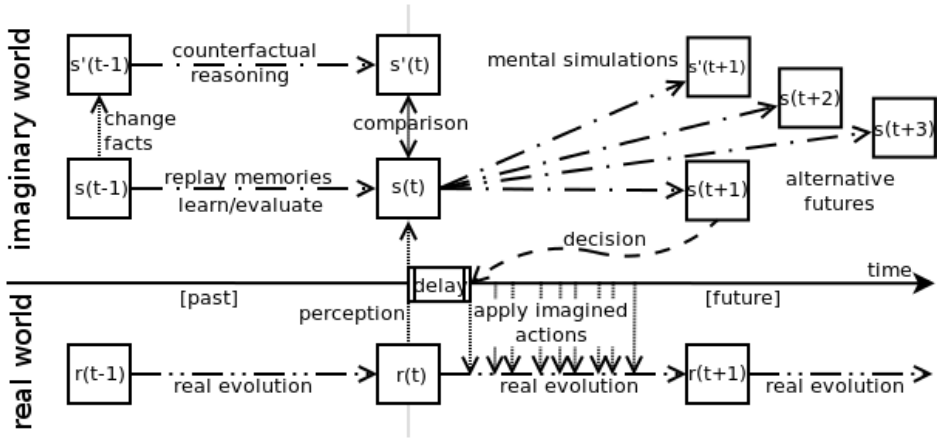


Figure 3.2: Functionality of our architecture at an arbitrary time t: the state of the real environment $r(t)$ is transformed into a mental image $s(t+1)$ through perception, from which alternative futures ($s(t+n)$, $s'(t+n)$) can be obtained, past events $s(t-m)$ can be reenacted or changed to produce "what if" scenarios (s'). Decisions can be based on evaluating alternative futures.

The following subsections break down the architecture functionality into its main components: perception (Subsection 3.1.1), mental simulation (Subsections 3.1.2.1 - 3.1.2.3), decision-making (Subsection 3.1.2.4), imagination (Subsection 3.1.2.5) and learning (Subsection 3.1.3). Here, we introduce concepts that are used in the architecture and explain how an agent would utilise it. Thereafter, we focus on the structure and functionality of imaginary world, after which we describe the importance of the abstract world for an agent's capability to adapt.

3.1.1 Perception

First, by definition, an agent must perceive its environment to be able to perform any meaningful actions within it. Research in computer vision – the field focusing on how machines process and interpret visual information from sensors – provides an extensive range of algorithms for object detection and recognition, depth perception and scene reconstruction.

Our approach is based on the classic discretisation of the real environment evolution into time-dependent states; *i.e.* $r(t)$ is the state (value) of the real world evolution process (function) r at an arbitrary time t (Fig. 3.3).

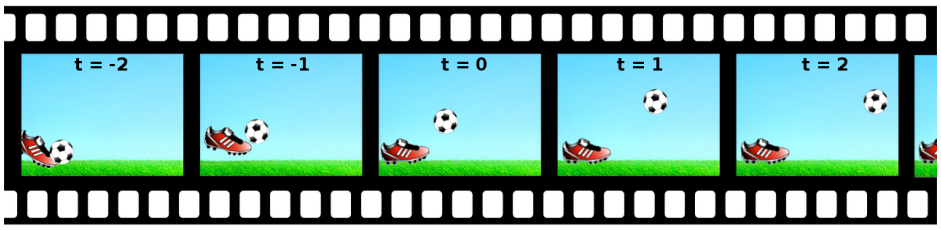


Figure 3.3: Visual perception of the real world evolution – $r(t)$ – as a series of timestamped images.

The next step is to create an analogous imaginary world by mirroring the real world evolution. We start the construction at the present time, let it be t, with the agent's perception of the real world which results in an representation $s(t)$ which we refer to as a "mental image". Images are of course static, they only represent the configuration of the environment at a given moment in time. The aim is to obtain novel configurations from acquired ones, which can represent what would happen at a future time, or what could have happened if the past were different. Therefore, a process that transforms these mental images is required: "mental simulation".

3.1.2 Imaginary World

Our approach to constructing the imaginary layer of the architecture is to use mental images as the building blocks of mental simulation. Because mental images are originally obtained through perception as previously mentioned, using them to describe the state of the imaginary world leads to direct translation between reality and imagination.

In the following, we elaborate on how mental images can be constructed, and how they can be transformed through mental simulations for different purposes within the proposed architecture.

3.1.2.1 Mental Images

In humans, representations of reality seem to be built from sensory and perceptual information (Moulton and Kosslyn (2009)) such as shapes, colours, textures, sounds or movement patterns, which form aggregates which we abstractly consider as objects. If we follow this intuition, then the inner imaginary world of an individual could be described as a collage of real world data. Additionally, to account for behaviour, this collage has the ability to evolve into other configurations that represent other states of reality. Hence, one way of constructing an imaginary world for our agent is to use a type of representation that is easily translatable from perceptual input and which can later be evolved into different states.

We define a mental image as a configuration that statically describes the state of the imaginary world at an arbitrary time t. The structure of this element determines the representation capabilities of the agent architecture. For example, on the one hand, traditional logic-based systems are proficient at manipulating discrete elements which have robust rules that govern them but on the other hand are limited in representing raw data from sensors. In contrast, connectionist approaches such as neural networks perform well in noisy environments using direct sensory input, but are not meant to be able to construct and reason on conceptual knowledge. Therefore, for our approach to be generic, we must consider a variety of possible representations of reality.

One of the oldest such representations is obtained by considering that a world state can be described as a collection of entities, each of which has a set of properties (Fig. 3.4). This approach can be used for a large subset of real phenomena, especially for describing their physical state. Hence, we could assume that the world, at a given moment, is composed of a list of entities such as rocks, birds, ants, people or droplets of water, each described by a given set of properties such as position, rotation, shape, linear and angular velocity.

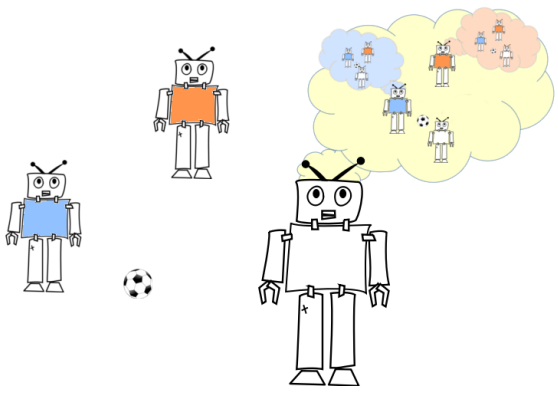


Figure 3.4: Belief representation using entities with properties approach.

This may appear to be a superficial world view, because it only describes directly observable information. However, the approach can be extended to represent hidden information by creating strictly imaginary analogues for them. For example, imaginary properties such as taste or smell can be attributed to entities even if they do not exist as separate elements in reality. The beliefs of entities can also be represented by mental images that describe what they perceive about the world around them.

Another approach could be to replace rigidly delimited entities with probabilistic structures (Battaglia et al. (2013)). While decreasing the accuracy of inference results in perfect perception scenarios, probabilistic representations can be more robust to perception errors where more deterministic approaches fail to give accurate predictions. Other representations could include lower level structures such as raw sensor data maps, however the high complexity may not be scalable to higher level reasoning.

In this thesis we focus on rigid bodies for the physical aspect of prediction, however the architecture could support different structures as mental images. Henceforth, we describe the second component of the imaginary world of our agent, which enables the evolution of mental images into different configurations, as a function of time.

3.1.2.2 Mental Simulation

Analogous to how the real world evolution is a process that changes the environment state, we consider mental simulation as a process that computes the evolution of a mental image into a future one. Two factors that influence the real future $r(t+1)$ can be identified, from the system's perspective: prediction errors and system-generated actions.

Inaccurate prediction can be caused by faulty knowledge acquisition due to noisy sensor data for instance, or by the intrinsic stochastic nature of the real world, leading the agent to observe a discrepancy between its imagination and the world state. Stochastic processes govern the real world and have received significant attention from the research community. The inability (or impossibility) to observe the exact state of the environment prompts the need to consider more than one possible outcome of an event (Fig. 3.5).

The agent's own actions can also influence the future state of the environment by interacting with other entities. Performing an action may change the state of the environment, therefore leading to a different outcome than that which would have occurred if the action had not been fulfilled. This type of alternative future generation can be considered controllable as it is internally decided by the agent (Fig. 3.6).

Hence, for the agent to account for these expected discrepancies between reality and imagination, the symmetry between the two worlds has to be broken by allowing multiple alternative futures. We consider this functionality in our architecture by branching and altering the evolution of mental simulations

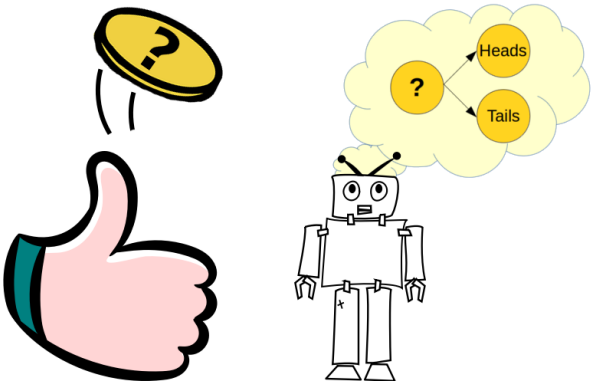


Figure 3.5: Uncontrollable alternative futures.

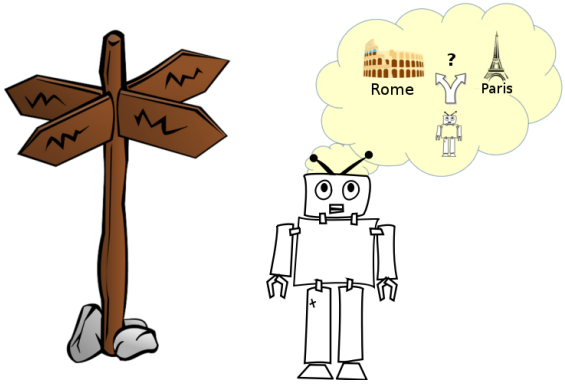


Figure 3.6: Controllable alternative futures.

resulting in sets of mental images that correspond to arbitrary future times. For the agent to be capable of decision making, given that the number of possibly conceivable futures is virtually infinite, it must be able to control its mental simulations, as discussed in Subsection 3.1.2.4, to avoid intractability of the search through possible outcomes.

Analogous to how the future could be obtained from the present, so could the present be obtained from the past. It is conceivable that the agent can apply the same process to a past event to obtain subsequent but known states. Because now the "future" state is available (already observed), the agent can perform learning in order to improve the overall accuracy of its mental simulations, as discussed in Section 3.1.3: the abstract world.

In addition to evaluating the future, controlling mental simulations allows the agent to arbitrarily construct imaginary pasts from which potentially important information can be extracted, known as "what if" scenarios (Subsection 3.1.2.5). Such alternative series of events could also be placed outside of real time (*i.e.* not requiring an real moment of time as an anchor, but still

depending on a virtual time scale to evolve), which brings the functionality closer to the human ability of imagination.

An important aspect of mental simulation, from the computational point of view, is how effects from different sources can be interwoven within the same framework. Physical reality and behaviour of living things is clearly interrelated but have traditionally been treated separately due to the different laws that govern them. For example, it comes natural to describe a wooden stick as having a position, material properties and possibly a trajectory if in free fall or thrown. An animal using it to retrieve unreachable food (Seed and Byrne (2010)) may however be described as a cognitive system that has beliefs, desires and intentions and can act upon its environment for its own benefit. Although differently modelled, these entities do interact, and it is plausible to represent and reason about them within the same system. In the following we discuss our approach to integrating different mental models within the same framework of mental simulation.

3.1.2.3 Interweaving Effects

Based upon the premises that a single monolithic world model that could explain the entire environment evolution is, to say the least, difficult to achieve, we adopt a hybrid approach for knowledge representation in our agent architecture. Herein, we consider that the behaviours of separate entities may be best approximated via different models. This approach however requires a means of interweaving the effects of several evolutions into a single framework.

Central to our approach is the process of merging effects from different sources (physics, behaviour) into the evolution of the imaginary world as a whole. In order to combine the effects of multiple entities on the environment, each entity is assigned mental models, from an available pool, that dictate the way it behaves. The model assignment for each entity can be based on the estimated error between mental simulation and reality, using an error measurement of choice. Within our architecture, a model can be viewed as a function that takes a state of the world (the assigned entity and its perceptions) as input, and computes a time-dependent change in this state (details in Section 3.1.3). In this sense, the proposed imaginary world falls into the category of multi-agent systems. The process of merging consists in applying a pool of models to their assigned entities in a mental image and accumulating their effects to obtain the subsequent mental image.

A key aspect in the effect merging process in our approach is its granularity. Traditionally, abstract rules are used to describe the evolution of an environment, such as for example: “If [Pencil is not held] Then [new location of Pencil is on Floor]”. This type of environment evolution description does not include what happens to the pencil while it falls (obstacles in trajectory), why it falls (gravity), nor what the effects of such process would be, unless specified through a complex array of other rules. In contrast, our approach is

to simulate the falling pencil, and therefore naturally represent its interactions with other entities such as the table that may be between the pencil and the floor, or the cat lying on the floor which will run away after the pencil nudges it. In consequence, it is important for this simulation to occur in small time steps, as it is not feasible to combine large effects in the context of complex interactions between entities and their environment. In addition, operations that are not commutative and require an order, such as rotations, can be forced to behave in a nearly commutative way by using small time steps.

We integrate granularity into mental simulations by constructing them from a series of successive steps (Fig. 3.7). This way, the mental simulation is divided into n steps where, at the i^{th} step, each mental model (M^j where j is the model identifier) takes the i^{th} mental image as input and computes a small change (δ^j) that is accumulated with the rest of the effects to obtain the $(i+1)^{th}$ mental image.

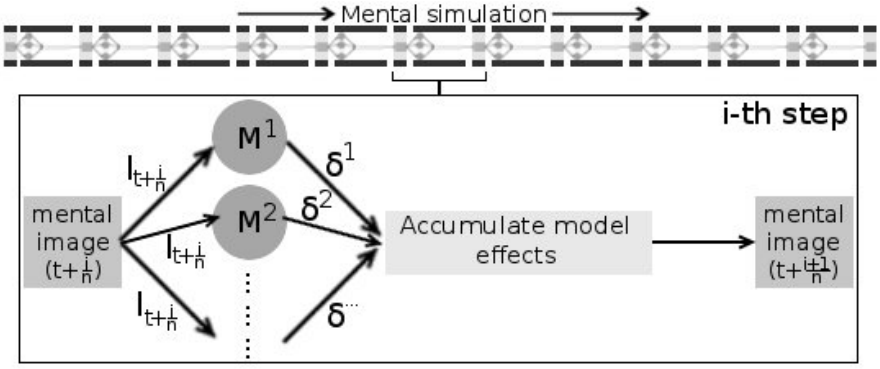


Figure 3.7: The mechanism through which a mental image is evolved in one step ($i \in [0, n-1]$), by interweaving effects from models.

We note however that fine granularity is not a constraint in our architecture, but rather a capability to account for interaction effect merging. There are cases in which using large steps to perform a mental simulation may be more efficient, especially when a well known event is reasoned up for which the effects had been learned in advance. For example, with the flick of a light switch, it is not efficient to mentally simulate the internal mechanism of the switch, the electrons that flow through the circuit and the internal design of the light bulb, but it is enough to take a much faster route directly from switching on to the lighting of the bulb. However, it is desirable that the agent can take the longer path as well, in the case where, for example, the light does not turn on as expected after flicking the switch. In this exceptional case, the agent could refine the granularity of its mental simulations to evaluate whether the malfunction is caused by the switch, the circuit or the light bulb itself.

The step-based approach to constructing mental simulations also allows

their evolution to be controlled by integrating different courses of actions for entities and for the agent's mental self, as follows.

3.1.2.4 Decisions and Control

From the perspective of the imaginary world layer, we view mental models as black boxes that contribute to the evolution of mental simulations. In order to control the course of a mental simulation, we consider that these black boxes offer an interface that can change their behaviour. Specifically, the mental models that are assigned to entities can take parameters that control how they generate an entity's behaviour during a mental simulation, feature which we refer to in this work as an "action interface" (Fig. 3.8).

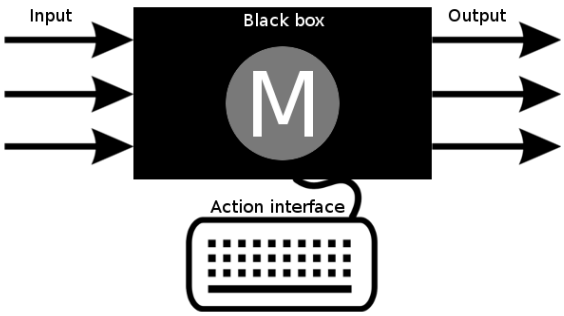


Figure 3.8: Mental model seen as a black box, controllable through an action interface.

The possibility of having an action interface on a mental model gives a generic way to apply decisions into mental simulations. This is a fundamental feature of the agent's self model, which can be controlled directly by system, as previously illustrated in Figure 3.6 (Controllable alternative futures). Additionally, other directly non-controllable entities can be mentally controlled to some extent in order to evaluate various possible turns of events, as previously mentioned in Figure 3.5 (Uncontrollable alternative futures).

An important aspect of utilizing this design is that, for simplicity and computational efficiency, the interface should be shared between the imaginary world and reality. Let us suppose that this would not be the case and the commands issued for the mental models are different than those sent to the motor system of the agent, then a translation would be necessary so that imagined actions are transformed into motor commands and vice versa. Therefore, it is computationally more efficient for the commands issued to the mental and real selves to coincide. This constraint also offers another advantage: in the context in which the imaginary world is populated and animated with information from reality, the common command constraint also facilitates learning the effects of one's actions. The mechanism behind this feature is that when

the agent issues a motor command, let it be C, it can learn the effects it produces in reality in direct association with C. Commands need not be discrete, but can be represented by values in the continuous domain. As such, other model-specific parameters can also be specified, if required.

Conceptually, the action interface consists in a link between reality and the imaginary world. However, for it to function, it requires a control module that issues commands through it. In our approach, we refer to these elements as "simulation controllers". Controllers may also directly contribute to the agent's decision-making process by issuing commands to the imaginary representation of self, commands which can then be applied to reality via the motor system, or they can simulate the decision process of other entities within the imaginary world. Simulation controllers are part of the abstract world of our architecture, and will be detailed in Subsection 3.1.3.

Decision-making depends on goals; in our approach, goals are attached to mental simulations and are evaluated at each step. We define goals as conditions that apply to one or more mental images, so that the imaginary environment configuration is evaluated. For instance, considering a scenario where the agent has to position itself in a certain location, the goal can be defined as the condition that the imaginary self representation finds itself into the given area, case in which the evaluation of the mental simulation's goal satisfaction would give a higher value. Goal satisfaction levels can have continuous values, so that partial success can be represented. In effect, the agent can be given a desired situation/configuration as a goal, for which it must find a way to arrive in. Within the architecture, goals can be provided by a human by specifying the constraints on the environment's configuration, but we also envisage the possibility of system-generated goals based on long-term learning.

Mental simulations and actions that are performed inside them also embed the notion of the corresponding real time, so that synchronization can be done correctly when the agent performs the actions in the real environment. When a mental simulation is created and run, its results will be available after a time delay which is dependant on the complexity of the process being computed. This way, to make a decision, the agent can perform one or a set of parallel mental simulations, wait for them to finish computing and then select a course of action which best suits its goals. However, we identify an important issue with this approach: with the assumption that mental simulations will be subject to error (which is generally the case in complex environment situations), this discrete manner of "thinking and then acting" is bound to fail if the period of time is sufficiently large that errors accumulate beyond the point where decisions remain valid. To this end, our approach allows the agent to continuously perform mental simulations and constantly update the set of actions it is about to perform.

The main challenge of maintaining the validity of mental simulations resides in guaranteeing that the system's real actions are not different from those

which are being simulated, during the time it takes to compute the prediction. The time interval that corresponds to mental simulation processing time must be fixed inside the prediction, because otherwise actions simulated in that interval would not be executable before the mental simulation result is available, and therefore action synchronization would be broken.

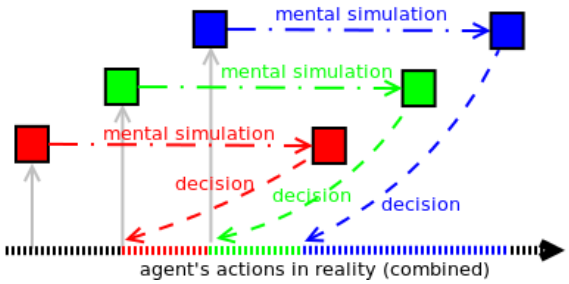


Figure 3.9: Action synthesis from multiple mental simulation results.

Figure 3.9 illustrates the process of continuously updating the agent's course of actions using successive mental simulations. We resolve this challenge by including the currently planned actions into the new mental simulations, so that they maintain their validity regarding the previously mention issue. The process consists in evaluating a sequence of overlapping mental simulations to construct the future course of actions that the agent will take, based on the expected goal satisfaction. This way the action plan, in analogy to figure 3.9, would first consist in the decision from the *red* mental simulation; while the agent is performing these actions, the *green* mental simulation is being computed and as soon as the result is available, the decision can be included in the future action plan. This process can be repeated continuously, and allows the agent to maintain a more robust imagination-reality decision correspondence, while also allowing it to integrate novel, unpredicted information into the decision-making process and therefore increase its adaptability.

3.1.2.5 Imagination

Humans are capable of altering reality in their minds, towards various purposes. Children engage in pretend play, while adults often root alternative pasts into reality to generate outcomes that might have been more desirable. Altering the past to obtain "what might have been" is referred to as *counterfactual reasoning*. From a functional point of view, this ability is used by humans for various purposes (Byrne (2002)) such as to improve their future performance (*"if I had studied more, I would have received better scores"*) or to manage guilt attribution (*"if only I knew about the traffic jam on street X, I would have taken another route to work"*). However, such alternative scenarios need not be fixed in time and space like counterfactuals, but can be

purely a result of one's imagination, where real and/or fictional elements can be assembled in novel ways.

Within our architecture, the mental simulation process can also be applied to past mental images, obtaining results in the past or present, or can be a timeless assembly of elements that evolve according to previously learned rules. This way, the agent has the ability to mimic the creative reasoning process of humans within its own imaginary world.

Using this paradigm, when confronted with the question *"why did you perform action A ?"*, the agent would have the mechanisms to provide answers such as *"because if I had not, event E would have occurred which does not satisfy my goal"* or *"because actions B and C would have led to undesirable outcomes"*. This type of answers resembles how humans generate explanations for their actions, and has the potential to increase the agent's level of believability.

3.1.3 Abstract World

For mental simulations in our approach to function, the agent requires a means of representing the processes that give the evolution of the environment and the entities that populate it. In the following we discuss the details of how our architecture represents this type of knowledge, how the agent can acquire new knowledge and how available mental models can be evaluated.

3.1.3.1 Structure

Humans can make decisions without a complete understanding of how everything works; instead, our brains represent partial information which is necessary to perform tasks. Moreover, people can extrapolate certain knowledge to diverse scenarios and recombine the effects of each element in the environment to obtain novel knowledge of previously unseen outcomes. Our agent architecture aims to achieve this flexibility by providing a means of representing knowledge of how entities behave through a heterogeneous pool of mental models. This way, multiple behavioural aspects of an entity can be modelled separately, through possibly different mechanisms, which are then brought together within the mental simulation. As previously described, the effects of each mental model are thus interweaved (Subsection 3.1.2.3), producing larger scale outcomes. This approach is inspired from how systems following simple rules can exhibit emergent behaviour, through simulation.

In our architecture, each mental model follows a common template: it considers an aspect of the real world as a function that takes the environment configuration at time t as input, performs changes on it, and provides a new configuration as output which corresponds to time $t+\delta t$. These configurations are proprietary to the agent, and are represented as mental images in its imaginary world. For simple behaviours, such as physical phenomena, where

a given state depends only on the previous environment configuration, the model can function only with one mental image as input. In the case of more complex behaviours, such as other autonomous entities, models must contain internal states in order to correctly predict their behaviour, and therefore the input consists in a series of mental images, so that its internal states are synchronised with the simulated context.

From a technical point of view, the purpose of a mental model is to reenact and predict a time series. The evolution of a given entity's behaviour or sub-behaviour is considered as a time-varying signal which is to be reconstructed in the agent's imaginary world.

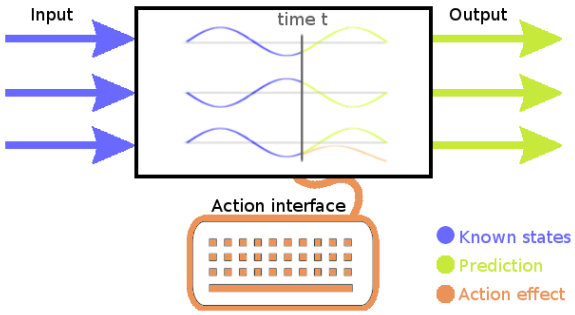


Figure 3.10: Functionality expected from a mental model within our agent architecture.

Figure 3.10 illustrates the functionality that a mental model should provide in order for the mental simulation process to function. Within a mental simulation, at each step, a mental model will receive the current state of the imaginary world (a mental image) as input. The model can build its own internal state based on one or more input images which constitute the "known states". Based on its internal state, if any, the mental model produces a prediction that represents the state which corresponds to the next time step in the mental simulation, given a step size. Optionally, in the case where the entity to which the model is assigned represents the agent's self or is in any way controllable by the agent, the mental model's action interface can be used to change its behaviour.

An important aspect of mental simulations is that they are self-contained; *i.e.* they evolve separately from the real environment and each other. As previously described, the models in a mental simulation are given the current imaginary state as input but, because of this self-containment, the current state is in fact a prediction made at the previous step. Therefore, mental models must be able to process their own output throughout the evolution of the imaginary scenario, without exhibiting divergence.

In the following, we describe how learning can be performed within our architecture and the system can evaluate the accuracy of its mental models.

3.1.3.2 Model Training and Evaluation

Online learning is a critical requirement for an agent that must act in an autonomous fashion and adapt to changes in its environment. The learning process must be continuous to improve the accuracy of short term predictions which prove vital for humans and animals alike. However, the agent should also be able to learn about processes that exhibit patterns over a wider time length. Long term learning allows the agent to optimise its overall behaviour to meet its goals, and enables it to perform predictions over longer periods of time such as planning.

Our agent architecture allows mental models to learn continuously from the agent's perception data. Learning is performed in a way similar to how mental simulations function: at each perception step, two environment configurations (mental images) are paired, representing the time dependent transition (initial state and expected output) to be learned by a mental model. At this level, the purpose of the models is to describe reality as accurately as possible, with the aim of predicting its evolution. This leads to a supervised learning context, where the sequence of mental image pairs constitute the sample dataset on which a mental model is to be trained using a wide range of contemporary algorithms.

Such as with mental simulations, the granularity of learning can be varied to obtain models that learn short and long term patterns in the data. By allowing multiple levels of abstraction to be learned for a given process, the agent can first perform mental simulations with lower granularity which can be computed faster but at lower resolution and then elaborate on details using high granularity mental models.

The abstract world is separated into a parallel learning loop which asynchronously receives input from the perception layer and can provide the latest available mental models to the imaginary world on demand, to perform mental simulations. Based on the assumptions that (1) certain learning algorithms perform better than others in given situations and (2) the evolution of the environment can be broken down into a set of interdependent processes, training is organised into learning sessions, each containing a set of models that learn together. First, attempting to learn the same process using different algorithms within separate learning sessions allows the agent to prefer one set of algorithms over another, depending on how accurate the model is on the given learning target. In addition to exploiting performance niches, sessions allow integrating the effects of each model into the training; for example, given two mental models, one that describes the laws of physics and another that describes the behaviour of a bird, the latter can be trained using the data resulted from subtracting the effects of physics (gravity, inertia, friction). This approach can allow the agent to use mental simulation to extrapolate the effect of a given behaviour to hypothetical scenarios such as, for example, the impossibility for the bird to fly in the absence of air friction or how it may

behave in zero gravity, without explicitly witnessing these effects.

Accuracy measurement is important for choosing mental models that best describe the process to be predicted. During training, at each step, prediction error can be computed locally using the difference between a model's output and the expected configuration. Based on the local error of each mental model in a session, an overall accuracy of a training session can be computed which describes how well the model set predicts a given process for one step. In addition to per-step error measurement, the training session is to be evaluated in isolation over multiple time steps, to compute the confidence that the mental simulation can be accurate enough for at least a given period of time. This way, locally accurate models can be used for short predictions, while more robust models can be used in longer term mental simulations.

3.2 Implementation

ORPHEUS was implemented in `C++` with multi-thread support. The `C++` programming language was chosen to allow the agent to achieve high performance in dynamic real-time scenarios, while also providing a straightforward way of implementing the conceptual framework through object-oriented programming techniques.

Memory management becomes critical for an agent that performs multiple parallel mental simulations, which require separate resources for each execution thread. We simplify this aspect in ORPHEUS by using the shared pointer implementation, along with multi-threading and other tools from the Boost[1] library.

Representing entities in a mental image requires storing their positions, rotations, velocities and other values that describe its state. These data are to be processed within mental simulations to replicate their real world evolution. As this process represents a central part of our system, computation speed is critical and therefore we make use of the efficient matrix implementation from the Eigen[2] template library.

3.2.1 Software Architecture

The functionality of the proposed agent architecture is organised into a series of interconnected packages. Its core consists in the ability to perform and manage mental simulations - the `SimulationEngine`. The remaining packages serve as encapsulated functionalities on which the core depends (Fig. 3.11) to represent (`Metamodel`), compute (`Models`), learn (`ModelTraining`) and distribute (`Communication`) the evolution of the environment, allowing the agent to reach a desirable situation (`GoalSystem`).

[1] http://www.boost.org/
[2] http://eigen.tuxfamily.org/

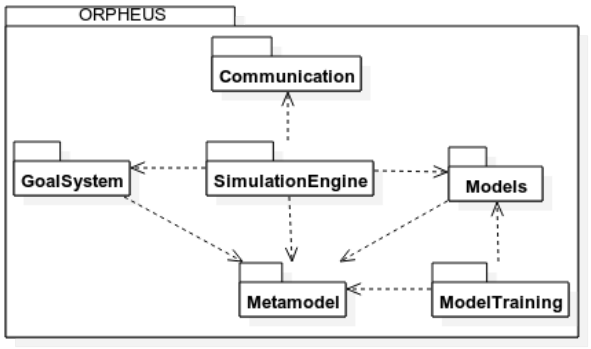


Figure 3.11: Packages of the ORPHEUS agent architecture and their interdependency.

The main components of the `SimulationEngine` package consist in the *Simulation* class with its network-related derivatives and the *SimulationControllers* that control the *Models* within a given mental simulation (Fig. 3.12).

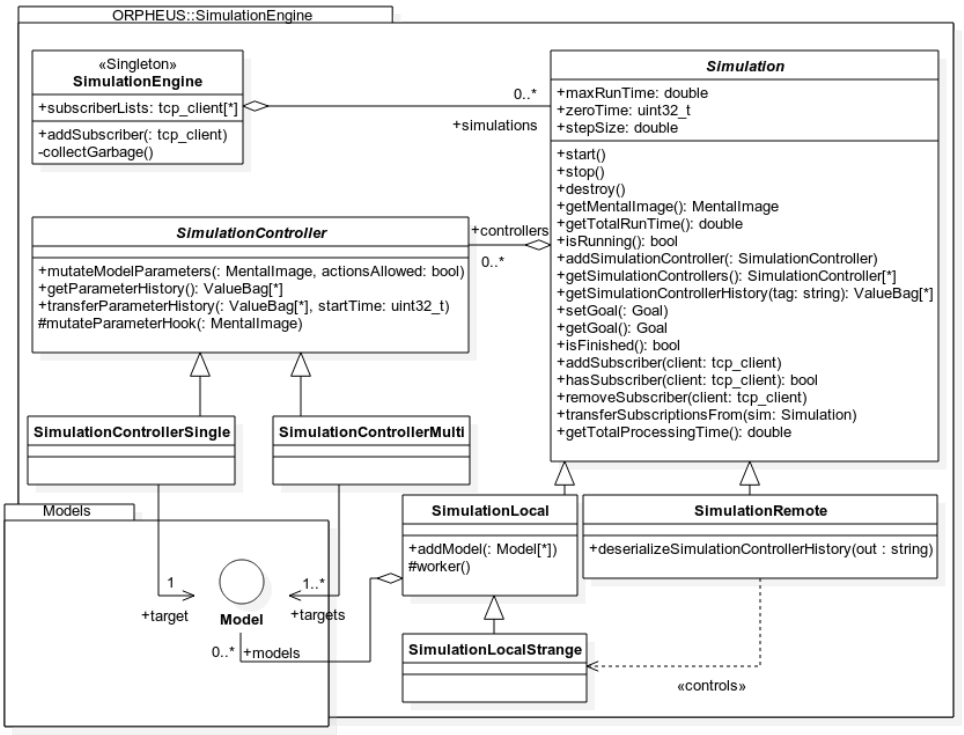


Figure 3.12: Class diagram - principal components of the `SimulationEngine` package.

The *SimulationEngine* class, which shares its name with the package, plays a management role for all mental simulations performed by the agent. This management consists in easy access to any of the mental simulations, either running or finished, and a parallel garbage collection system for simulations with heavy resource use. It allows mental simulations to be accessed via network, using the `Communication` package, by external Graphical User Interfaces (GUI) that enable the visualization (Subsection 3.2.2) of the agent's decision making process.

The paradigm of mental simulation is implemented into the *Simulation* class, which provides a set of operations to control their use. Mental simulations each contain a Boost thread object and once started, the entire simulation process is run on a separate thread until a goal has been met or the maximum simulation time has expired. Mental simulations can asynchronously provide information about their state (running or finished), final or intermediary mental images that reflect the current evolution state of the world being emulated, and can also feed this information to remote subscribers.

Each mental simulation executes Algorithm 1 within a separate thread. The initial mental image can represent the current state of the real environment, or an arbitrary imaginary situation. This image is sequentially updated by applying the assigned mental models which perform the changes that contribute to its evolution.

Algorithm 1 Mental simulation

Prerequisites: Initial mental image (I), Mental model list
if not models initialised **then**
  for each *model* **as** M_i **do**
    $M_i.initialise(I)$
while $maxRunTime - totalRunTime < \varepsilon$ **do**
  $I_{orig} \leftarrow I$
  for each model **as** M_i **do**
    $M_i.step(stepSize, I_{orig}, I)$ /* I is modified */
  if $passedTime > subscriberFeedInterval$ **then**
    $feedImageToSubscribers(I)$
  for each *simulationcontroller* **as** C_i **do**
    $C_i.mutateModelParameters(I, actionsAllowed())$
  if *hasgoal* **then**
    $goal.update(I)$

If controllers are assigned to any of the entities within the mental simulation, the algorithm verifies whether the virtual time exceeds the total estimated time required to compute the simulation. This is important because the imaginary actions which are performed before the mental simulation would finish computing cannot be applied to the real environment, and therefore renders the simulation results invalid. Hence, the influence of simulation con-

trollers must be postponed until the simulation virtual time corresponds to a moment after the real processing has finished, using the *actionsAllowed()* function illustrated in Algorithm 2.

Algorithm 2 Estimating computation time

Function *actionsAllowed*()
    $estimatedTotalRuntime \leftarrow \frac{maxRunTime * timeSinceStart}{totalRunTime}$
    if $virtualTimeSinceStart > estimatedTotalRuntime$ **then**
        return *true*
    else
        return *false*

Mental simulations produce the evolution of imaginary scenarios to be evaluated according to system goals, but can also provide the history of actions taken by the controllers it contains. Subsequent mental simulations will result in interleaved action sequences to perform by the agent, as previously illustrated in Figure 3.9. The `SimulationEngine` package of ORPHEUS provides the means of managing the results of mental simulations through the *ActionSynthesizer* class (Fig. 3.13).

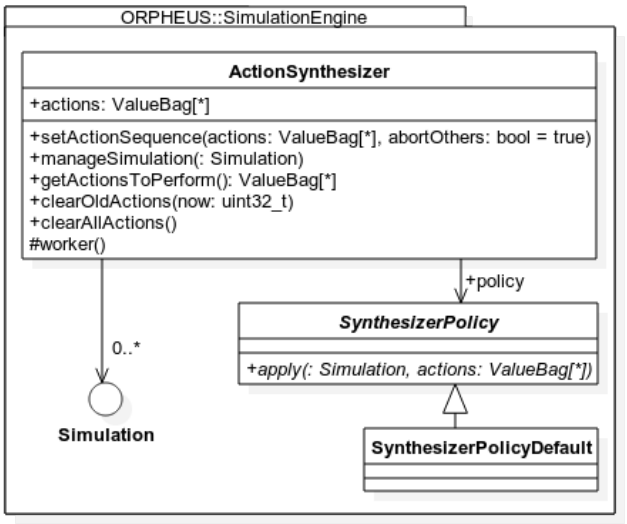


Figure 3.13: Class diagram - action synthesis in the `SimulationEngine` package.

Action synthesis consists in the management of a set of mental simulations with the aim to collect their results for constructing the agent's subsequent actions based on the highest goal satisfaction level. The *ActionSynthesizer* will constantly verify which mental simulations have finished processing and for those who have, it will search for a simulation that maximises the goal sat-

isfaction level. If such a mental simulation is found, it will apply the available policy. This will result in a positive response if the policy has been applied and negative otherwise. In the case where the policy is applied and hence the agent's subsequent actions are modified, concurrent mental simulations will become invalid due to the fact that they no longer take into account novel changes (Algorithm 3).

Algorithm 3 Action synthesis

Prerequisites: Mental simulation list (S), Policy (P), Agent actions (A)

```
while not interrupted do
    bestScore ← 0.0
    bestSimulation ← null
    for i ← 0...S.size() do
        if S_i.isRunning() and not S_i.isInvalid() then
            continue
        if not S_i.isInvalid() then
            if S_i.hasGoal() and S_i.getGoal().getSatisfaction() > bestScore
            then
                bestScore ← S_i.getGoal().getSatisfaction()
                bestSimulation ← S_i
        toRemove.add(S_i)
        S.remove(i)
        i ← i − 1
    actionsChanged ← false
    if exists bestSimulation then
        actionsChanged ← P.apply(bestSimulation, A)
    for each toRemove as r do
        SimulationEngine.removeSimulation(r)
    if actionsChanged then
        for i ← 0...S.size() do
            S_i.setInvalid(true)
```

The application of a policy to the best mental simulation consists in recovering the action history within that simulation and modifying the agent's current course of action using that history. The default policy (Algorithm 4) replaces the agent's actions beginning at the virtual time corresponding to the start of the mental simulation. Depending on context, additional policies may be added, which perform other operations on the agent's action plan, by deriving the *SynthesizerPolicy* class and overloading its *apply* method.

The real-imaginary loop is therefore closed: the agent perceives the environment, performs mental simulations and an action plan is synthesised and applied back to the real environment. The action plan is accessible via the *getActionsToPerform* method of the *ActionSynthesizer* class.

Algorithm 4 Applying the default *SynthesizerPolicy*

```
Function apply(simulation, agentActions)
    actionHistory ← simulation.getSimulationControllerHistory()
    if actionHistory is empty then
        return false
    satisfactionLevel ← simulation.getGoal().getSatisfaction()
    startTime ← actionHistory[0].getTimestamp()
    index ← 0
    while index < agentActions.size() and
            agentActions[index].getTimestamp() < startTime do
        index ← index + 1
    if index < agentActions.size() then
        if agentActions[index].getSatisfaction() < satisfactionLevel then
            agentActions.erase(index, end)
        else
            return false
    for i ← 0...agentActions.size() do
        copy ← actionHistory[i]
        agentActions.add(copy)
    return true
```

Additional to prediction, mental simulations also constitute a basis for evaluating imaginary scenarios. Whether a scenario is favourable can be measured through the prism of system goals (Fig. 3.14).

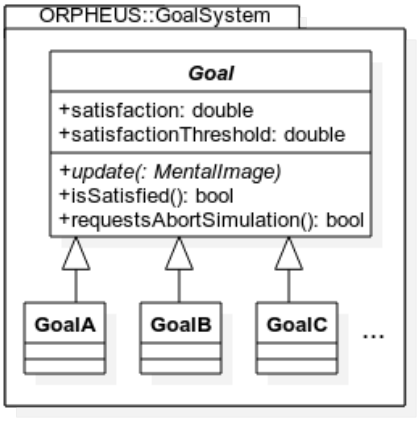


Figure 3.14: Class diagram - creating system goals.

If a goal is assigned to a mental simulation, the current state of the imaginary evolution of the environment is communicated to the goal system for evaluation. Depending on the agent's requirements, system goals can be added by

extending the Goal class and implementing its *update* method. The paradigm used for the goal system consists in the evaluation of one or more mental images and identifying whether a certain configuration is reached and to what extent.

In our approach, a mental image is a collection of entities, each characterised by a set of physical attributes such as shape, position, rotation and velocity (Fig. 3.15). Mental images are assigned a moment of time which corresponds to the beginning of the mental simulation to which they belong and a time difference that gives how far in virtual time the image is located. This relation to time is especially important in future action synchronization, as predictive mental simulations run ahead of real time.

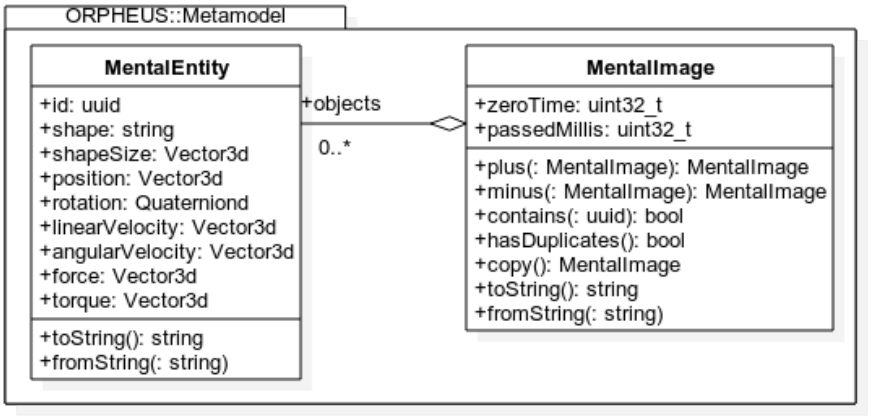


Figure 3.15: Class diagram - the `Metamodel` package: mental images and their structure.

To achieve effect interweaving in mental simulations, we implement two operations - addition and subtraction - in mental images. This allows mental images to be obtained from a sum of small effects brought by each participating mental model. Given a finite number of models $M_j, j = 1, 2, \ldots, m$, we compute each model's effect defined by $\delta_j := I_j^{out} - I^{in}$, *i.e.* the difference between the mental image outputted by M_j and the original input. Starting from input mental image I^{in}, corresponding to time t, the step result (time $t+1$) is computed by adding the effects of each model to the input mental image: $I^{out} := I^{in} + \delta_1 + \delta_2 + ... + \delta_m$. The order in which the effects of the models are applied to compute the resulting image would be neutral if the operations on mental images were commutative. Although combining rotations break commutativity, when the time step of the simulation is significantly small, the rotation given by the quaternion evolves slowly enough for the quaternion dot operator to behave in a nearly commutative fashion, because the associated rotations have nearly the same axis. We could define the linear approximation of the quaternion dot in order to impose commutativity, if needed.

ORPHEUS manages mental models through the *Model* abstract class (Fig. 3.16). The functionality provided by all models consists in initialization, learning, reenacting (step) and copying itself.

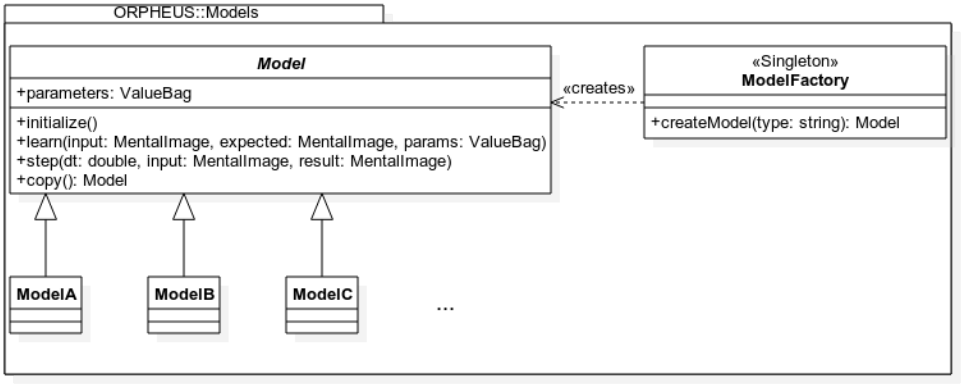


Figure 3.16: Class diagram - mental model abstraction and on-demand creation.

Upon initialization, the mental model object should allocate its necessary memory space; models are initialised at the beginning of each mental simulation. Learning consists in receiving a flux of mental images from the agent's perception (*input* mental image) and updating the model's inner structure in order to predict the target's evolution (*expected* mental image). During learning, the mental model also receives the actions which have been performed by the agent at the given time, so that the sensory-motor link can be learned, if the model provides this functionality. Applying the learned knowledge to the evolution of mental simulations is performed through the *step* method which is called at each simulation step. Given the time interval *dt* and the input mental image, the *step* method is responsible for computing the following state of the target and apply the change to the resulting mental image. Finally, models must implement the *copy* method which constructs an identical deep copy of the object (all its resources are duplicated), that can be used within another mental simulation (*i.e.* on separate execution threads, that evolve differently).

Mental model heterogeneity is a design principle of ORPHEUS. Hence, we consider them as separate from the main architecture, but allow them to be dynamically linked at runtime, so that loading them does not develop into a performance bottleneck. Dynamic library loading is implemented in the *createModel* method of the *ModelFactory* class (Listing 3.1), and uses the *libdl*. This way, each mental model can be implemented separately and compiled into a shared library which can then be loaded at runtime by an agent using the ORPHEUS architecture.

Listing 3.1: *ModelFactory* class, *createModel* method (C++ code).

```
boost::shared_ptr<Model>
ModelFactory::createModel(std::string type)
{
    // check if dynamic library was not loaded before
    if (_handleMap.find(type) == _handleMap.end())
    {
        //try to load dynamic library with this name
        std::string libName = "./lib" + type + ".so";

        //load library
        void *handle=dlopen(libName.c_str(),RTLD_LAZY);
        if(handle)
        {
            _handleMap[type] = handle;
        }
        else
        {
            std::cerr << "Failed to load lib"
                      << type << ".so ! Reason: "
                      << dlerror()
                      << std::endl;
        }
    }

    // check if dynamic library exists in memory
    if (_handleMap.find(type) != _handleMap.end())
    {
        // found handle ! create and deliver the object
        void *sym=dlsym(_handleMap[type],"createObject");
        if(sym)
        {
            Model *(*fnct)();
            fnct=reinterpret_cast<Model *(*)()>(sym);
            return boost::shared_ptr<Model>(fnct());
        }
        else
        {
            std::cerr << "Failed to create instance of "
                      << type
                      << " ! Reason: "
                      << dlerror()
                      << std::endl;
        }
    }

    return boost::shared_ptr<Model>(); //null
}
```

Dynamic library loading not only provides an elegant encapsulation solution for the core architecture and the mental model implementations, but also allows ORPHEUS to interface with a wider range of languages in which models can be written (Fig. 3.17). Moreover, agents that use the architecture can have their model repository updated seamlessly, without requiring changes to their global logic.

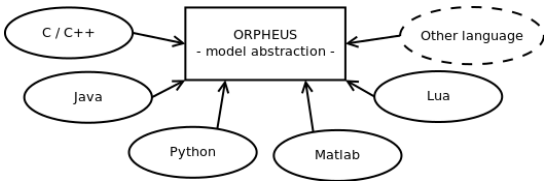


Figure 3.17: Mental model abstraction - integration of a heterogeneous (both structurally and functionally) collection of models at runtime.

In addition to seamless mental model replacement, ORPHEUS provides the tools for training and providing models as the agent evolves in its environment (Fig. 3.18). Perception data, in the form of mental images are fed into the ModelTrainer via its *addEvidence* method, where it is buffered for training. Learning takes place in parallel sessions in which sets of mental models are trained collectively.

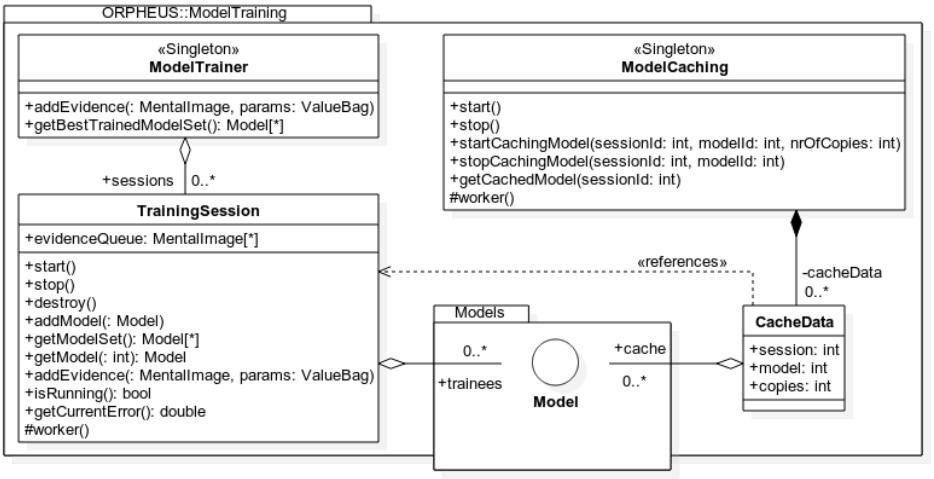


Figure 3.18: Class diagram - mental model training and caching.

This feature is based on the assumption that isolated models may not be able to reproduce all aspects of the evolution of the real environment, and therefore a collection of different models are used for learning each particularity of the data. This training method is consistent with the way mental

simulations are performed in ORPHEUS, where multiple models are applied at each step to produce the overall outcome of an imaginary scenario.

The procedure of a training session is illustrated by Algorithm 5. Given a list of mental models and perceptual data for training, a training session consists in a separate execution thread which iterates over the training data and selects pairs successive mental images which are given as input to each model's *learn* method. Mental model evaluation occurs at each step, and provides a measurement of error for selection purposes. The remaining effects to be learned are determined by performing a mental simulation step on each model and subtracting its effects from the expected configuration used in subsequent learning.

Algorithm 5 Training session

Prerequisites: Mental model list, Evidence data (mental images, agent actions)

```
if not models initialised then
    for each model as M_i do
        M_i.initialise(I)
while not interrupted do
    if evidenceQueue.size() > 1 then
        imgBegin ← evidenceQueue[0]
        imgEnd ← evidenceQueue[1]
        paramsBegin ← evidenceParamsQueue[0]
        evidenceQueue.remove(0)
        evidenceParamsQueue.remove(0)
    if has imgBegin and imgEnd then
        dt ← imgEnd.getZeroTime() − imgBegin.getZeroTime()
        modifiedImg ← imgBegin.copy()
        remainingImg ← imgEnd.copy()
        for each model as M_i do
            M_i.learn(imgBegin, imgEnd, paramsBegin)
            M_i.step(stepSize, imgBegin, modifiedImg)
            remainingImg ← imgBegin.plus(imgEnd.minus(modifiedImg))
```

The agent can make use of its models for mental simulations even though these models are still in training. This way, the best performing mental models can be requested asynchronously to the *ModelTrainer*, trading off accuracy for availability. Using mental models which are still in training requires making a copy of the model object and of its resources and may therefore be a slow process. To ameliorate this issue, parallel caching is used for making copies of heavily used mental models. By specifying the model, the training session it is participating in and the number of copies to be stored, the *ModelCaching* singleton will construct a pool of readily available cached copies, which can then be instantly accessed and used in mental simulations.

3.2.2 Distributed Processing and Visualization

In situations where searching the entire state space generated via mental simulation exceeds the computation capabilities of the agent, ORPHEUS provides the `Communication` package (Fig. 3.19) that can interconnect multiple agents together. The purpose of this approach is to enable sharing the mental simulation workload throughout a network of agents; *i.e.* an agent can make requests to other agents to help with computing one or several mental simulations for its own use.

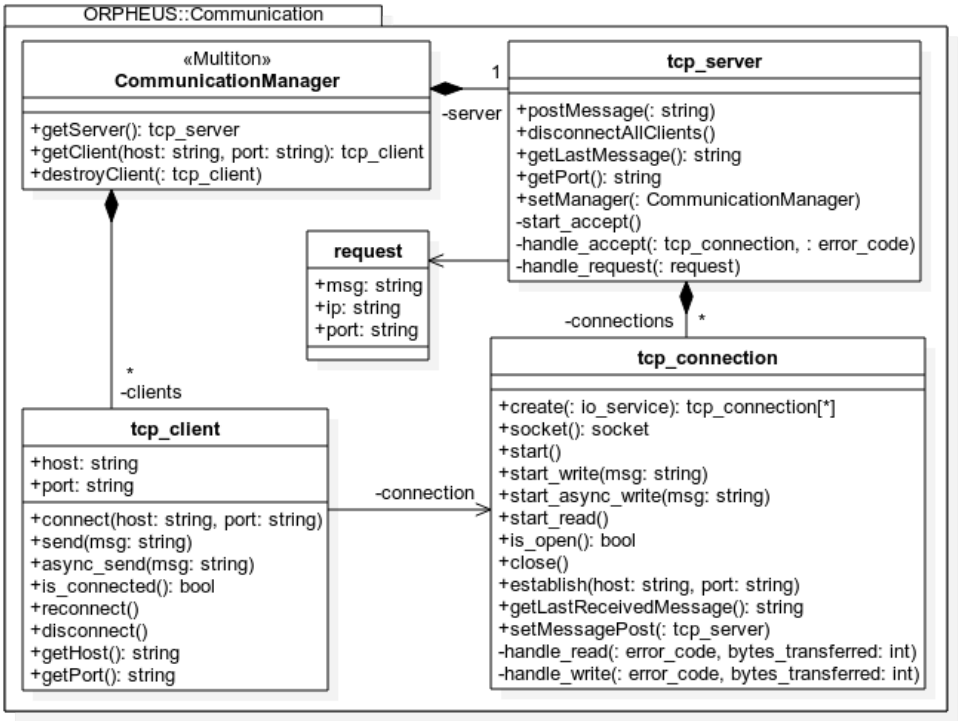


Figure 3.19: Class diagram - `Communication` package.

The *CommunicationManager* class is implemented as a multiton, *i.e.* static key-based access to its instances. The multiton implementation allows agents to manage multiple networks, as each server can be set to listen to a different network interface. Each *CommunicationManager* instance creates a TCP server that starts listening for connections on a free network port. The server asynchronously establishes and handles connections with the clients which connect to it, and decodes incoming requests. Additionally, the communication manager also provides the functionality to connect to other servers, by providing their IP address and port number. In this situation, the manager establishes and maintains the asynchronous communication channel until one of the participants is disconnected, or the connection is explicitly interrupted.

With this approach, two main network architectures are possible (Fig. 3.20): 1) a group of fully functional agents that share their computational resources, which can be used to those that require it most, and 2) one "master" agent that makes use of a group of "slave" instances of the framework which serve as a processing cluster.

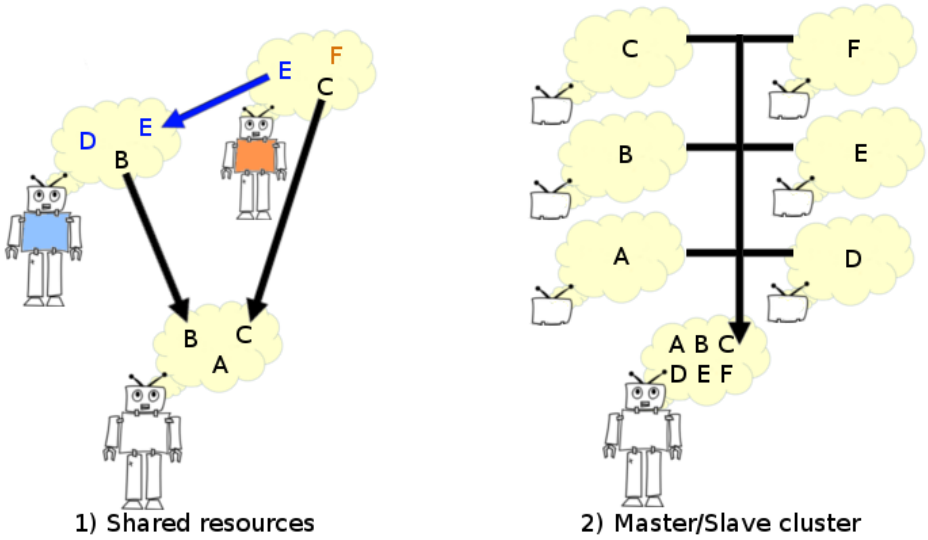


Figure 3.20: Resource management for mental simulations over a network of agents. Letters A-F signify simulated scenarios for which computation can be done by other agents in the network. Example: the orange agent computes scenario E for the blue agent.

By using these architectures, the agent is able to enhance its cognitive abilities by utilising the spare computational resources of other agents (either robots or regular computers). This is especially useful when an embodied agent finds itself in a critical situation that requires superior decision making capabilities which take into account multiple scenarios. In effect, the agent is able to receive aid from its peers and perform mental simulations in a distributed fashion.

In addition to computational distribution over the network, the *CommunicationManager* also allows broadcasting the state of mental simulations to subscribing Graphical User Interfaces (GUI). A GUI consists in a 3D visualization environment which is updated from information contained in mental images (Fig. 3.21). The GUI may use the *CommunicationManager* to connect as a client to an agent performing mental simulations, which in turn returns a stream of mental images from a chosen simulation that is in progress.

[3]Robocanes team (University of Miami, FL, USA), http://robocanes.cs.miami.edu/

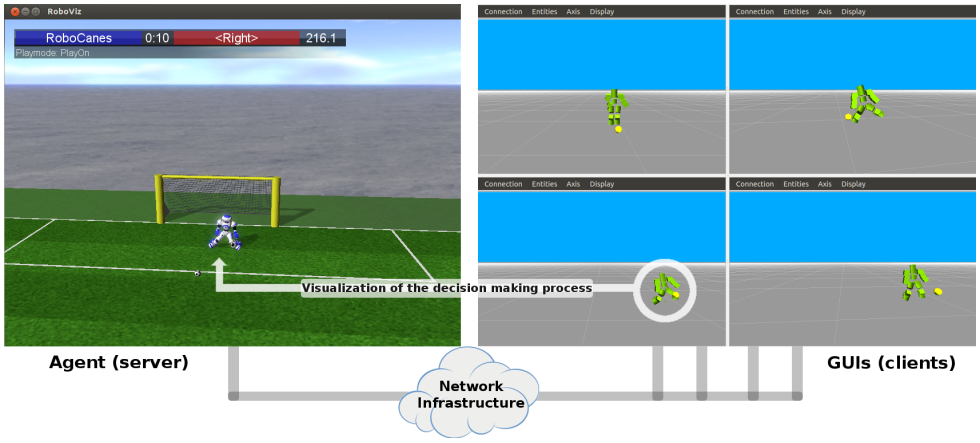


Figure 3.21: Network-based Graphical User Interface (GUI) for the agent's decision-making process. **Left:** ORPHEUS agent in virtual environment (Roboviz renderer by Stoecker and Visser (2012)); **Right:** visualization of decision-making mental simulations of the ORPHEUS agent which controls the goalkeeper. **Context:** Virtual Nao goalkeeper for the RoboCup 3D Simulation League, in collaboration with the RoboCanes team[3].

An advantage of using mental simulation as a decision making mechanism is its intrinsic decision explanation ability. Classic logic-based systems also have this quality, but are not well suited for complex environments which we target using our approach. Moreover, newer approaches in literature are generally regarded as black-boxes which provide reasonable decisions but cannot provide an explanation on how they came to their conclusion. Our mental simulation based approach provides the human user with a comprehensive chain of events, communicated via the GUI, from which the reason for a certain decision becomes visible, although not explicitly specified within the system.

3.3 Discussion

In light of the difficulty of decision-making in complex dynamic virtual environments, we have proposed a generic mental simulation architecture that allows the agent it controls to simulate the behaviours of itself and others in different contexts. Our aim was to exploit the advantages of simulation as a mechanism to make predictions but also as a learning tool, using the imaginary outcomes of mental simulations and comparing them to reality. This way the agent can learn and perfect its mental models of the reality which it inhabits.

Our agent architecture consists in three layers of abstraction which can be regarded as asynchronous processes, as suggested by Buche et al. (2010); Buche and De Loor (2013). In contrast to previous works, the novelty of

our approach resides in its genericness. ORPHEUS extends the functionality of existing approaches, details on the mechanisms required by the mental simulation paradigm in a computational context, and provides a common framework for physical and behavioural anticipation. From the theoretical point of view, we have provided a conceptual overview of how mental simulation can be beneficial for decision-making in an intelligent agent and discussed the core mechanisms that enable reasoning within this framework, from perception, to cognition and finally to action. Thereafter, we illustrated the system architecture in detail, along with an algorithmic approach to the previously identified mechanisms.

ORPHEUS was implemented in `C++` with multi-threading support, with the aim for computation speed and portability. The complete implementation of our agent architecture has been published as open source, and can be freely accessed at **https://bitbucket.org/polceanum/orpheus**, in the hope that it will be used as support for agent decision making in open, dynamic environments. The architecture is compiled into a dynamic library, so that agents can make use of its functionality, with or without integrating it with third-party infrastructures. Compatibility has been validated with various Linux-based systems (of both low and high performance) and also on the Nao humanoid robot from Aldebaran Robotics[4].

[4] Aldebaran website: https://www.aldebaran.com

Contents of Chapter 4

4 Applications **61**

4.1 Introduction . . . 62
4.2 Angry Birds Agent . . . 63
4.2.1 Perception . . . 64
4.2.2 Abstract world . . . 66
4.2.3 Imaginary World . . . 66
4.2.4 Decision and Control . . . 67
4.2.5 Preliminary Results . . . 67
4.2.6 Competition Results . . . 69
4.3 Orphy the Cat . . . 71
4.3.1 Perception . . . 72
4.3.2 Abstract world . . . 73
4.3.3 Imaginary World . . . 73
4.3.4 Decision and Control . . . 74
4.3.5 Results . . . 74
4.4 Nao Robot Goalkeeper . . . 79
4.4.1 Perception . . . 81
4.4.2 Abstract world . . . 82
4.4.3 Imaginary World . . . 83
4.4.4 Decision and Control . . . 85
4.4.5 Results . . . 86
4.5 Discussion . . . 88

Chapter 4

Applications

Summary of Chapter 4

This chapter presents the instantiations of the generic agent architecture that was proposed in this thesis. Three applications are discussed, in the order of increasing complexity. First, an agent that solves puzzles in the complex 2-dimensional virtual environment of the "Angry Birds" game. Here, the agent receives input from the game in the form of snapshot images of the scene, being required to extract knowledge about the objects composing the scene and take the appropriate actions to win the game. Evaluation consists in three participations in an international competition, resulting in 8^{th} and 4^{th} place against other autonomous agents, and 1^{st} place (best agent) in a "man versus *machine" event. Second, we subject our proposed architecture to an adaptability experiment that includes environmental and behavioural aspects inside a dynamic 3-dimensional virtual environment. Here, our agent is able to* **adapt** *its behaviour based on changes in its environment and exhibits* believable *behaviour such as hiding behind obstacles, without any semantic knowledge of its real counterpart or any strategy. Finally, we increase the complexity of the environment and the agent's embodiment and introduced noisy perception, in the context of the "RoboCup 3D Simulation" competition. Our agent plays the role of the goalkeeper of a humanoid robot soccer team, obtaining improved results in comparison to another approach that was used in the competition.*

4.1 Introduction

The challenge addressed in this chapter is to show that our contribution – ORPHEUS – meets the requirements identified in our study of related works on computational mental simulation. More precisely, the goal is to demonstrate the **genericness** of ORPHEUS and its **effectiveness** in dynamic environments of various levels of complexity, and to assess the **environment characteristics** where it can function, the **types of targets** it is able to include in its predictions and the **cognitive functions** it is able to perform.

To this end, we propose three test cases (Table 4.1), in increasing order of complexity, where we instantiate ORPHEUS into agents which are required to achieve a set of goals, **without changing their architecture** from one scenario to another. The properties of the three environments used for the evaluation emerge from the study performed in Chapter 2 where we analysed the conditions in which existing approaches to computational mental simulation are deployed.

	ORPHEUS agents		
	Angry Birds Agent Section 4.2	**Orphy the Cat** Section 4.3	**Nao Robot Goalkeeper** Section 4.4
Space complexity	2D	3D	3D
Time complexity	real time	real time	real time
Time constraints	✓		✓
Embodiment complexity	low	medium	high
Physics challenge	✓	✓	✓
Behaviour challenge		✓	✓

Table 4.1: Challenges in each proposed evaluation scenario of ORPHEUS.

In the following three sections (4.2, 4.3 and 4.4), we present the context of each test scenario and discuss the characteristics of each ORPHEUS based agent with emphasis on how **perception** was constructed, the contents of the agent's **abstract world**, the functionalities of the **imaginary world** and how the agent makes **decisions**. Additionally, we evaluate the extent to which each agent is able to achieve its goal in the given environment conditions. The final section (4.5) of this chapter analyses how the three applications satisfy the criteria based on which we evaluated related works, and demonstrates the genericness and effectiveness of our agent architecture.

4.2 Angry Birds Agent

For the first evaluation of our proposed architecture, we implemented a minimalistic instance of our mental simulation framework within an agent designed to play the Angry Birds game.

Angry Birds[1] by Rovio Entertainment Ltd. is a popular game in which the player is in the control of various types of birds, each with special abilities, and goal is to help these birds seek revenge on their pig enemies (Figure 4.1). The game is based on a 2-dimensional physics simulation in which pigs are placed in structures built from blocks of various shapes, sizes and materials. The challenge of Angry Birds is to anticipate the effects of shooting birds towards the pigs and their buildings so that all pigs are destroyed before the player runs out of birds to shoot.

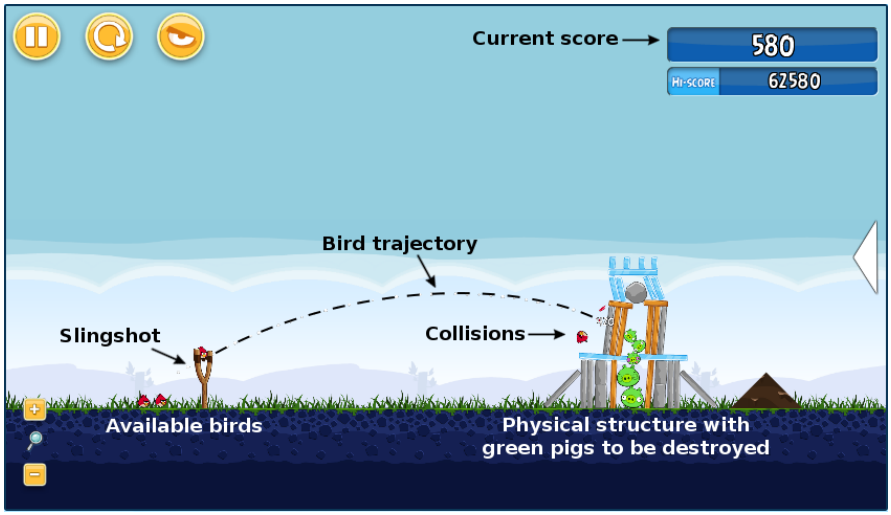


Figure 4.1: Annotated in-game view of Angry Birds.

The AIBirds contest[2] is an artificial intelligence competition that challenges researchers and AI enthusiasts alike to build an autonomous agent that can play Angry Birds as well as possible. In the 2013 edition of the AIBirds contest, which was organised as a symposium in the International Joint Conference on Artificial Intelligence (IJCAI), a naive agent [Ge et al. (2013)] was made available to the participants, which contains a set of tools to detect the locations of objects in the Angry Birds scene, and a simple decision algorithm that consists in randomly choosing a pig and shooting the bird on the trajectory that intersects the chosen pig. Object detection is limited to acquiring the bounding boxes, or minimal bounding rectangles (MBR), that are aligned to the game coordinate system.

[1] Angry Birds official website: https://www.angrybirds.com/
[2] AIBirds contest website: http://www.aibirds.org

Figure 4.2 illustrates how the instantiated agent fits into the specifications of the original conceptual framework by Buche and De Loor (2013).

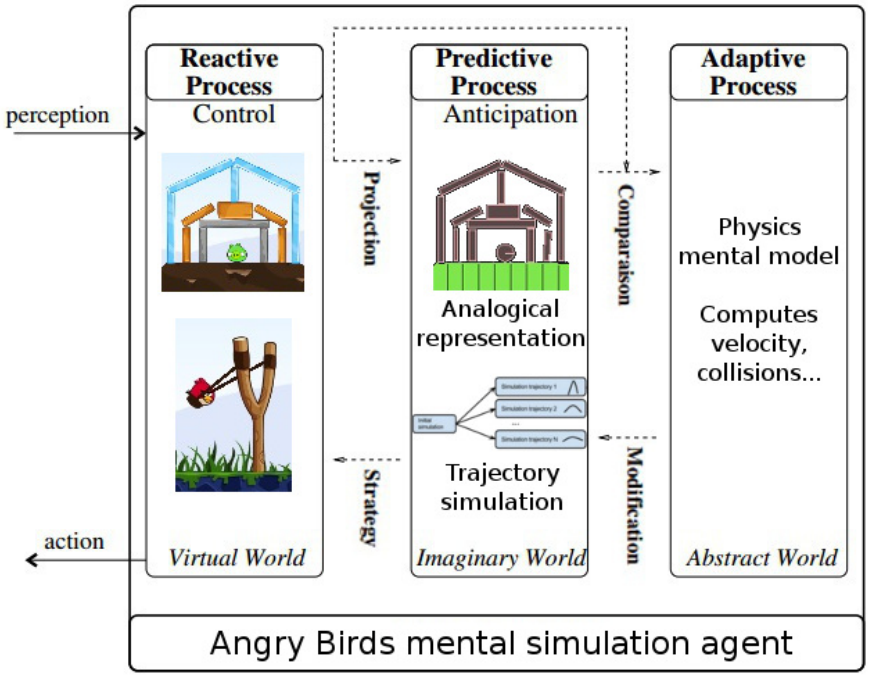


Figure 4.2: Conceptual overview for the Angry Birds bot.

We consider that the AIBirds contest represents an opportunity to test and evaluate our architecture in the context of a physical environment with moderately stochastic results which are due to the unavailability of the game's inner mechanisms. In this section we describe the implementation of this agent, with regard to the proposed architecture.

4.2.1 Perception

The game state is obtained by the agent through an extension for the Google Chrome browser which returns 2-dimensional images of the game, identical to what a human user would see when playing the game. The challenge for the agent is to extract useful information from these images in order to play the game.

An approach to extracting general solid rectangles (GSR) from MBRs was proposed by Ge and Renz (2013), and consists in defining an algebra to describe the GSRs and computing all possible contact combinations and finding a stable configuration that satisfies all the requirements. Using an algebra to reason directly on MBRs does not require further analysis of the original image. However, due to the assumption that the MBRs are correct,

the case in which one object is not detected can trigger a significantly different result, as it may have played an important role in the scene configuration. Furthermore, in the rare situation where MBRs are vertically aligned, there may exist two mirrored possibilities to represent the GSRs.

In order for the mental simulation to reflect the game environment, one must take into account boxes that are rotated either as result of collision or that being their initial state. Our approach to this problem was to improve the existing algorithm by taking into account the already calculated positions for each object, computing the set of pixels that make up the object using a filling algorithm, and fitting the minimum rectangle onto the convex hull of each pixel set by iterating over all edges of the hull. Results of this approach are illustrated in Figure 4.3, where (c) represents an overlay of the detected pixels (white) and the points of the convex hull (black), and (d) shows the minimum rectangles fitted on the convex hulls, which also represent the mental entities (image taken from a rendering of our virtual scene).

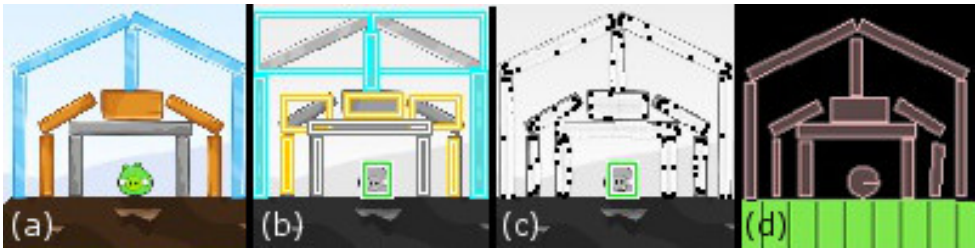


Figure 4.3: Obtaining correct object positions: (a) original scene image, (b) default bounding boxes, (c) improved algorithm, (d) obtained virtual reconstruction.

Additional to this detection process, the dimensions of detected rectangles are equalised with similar instances, based on the fact that the game uses fixed size objects. Furthermore, because it is possible for the reconstruction to be unstable, objects may fall without being touched. To avoid such scenarios, the agent makes the assumption that the configuration in which the environment is found before a shot is a stable one. Therefore, objects are held in place within the mental simulation, and must become "active" in order to move. At the start of the mental simulation, only the shooting bird is active, then it activates other objects upon collision. In other words, each object becomes active if a collision happened between itself and an already active object.

Using original image analysis to compute GSRs is arguably more computationally expensive, but it allows to limit detection errors to a local scope. Moreover, by using a convex hull to determine the rectangles, the number of points of the hull can be used to decide, with an acceptable accuracy, which objects are rectangular and which are circular (i.e. circular objects have more hull vertices than rectangular objects).

4.2.2 Abstract world

Once detected, objects are reconstructed inside the agent's imaginary world. All objects in the mental simulation are animated by a two-dimensional physics engine, which corresponds to a mental model within the proposed framework. Each type of bird (red, yellow, blue, black and white) is modelled separately, depending on its special ability. These mental models allow the agent to imagine the effects of performed actions and the environment evolution in time.

This representation allows different behaviours to be easily assigned to sets of instances. In the case of this application, the values of attributes are empirical in origin and not automatically learned, for convenience. However, this will be the ultimate goal of the proposed framework.

4.2.3 Imaginary World

Once the environment structure is interpreted, the agent will use its imaginary world to create a number of simulations that differ by a small shooting angle (Figure 4.4).

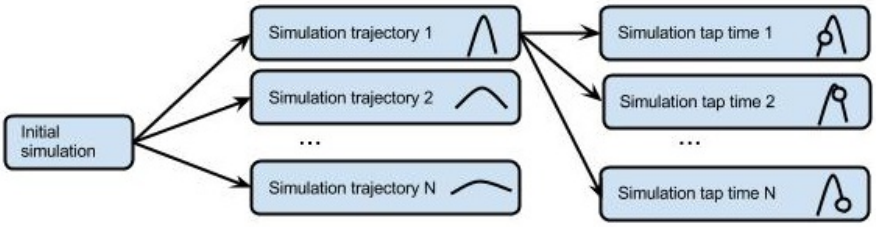


Figure 4.4: Mental simulation tree with two levels: trajectories and tap times.

The depth of the mental simulation tree can be increased, by adding the "tap" function which triggers special behaviour in some types of birds. For each shooting angle, the agent is able to choose a time to perform the tap which consists in another array of possibilities, therefore adding another level in the tree (Figure 4.4). This is done similarly to the first level, through duplication. This method proves to be computationally inexpensive, as the initial object recognition and mapping are not remade, but their results are copied and simulated in another way (different angles or tap times, but with the same initial scene configuration). In the mental simulations, a special bird behaviour triggers similar effects as in the real game, for example blue birds spawn new instances at different angles, yellow birds gain a speed boost, black birds explode and white birds shoot projectiles downwards. This way, the model can be modified to better fit the game without changing the decision making process.

4.2.4 Decision and Control

Based on the results of mental simulations, the agent will choose one which best suits its goal: to destroy pigs. However, the mere evaluation of how many imaginary pigs are killed does not prove to give the most successful results. This issue is not because the selection process is incorrect, but because the model of the environment is not completely accurate. This causes mental simulations to give similar but not precise outcomes compared to the actual game scenario. For example, it may happen that in one mental simulation, conditions are just right for a record number of pigs to be killed, although it is not the same case in the real game; this leads to a erroneous score record, and therefore the actions from this mental simulation are incorrectly chosen as the final decision.

Coping with uncertainty caused by a partially correct model can be addressed by considering in advance that results are stochastic in nature, and therefore taking into account a range of results instead of isolated cases. In other words, instead of evaluating each mental simulation by its final result, one can evaluate a set of results from minimally different imaginary scenarios to obtain a probabilistically good decision. This way, isolated cases of false success can be pruned out by the selection algorithm, leaving higher chances for the agent to take a truly efficient decision.

Once the Orpheus agent has made its decision, it will trigger the slingshot with the imagined shooting angle, by communicating it to the browser extension.

4.2.5 Preliminary Results

Our implementation creates a number of 106 mental simulations which differ by a shooting angle of 0.01 radians from each other. Normal computation time is approximately 15 seconds, however the mental simulations are executed at three times the normal speed, in accordance to the time limitation imposed by the contest rules.

We compare our implementation with the default "naive" agent provided by the competition organisers. This choice for benchmark was made due to the fact that the current default agent is an extension of the winning agent of the 2012 edition of the contest, that provides an improvement of the visual recognition module.

Because the default agent is based on a random target selection, and this may influence the resulted scores, we limited the levels used in our comparison to the first 9 levels of the freely available "Poached Eggs" episode of Angry Birds, which only feature red birds that do not have any special ability. Level difficulty is generally directly proportional with its number, while the maximum obtainable score depends on the number of pigs in the scene.

Lvl	Trial 1	Trial 2	Trial 3	Trial 4	Avg
1	30090	28960	28970	28970	*29247.5*
2	42650	34160	34160	43180	*38540.0*
3	31670	40730	40260	40260	*38230.0*
4	28470	18870	28160	28330	*25957.5*
5	65140	64500	64710	63510	*64465.0*
6	34010	24450	25630	24490	*27145.0*
7	20020	29390	27680	22910	*25000.0*
8	38280	36170	58010	38650	*42777.5*
9	32870	31770	32930	29090	*31665.0*
tot.	**323200**	**309000**	**340520**	**319390**	***323027.5***

Table 4.2: Scores of the default naive agent.

After evaluating the naive agent that was provided by the competition organisers (Table 4.2), we found it to fail a total of 22 times while playing 4 trials of 9 levels each. In comparison, the scores of the agent proposed in this thesis are shown in Table 4.3 and exhibit a 10.9% average improvement over the naive agent, and only 4 total failures to finish a level.

Lvl	Trial 1	Trial 2	Trial 3	Trial 4	Avg
1	32350	32350	32330	32330	*32340.0*
2	60820	52070	60630	51530	*56262.5*
3	42630	42630	41820	42630	*42427.5*
4	28120	28120	28430	28120	*28197.5*
5	64330	69100	63520	58070	*63755.0*
6	24470	33510	26270	33510	*29440.0*
7	33200	22660	23300	31940	*27775.0*
8	34710	47520	39120	44820	*41542.5*
9	24370	40010	40920	41440	*36685.0*
tot.	**345000**	**367970**	**356340**	**364390**	***358425***

Table 4.3: Scores of the mental simulation agent

The most failures of the naive agent were on the 7th level (10 failures) and the second most on the 4th level (4 failures). Our implementation also experienced difficulty in level 7 but with only 3 failures, and on level 9 with only one failure.

During the trials, the situation occurred where our agent obtained the same score on two or more attempts. Identical scores are due to the same decision taken more than once, as the visual recognition of the objects in the scene returned the same results. It is important to note that most of these

situations occurred when the agent solved the level using one bird (levels 1, 3 and 4). An exception to this rule was level 6, where due to the lower number of objects, the scene tends to give the same results after the first shot. However, this shows that given exactly the same conditions, the mental simulations are stable enough to output the same decision for the agent.

Tests show that errors in the model are bound to make the agent take less efficient decisions, because mental simulations do not correctly reflect the game reality. However, when the reconstruction has a higher level of correctness, the agent is able to find the most effective strategy to complete the game, as shown in Figure 4.5 which illustrates the second level of the freely available "Poached Eggs" episode of Angry Birds.

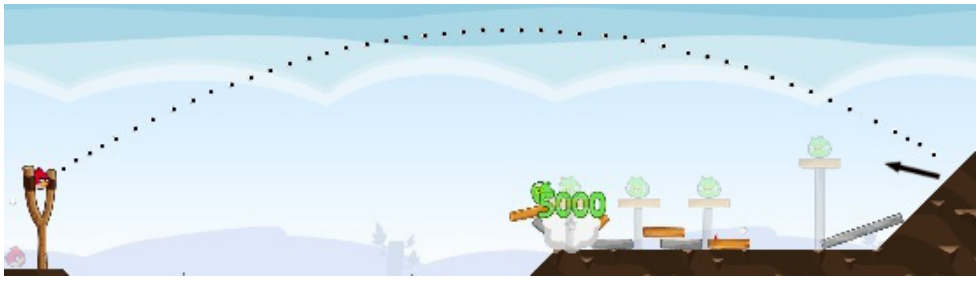


Figure 4.5: Best solution for Level 2 of "Poached Eggs" episode, found by our implementation (overlaid original state and highlighted trajectory)

These results indicate that it is possible to take good decisions even if the model of the environment does not perfectly match the game environment. However, to achieve such good results in the majority of scenes, further efforts must be made to improve the model.

4.2.6 Competition Results

In the AIBirds competition, our agent has obtained 8^{th} place in 2013 (scoreboard in Table 4.4), 4^{th} place in 2014 (Table 4.5) and 1^{st} place (best agent) in the Man *versus* Machine Challenge at the Australian National University "Open Day" in 2013, with score 243480, followed by HungryBirds with 173510 and TeamWisc with 135230. Over 50 humans participated at the ANU Open Day to compete against the qualified agents from AIBirds 2013. Only 3 of the humans could obtain a higher score than our mental simulation based agent, with a small margin.

Qualification			**Quarter Final 1**	
1. Angry-HEX	584600		1. Angry Hex	283970
2. Luabab	570530		2. Beau Rivage	179730
3. Beau-Rivage	537380		3. Naive	93090
4. Angry Concepts	532310		4. IHSEV	85560
5. Naive Agent	506070			
6. Hungry Birds	474810		**Quarter Final 2:**	
7. Team Wisc	452110		1. Team Wisc	209170
8. IHSEV	435950		2. Angry Concepts	207300
9. A. Wang	416530		3. Luabab	190310
10. Dan	396400		4. Hungry Birds	151480

Table 4.4: AIBirds 2013 competition results of top 10 entries (from 20 total), including our agent (IHSEV).

Qualification			**Quarter Final 1**	
1. DataLab Birds	423280		1. DataLab Birds	346260
2. PlanA+	372810		2. Angry BER	224860
3. s-Birds Avengers	361770		3. Impact Vactor	173710
4. Angry Dragons	317300		4. s-birds Avengers	167860
5. Naive Agent	302710			
6. Impact Vactor	298390			
7. Angry-HEX	294170			
8. IHSEV	292380		**Quarter Final 2:**	
9. Angry BER	253820		1. PlanA+	360920
10. BeauRivage	238080		2. IHSEV	277530
11. RMIT Redbacks	188890		3. Angry-HEX	129610
12. Auto Lilienthal	0		4. Angry Dragons	78970
Semi Final			**Grand Final**	
1. DataLab Birds	232790		1. DataLab Birds	406340
2. Angry BER	206680		2. Angry Dragons	243880
3. PlanA+	206620			
4. IHSEV	93100			

Table 4.5: AIBirds 2014 competition results, including our agent (IHSEV).

4.3 Orphy the Cat

In this second test scenario, we validate our proposed architecture by instantiating it in an autonomous agent which is given a goal in an environment that has the following properties: continuous 3D space, real time, physics (collisions, visual occlusions) and variable behaviour of other entities. In comparison to the previous scenario, the space complexity is now increased, other autonomous entities have been added in the environment which must be taken into account by our agent in its predictions and also, the embodiment of the agent is more complex, with a larger set of possible actions to perform.

Based on the previously mentioned environment property requirements, we chose a scenario (Fig. 4.6, left) inspired from the predatory behavior of the Felinae subfamily, which consists mostly in small to medium sized cats. It is common for their prey, such as birds, to have wide fields of sight and to be capable of predator detection and avoidance over a distance.

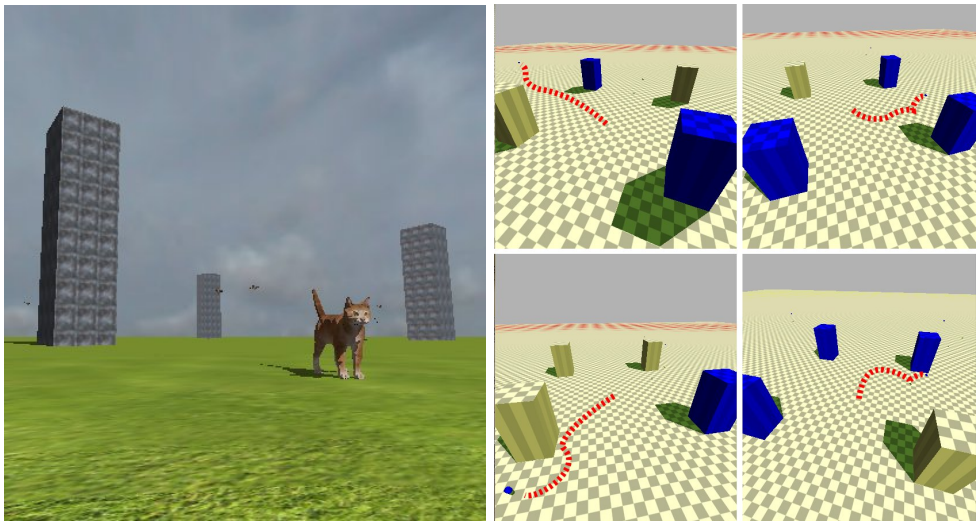


Figure 4.6: Instantiation of the "real world" (3D environment) featuring the cat and autonomous birds (left), and the "imaginary world" of the cat with examples of mental simulations where it evaluates different action sequences (right).

In the natural world, hunting behaviours of cats can be categorised into "mobile" and "stationary", the former being applied in areas abundant in prey while the latter, which consists in ambushing, when the cat is located in areas of interest [Dennis C. Turner (2000)]. Cats generally employ a stealthy approach followed by a short rush before striking [Kleiman and Eisenberg (1973)]. In the following, we discuss the instantiation of the generic architecture (Fig. 3.2) within the context of the chosen test scenario (Fig. 4.6), aiming to obtain adaptive behaviour from the agent through mental simula-

tion, without hard-coding any strategies of its real counterpart; i.e. the agent has no information/semantics that it represents a cat.

In the test environment, the ORPHEUS agent architecture is instantiated in the cat character. Here, birds have boid-like behaviour [Reynolds (1987)] and are also able to see the cat at an arbitrary distance, given that their vision is not obstructed by an obstacle (birds cannot see through obstacles). A bird that can see the cat will flee (turn away), making it more difficult for the cat to catch it. Figure 4.7 shows how this agent fits into the original conceptual framework by Buche and De Loor (2013).

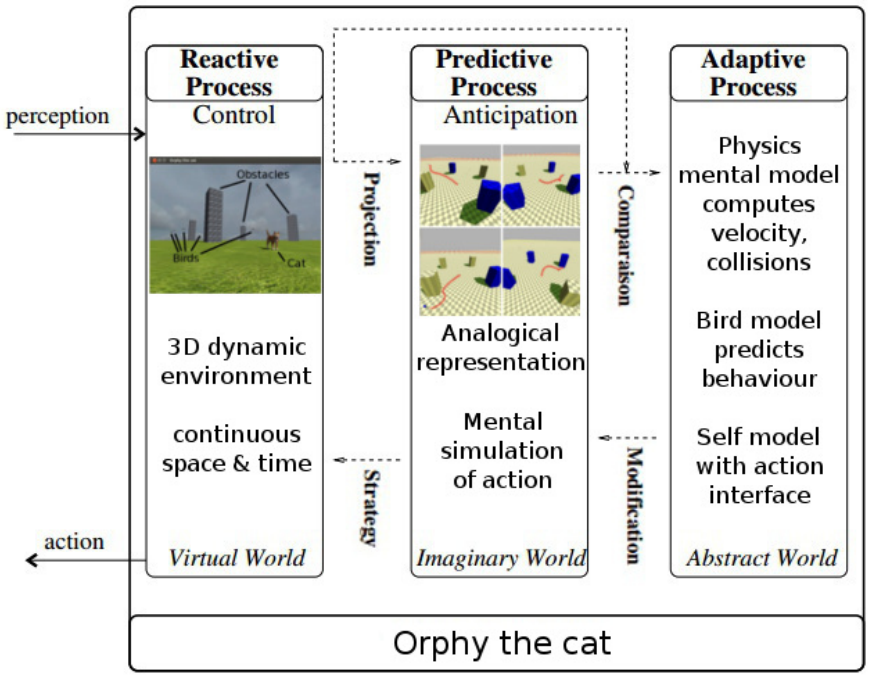


Figure 4.7: Conceptual overview for Orphy the cat.

4.3.1 Perception

We configured the agent to observe the environment (i.e. extract a mental image) at an arbitrary interval of 0.5 seconds to account for real time perception. Perception consists in obtaining the list of entities from the environment and creating mental images to be used in predictions. As with the other test environments in this chapter, each entity within a mental image has a set of properties assigned, including position (3D), rotation, velocity (linear and angular), shape and size.

While in this test scenario the agent's perception does not include noise, the behaviour of birds in the environment is non-deterministic and thus will

lead to some level of prediction error with which Orpheus must cope.

4.3.2 Abstract world

The agent using our architecture is supplied with three models to evolve mental simulations (Fig. 4.8). First, a physics model, implemented using Bullet Physics, that applies collision effects and linear and angular velocities on entities. Second, a model that closely approximates the behaviour of "real" birds with any of the three possible view distance values. Third, a cat model with parameters which acts only as a control interface. These parameters are a set of possible actions: run forward (values: yes/no), turn (values: left/no/right) and jump (values: yes/no). Model to entity assignments are managed by the agent, so that the error between the behaviour of imaginary birds and reality, computed in this application using the Mean Absolute Deviation [Konno and Yamazaki (1991)], is minimised.

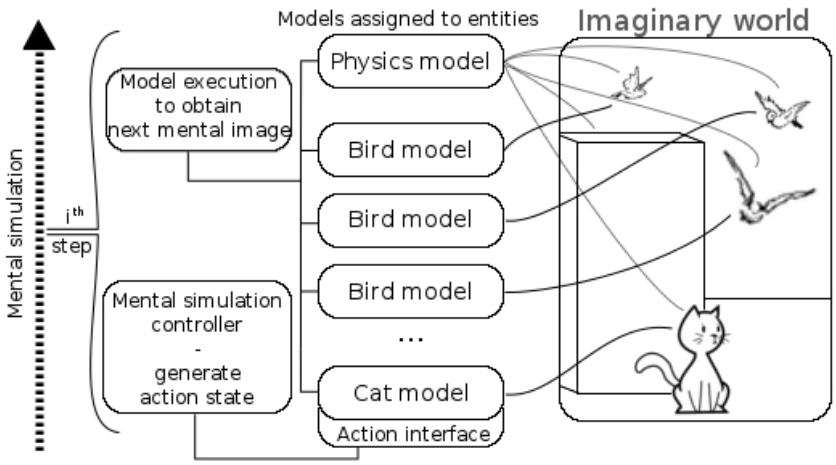


Figure 4.8: Models assigned to each entity in the imaginary world and action controller for the system's self model (cat). All models contribute to the evolution of the mental simulation at each step.

The agent can also automatically learn the trajectory of the birds in real time using a mental model based on the Kernel Recursive Least Squares [Engel et al. (2003)] algorithm. The mental model can seamlessly replace the specific model for birds, using the model abstraction capability of Orpheus.

4.3.3 Imaginary World

The mental images obtained through perception are then used to branch an arbitrary number of 20 parallel mental simulations (Fig. 4.6, right) of varying time lengths (1 to 15 simulated seconds) so as to predict the outcomes

of different courses of action into the future. On the test configuration, we obtained a speed ratio of ∼13x; i.e. 15 seconds of mental simulation take ∼1.15 real seconds to compute. When each mental simulation has finished, it is replaced by a new one based on the currently perceived state of the world. A mental simulation will finish when it reaches its time limit or it leads to goal achievement. If there are no mental simulations that lead to goal achievement, the agent will rest still until a solution is found. The number and duration of mental simulations can be varied to increase/decrease chances of a valid solution and to find a near/far solution, respectively.

Using the ability of Orpheus to interweave the effects of several models within mental simulations, our agent is able to anticipate both environmental and behavioural aspects of the environment, and to use them to predict the effects of its actions.

4.3.4 Decision and Control

If at any time the distance between the cat and a bird falls below a threshold value (the cat is required to jump), the bird disappears and is considered caught and the cat's goal is achieved. Once the goal has been satisfied in a mental simulation from the cat's imaginary world, the history of actions performed within it, which correspond to the list of parameter values, are applied in the real world. To avoid any bias from "cleverly chosen" strategies as used by Buche et al. (2010), the cat's mental simulations include uniformly distributed random parameter values (action sequences). The goal of this application does not rest in the random action generation itself, as this could be replaced by heuristic search or similar methods, but in illustrating the use of the proposed generic architecture.

Once the Orpheus agent has made a decision, which consists of the imagined set of actions to perform, it will execute these actions through its embodiment, while synchronizing the virtual time of the imagined actions with the corresponding real time of the environment.

Our hypothesis is that through the use of mental simulation, the cat, which is controlled by our system, will adapt its behaviour in function of the distance at which birds are able to perceive the danger. By having obstacles in the environment, we hope that the cat will use them to improve its success rate in catching birds, as *this behaviour is not explicitly specified.*

4.3.5 Results

The architecture and resulting agent were implemented in `C++` with multi-thread support and the tests in this work were run on a Intel® Xeon® 2.80GHz, Ubuntu 12.04 machine with 8GB RAM. We have also tested the system on less powerful machines, where expectedly less simultaneous mental simulations were possible.

The experiment consists in six test cases that are constructed by varying obstacle shape (with or without corners) and bird view distance (0 *units*, 50 *units* and 100 *units*). In the extreme case (100 *units*) the cat is not able to catch birds if no obstacles are present in the environment.

Each of the six test cases (Figures 4.9 and 4.10) consists of 4 symmetrically placed obstacles and 10 birds which fly in fairly regular circular patterns unless the cat is seen and avoided. Amongst test cases, distances at which the birds can spot the cat are varied and the obstacle shape is also subtly changed.

Results show that the increase in the view distance of birds significantly influences the frequency of locations for the cat, regarding available obstacles. As a general rule, the increase in bird sight determines values in the histograms for the cat (Figures 4.9 and 4.10, Subfigures "b") to shift left towards lesser distance from obstacles, and bird catching location frequencies (Figures 4.9 and 4.10, Subfigures "c") to invert slope as obstacles prevent birds to see the cat. That is, the agent's behavior consistently adapts to the variation of bird view distance, resulting in a set of distinguishable emerging techniques (Fig. 4.11) that increase the **believability** of the agent's behaviour.

The application of these observed hunting techniques conforms with intuition; namely the cat simply chases (Fig. 4.11.a) birds with no sight, but when the view distance is increased, it resorts to ambush (Fig. 4.11.b), hide and chase (Fig. 4.11.c) and, the strongest example of anticipation observed in the experiment, the blockade (Fig. 4.11.d) where the agent uses the predicted collision of the bird with an obstacle to catch it.

Varying obstacle shape also determines subtle adaptation in our agent, an example of which is illustrated in the 5th test case (Fig. 4.10, Subfigure 5.c) which features a gap in the cat's position relative to obstacles. This is caused by the increased difficulty of hiding behind cylindrical obstacles which also determines a less prominent use of the ambush technique.

In the following test cases, the simulation paradigm is used to control the cat, first using cuboid obstacles (Fig. 4.9) and afterwards cylindrical obstacles (Fig. 4.10).

Figure 4.9 illustrates three cases, in which cuboid obstacles are used, where: (1) the view distance of birds is 0 *units* (birds cannot see the cat); (2) the birds can see the cat up to a distance of 50 *units* (medium distance, roughly the gap between two obstacles); (3) the birds can see up to a distance of 100 *units*. The choice for these values was made so that the ability of the cat to simply chase and catch the birds, without more complex strategy, gradually disappears. In these three cases, the obstacles have a cuboid shape and are represented by four bold squares in the figures.

Results show that the increase in the view distance of birds significantly influences the frequency of locations for the cat, regarding available obstacles. As a general rule, the increase in bird sight determines values in the histograms for the cat (Figs. 4.9:1.b, 4.9:2.b, 4.9:3.b) to shift left towards lesser distance to obstacles, and bird catching location frequencies (Figs. 4.9:1.c, 4.9:2.c,

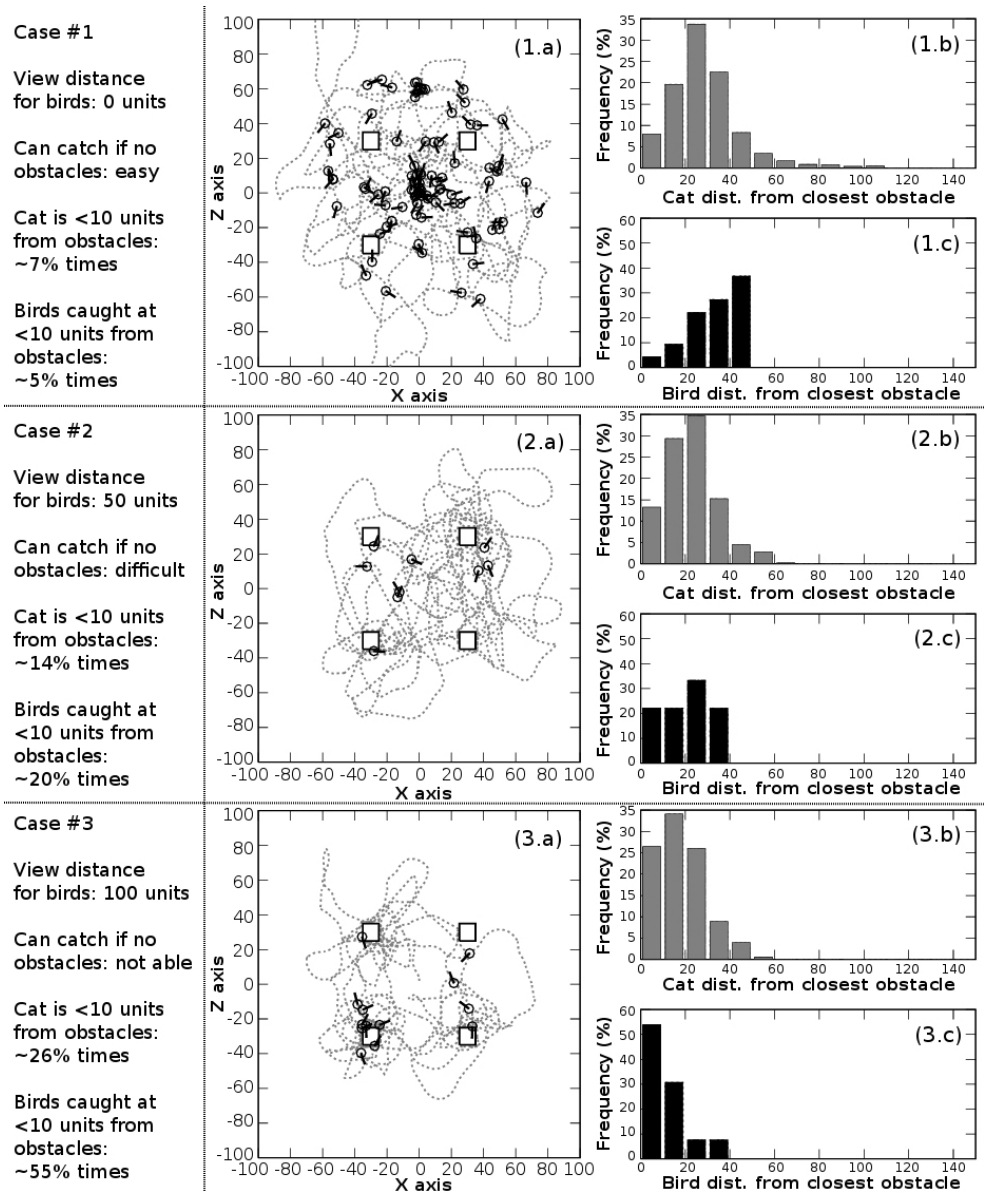


Figure 4.9: Agent behaviour in scenarios with cuboid obstacles (four black squares). Case (1) corresponds with 0 *units* view distance for birds, case (2) with 50 *units* and case (3) with 100 *units*. Subfigures (#.a) illustrate cat (grey) and caught birds (black) traces. Distance histograms are shown for the cat (#.b) and catches (#.c) relative to the closest obstacle.

4.9:3.c) to invert slope as obstacles prevent birds to see the cat. As follows, the same general rule also applies when the obstacle shapes are changed to

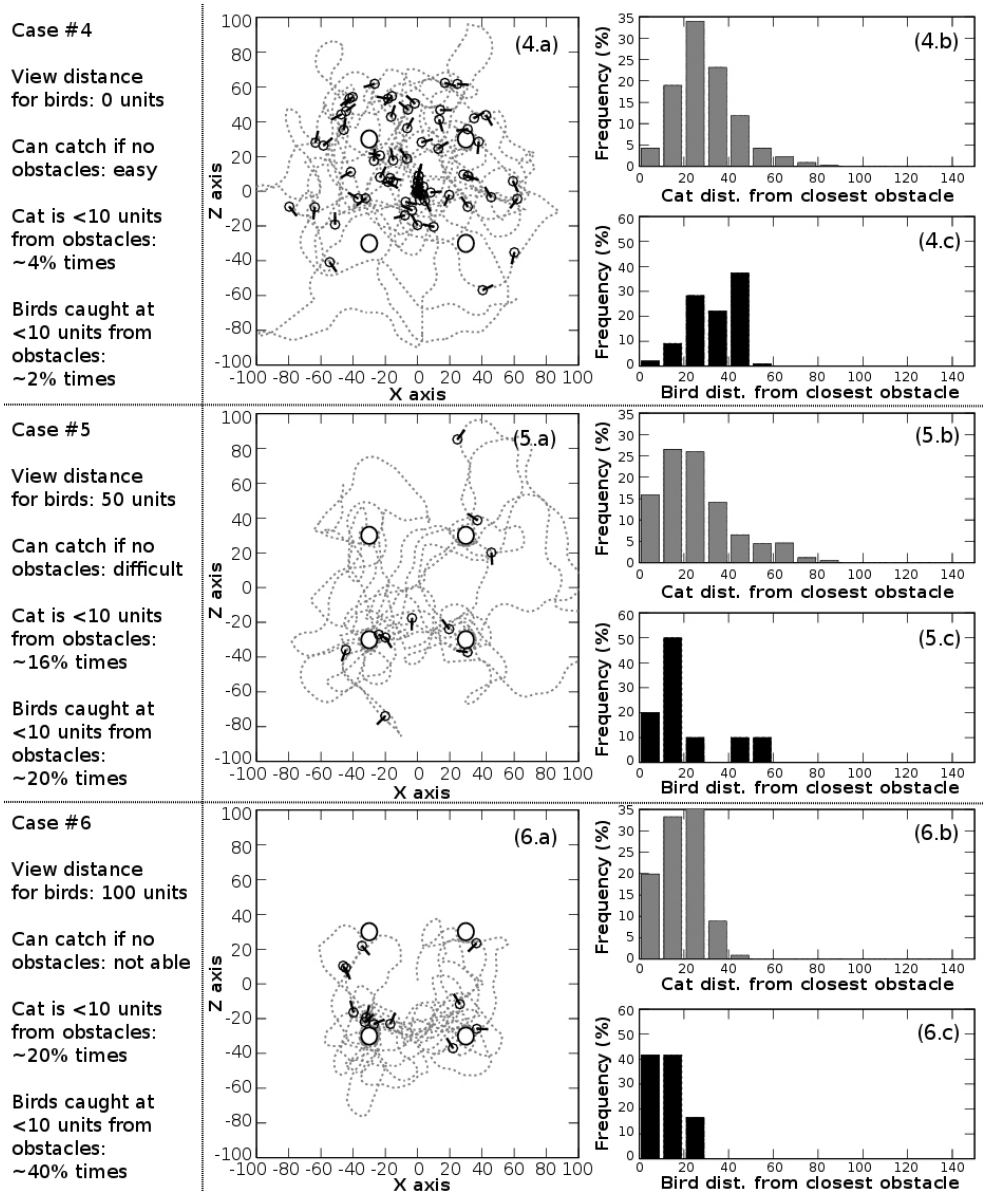


Figure 4.10: Agent behavior in scenarios with cylindrical obstacles (four black circles). Case (4) corresponds with 0 *units* view distance for birds, case (5) with 50 *units* and case (6) with 100 *units*. Subfigures (#.a) illustrate cat (grey) and caught birds (black) traces. Distance histograms are shown for the cat (#.b) and catches (#.c) relative to the closest obstacle.

cylinders.

Figure 4.10 illustrates the following three cases (4), (5) and (6) in which

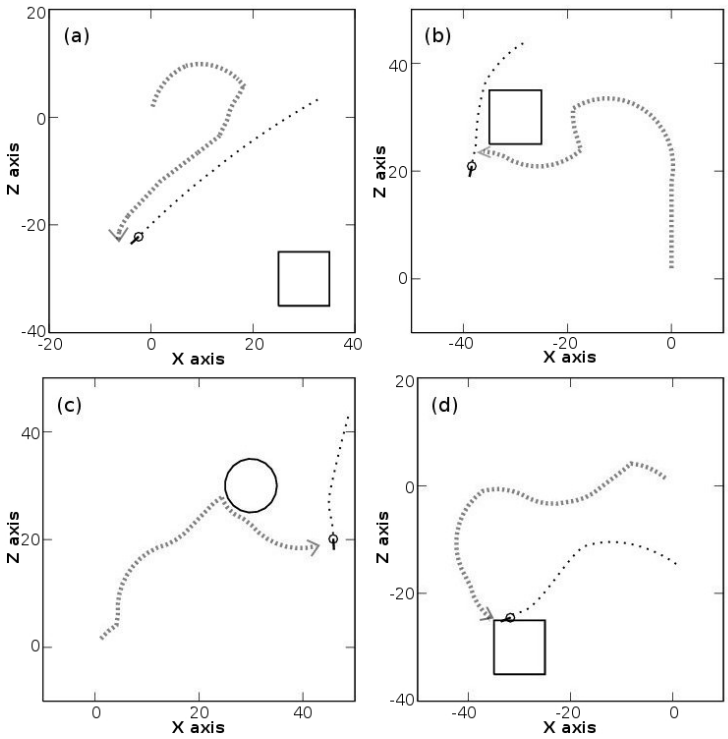


Figure 4.11: Examples of solutions found by the system for prey catching: (a) simple chase, (b) ambush, (c) hide and chase, (d) blockade. Cat traces shown with grey arrows, and catch locations with directed circles based on bird orientation.

the conditions are the same as in cases (1), (2) and (3) respectively, with the exception that now the obstacles are of cylindrical shape, represented as four black circles. The cylindrical shape leaves less hiding space for the cat and therefore may trigger different behaviour of the system, due to the increased difficulty for stealth.

In the following, we examine and discuss each of the six test cases, referring to which techniques are used more often.

Case (1) (in Fig. 4.9): the cat's path appears as a gradient with highest frequency (35%) between 20 and 30 *units* from obstacles, while walking very close to them only happens in 7% of its trajectory. In these conditions, birds are caught by chasing (Fig. 4.11:a) in many locations on the map, with greater ease in the centre where they are more frequently guided by the gaps between the obstacles.

Case (2) (in Fig. 4.9): the birds' view distance is high enough to prevent easy capture. The cat's behaviour in these conditions adapts to use the obstacles as leverage to catch prey. Frequency of walking very close to obstacles increases to 14%. Beside still being able to catch a small number of birds with

the simple chase strategy (Fig. 4.11:a), the hiding behaviour allows the cat to *ambush* (Fig. 4.11:b) birds which are unaware of the its location. Moreover, two additional strategies can be distinguished: first, a variant of the ambush where the cat is hidden but begins chasing over a short distance to catch the prey (Fig. 4.11:c) and second, due to the cuboid shape of the obstacles, the bird can be blockaded by the cat (Fig. 4.11:d) making use of level 1 **Theory of Mind** (level 1) to anticipate that the cat's approach will change the trajectory of the bird into being blocked by the obstacle and becoming catchable.

Case (3) (in Fig. 4.9): the cat is no longer able to catch prey without hiding. Results show a significant approach towards obstacles (frequency of walking very close to obstacles peaks to 26%) and the majority of birds are caught by ambush (Fig. 4.11:b), while there exist situations where the hide-chase strategy is also viable (Fig. 4.11:c).

Case (4) (in Fig. 4.10): cuboid obstacles are replaced by cylinders, little difference can be observed in comparison with case (1), due to the zero view distance of birds.

Case (5) (in Fig. 4.10): while the distance to the closest object drops, as in case (2), for both the cat and the caught birds, this happens with less amplitude. The effects triggered by cylindrical obstacles become visible, namely with a predominance of the hide-chase technique (Fig. 4.11:c) over the ambush (Fig. 4.11:b). It is notable that the caught bird histogram (Fig. 4.10:5.c) features a gap, caused by the more prominent use of hide-chase.

Case (6) (in Fig. 4.10): finally, like with case (3), the cat's possibilities are reduced to ambushing (Fig. 4.11:b) and less prominent hide-chase (Fig. 4.11:c) techniques. Due to the rounded shapes of the obstacles, the blockade technique (Fig. 4.11:d) is no longer used.

We note that in the cases where birds have zero view distance, all the four techniques are applied by the system. The fact that only a subset of techniques are more prominent in more difficult situations is true because in the cat's imaginary world, only simulations that lead to a solution are applied. These results are reproducible with a degree of variation in environment configuration such as number of birds and obstacle number and positions, as the agent behaves in the way that leads to goal achievement given different contexts.

4.4 Nao Robot Goalkeeper

With respect to the former two evaluations, this final test scenario further increases the complexity of the agent's embodiment and environment physics in addition to maintaining the 3-dimensional space and continuous time from the second test case. Moreover, a new challenge for Orpheus is represented by the time constraints which are far more strict than in the AIBirds competition,

as the goalkeeper must act in a complex fashion in less than one second from the time when the ball is shot.

RoboCup[3] Robot Soccer is an annual international competition that brings together researchers in robotics and artificial intelligence towards a common goal of obtaining robots that will play the game of soccer against humans by the year 2050. The competition is divided into several leagues, each with its own rules and regulations. Two of these leagues, 3D Simulation League and Standard Platform League, use the Nao humanoid robot from Aldebaran Robotics[4] within realistic computer simulations and real world scenarios respectively.

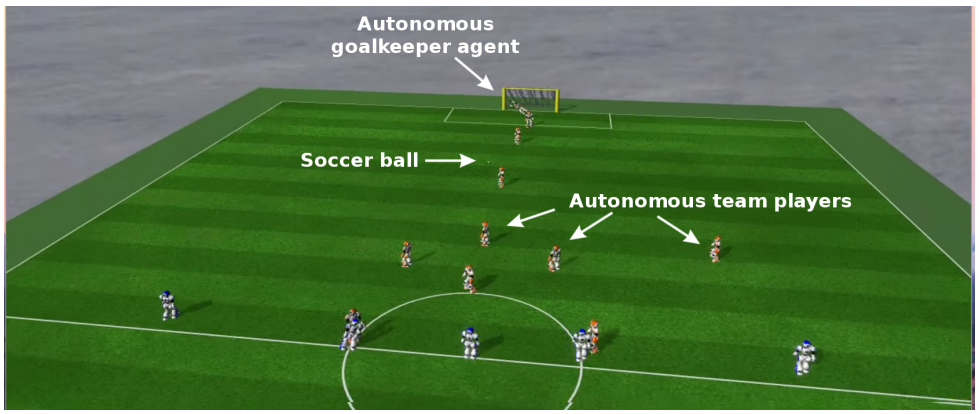


Figure 4.12: Annotated RoboViz [Stoecker and Visser (2012)] visualization of a RoboCup 3D Simulation League world championship final between the teams *UT Austin Villa* and *RoboCanes*, held in João Pessoa, Brazil, in 2014.

In the RoboCup 3D simulation league, teams of 11 virtual replicas of the Nao robot play against each other in a virtual game of soccer (Figure 4.12). Our approach consists in enhancing the team's goalkeeper with a mental model of itself that it can simulate ahead of time and make decisions on the efficiency of its current actions and eventually change its ongoing behaviour.

We first give a brief introduction to the concepts of this approach, the requirement of more precise velocity estimation and finally how the goalkeeper can use this cognitive layer to improve its results. Figure 4.13 shows how this agent fits into the original conceptual framework by Buche and De Loor (2013).

To enable the goalkeeper agent to predict its environment, itself included, we use our generic agent architecture for mental simulation to evaluate the outcomes of its actions in accordance with other objects (e.g. ball) and agents that populate the environment it is placed in. To clarify, the goalkeeper has its own functional representation of itself and the ball, and is able to manipulate

[3]RoboCup 2015 website: http://www.robocup2015.org
[4]Aldebaran website: https://www.aldebaran.com

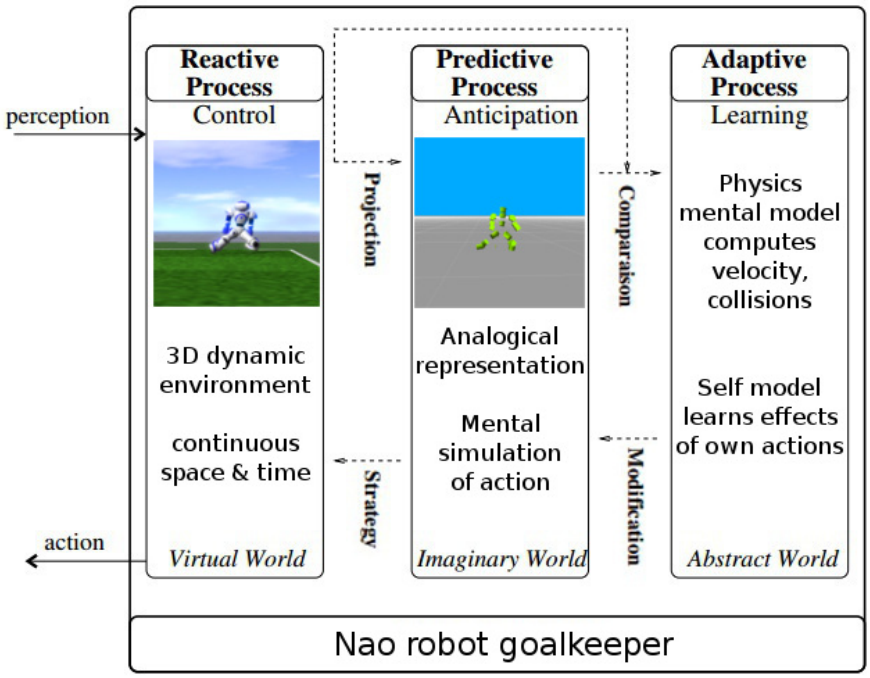


Figure 4.13: Conceptual overview for the Nao goalkeeper.

it towards improving its behaviour in the game. Providing the agent with models of itself and its environment enables it to evolve and evaluate various courses of actions in its imagination, based on which it can make decisions. This cognitive layer functions in parallel to the main agent behaviours, which makes it well suited in the context of robot soccer agents where interruptions in sending motor instructions cause unwanted results.

Two important aspects lead to a successful use of the mental simulation paradigm: perception and mental model accuracy. First, due to perception noise, data of the position and velocity of the ball and robot must be filtered so that future events can be successfully predicted. Second, the models of environment physics and the robot's own body motion must be as accurate as possible to obtain useful predictions. Hence our focus is on improving velocity estimation and constructing the mental models for the robot.

4.4.1 Perception

In the robot soccer simulation contest, as well as for real settings, the trajectory of the ball is shaped by damping factors. However, we can safely assume that, as long as the ball does not collide with any object as it's travelling, it will describe a smooth non-linear trajectory. Unlike linear models, using a curve to approximate the trajectory, even on the X and Y axes where gravity

does not intervene, can account for the damping factors.

In this approach, we use the Levenberg-Marquardt algorithm as implemented in Dlib-ml [King (2009)] to fit a curve on the raw perception data. The choice of algorithm was based on its good performance in practice, efficiency on small input datasets and that it allows describing the curve model, as is the case for the goalkeeper's perception of the ball trajectory. We develop the damped motion equations $x_n = x_{n-1} + v_{n-1}\Delta t$ and $v_n = \zeta v_{n-1}$ over multiple time steps, resulting in the model to be used in the algorithm:

$$\frac{p_1 \Delta t(1-\zeta^n)}{1-\zeta} + p_0$$

where Δt is the time difference between two perception frames, ζ is the damping factor, n is the step number, p_0 and p_1 are the position and velocity parameters respectively. With this approach, we store the ball perception frames after the kick and refit the model parameters at each step to gradually improve accuracy (Fig. 4.14).

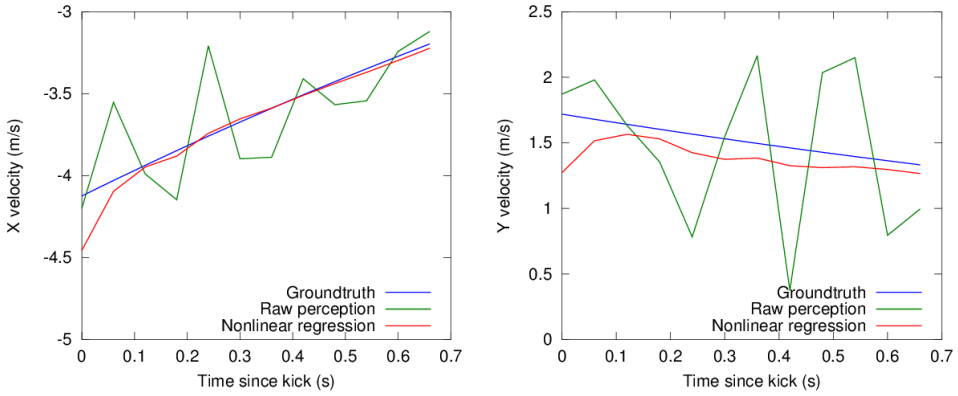


Figure 4.14: Comparison of ball velocity values for X (left) and Y (right) axes over time, since the kick.

Similar results to those shown in Figure 4.14 are also obtained for the Z axis, where the model is extended to account for gravitational acceleration:

$$p_2 \Delta t^2 (n - \frac{1-\zeta^n}{1-\zeta}) + \frac{p_1 \Delta t(1-\zeta^n)}{1-\zeta} + p_0$$

The obtained velocity estimation enables more accurate prediction of the ball for default behaviour such as walking towards the ball but also for prediction using mental simulations.

4.4.2 Abstract world

Once the ball can be predicted, the second step in this approach is to have the robot learn the effects of its actions and be able to reenact them in its

imaginary world. This produces a set of mental models which are then used to perform mental simulations.

To this end, we rely on querying the location of each of the robot's body parts which are represented with primitives in the imaginary world. Various actions can be learned by having the robot perform it, while a model training module provided by the ORPHEUS framework distributes the data to the models which are being trained. For the goalkeeper's actions, we use the K-Nearest Neighbors (KNN) algorithm as implemented in MLPack [Curtin et al. (2013)]. The dataset is constructed with the position, velocity and rotation of each body part at each time step. The number of learning trials (performed actions) varies depending on the type of action; simpler actions such as the long-step (Fig. 4.15) requires as few as three trials while walking or raising arms towards a given location requires between 30 and 40 trials to properly generalise.

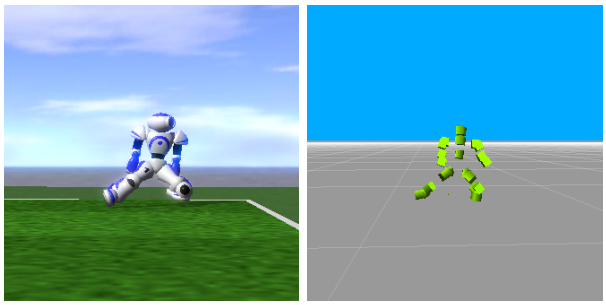


Figure 4.15: Long step action (left) with associated learned mental model (right).

Reenactment of each learned action with KNN uses the time and position of the robot's body parts and finds the closest corresponding velocities and rotations. From this point forward, ORPHEUS can perform mental simulations on demand, using the learned models through the process briefly described in the following.

Perception data (robot, body part and ball position, velocity and rotation) at a given moment in time is submitted to the cognitive layer and compiled into a mental image. Mental images represent the state of the imaginary world at a given time. Starting from this initial image, the mental simulation process constructs subsequent mental images by employing the available mental models. This process leads to obtaining a future state of the environment, to a certain accuracy, ahead of time. By evaluating this future state, the robot can make a range of decisions, as detailed in the following.

4.4.3 Imaginary World

There are limiting factors on how the goalkeeper agent can behave to deflect the ball. The main such factors are the high ball velocity achieved by other teams in the RoboCup 3D Simulation League, which gives little time to react. Moreover, the robot does not have the ability to move fast enough to reach the ball in many situations. There are however situations when extending its limbs or diving may help deflecting the ball.

Given that some actions can block the robot (such as leg extension or diving), we intend to tackle the problem of deciding whether or not to perform an action and if so, which one is more favourable, through self monitoring.

Testing showed that simply moving towards the predicted position of the ball when it would reach the defence zone performs well, given satisfactory estimations of the ball position and velocity. Therefore, starting from a reactive decision to move towards this location of interest, we use mental simulations to predict whether or not simply walking would enable the robot deflect the ball. As soon as the kick is detected, a coarse future location of the ball is computed and the robot starts moving towards that location. In parallel, it uses perceptual information to imagine the success or failure of this strategy.

We studied prediction accuracy (Fig. 4.16 with a set of 100 penalty shots from random locations, 5 to 10 units away from the goal gate. This resulted in an average available time of ∼0.5 seconds, after subtracting the time required to obtain reasonable ball velocity accuracy (∼0.1s) and the time required for mental simulations to finish; in our implementation, mental simulation speed ratio is ∼3x, i.e. the robot can simulate 1 second of real events in ∼0.33 seconds.

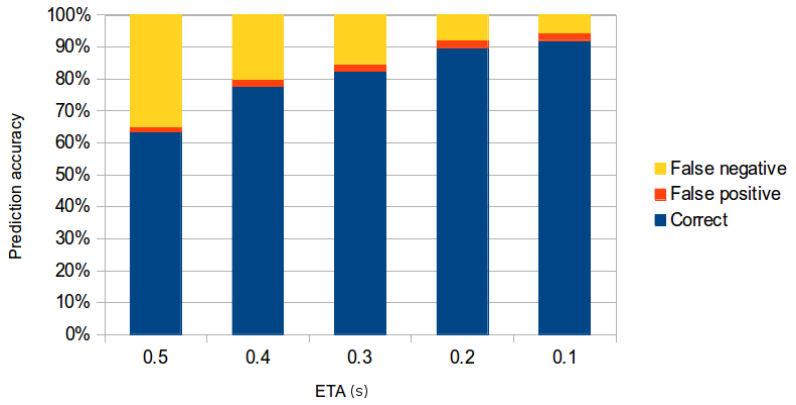


Figure 4.16: Outcome prediction accuracy, with ball Estimated Time of Arrival (ETA).

As shown in Figure 4.16, the accuracy of predicting the outcome of a set of actions rises above 50% (random choice) as soon as valid data is acquired from

the environment, and improves with time. Prediction failure is caused by the difference between mental models and reality, which could be later improved by learning more accurate models.

Concurrently with self monitoring, the goalkeeper also imagines and evaluates a set of actions that can be performed. This results in a set of solutions in the case in which it decides the current set of actions will not be effective. The goalkeeper uses the monitor prediction as a trigger or inhibition of changes in its behaviour. Therefore, if a failure is predicted, it will choose one of the previously imagined solutions if it had predicted that it will deflect the ball. From a cognitive point of view, the agent imagines the answer the question "what if I am not able to save the goal ?" which is a form of **counterfactual reasoning**.

4.4.4 Decision and Control

The decision process of the ORPHEUS goalkeeper consists in comparing the default scenario where it does not perform any special movements to the alternate futures where it chose to take action. A score is computed on the future mental images of each scenario using a distance metric to determine how close its imagined body pose was to the ball's location, and based on this score it selects one scenario which has a larger probability to deflect the ball (Figure 4.17).

Figure 4.17: The decision process of the ORPHEUS goalkeeper.

Once a decision has been made, the agent will transfer the action sequence from its imaginary world to its embodiment, while synchronizing the virtual time of the chosen mental simulation with the real time of the environment.

This experimental approach is limited by the set of possible actions that the robot can perform, and the time available to evaluate the outcomes of these actions. However, more efficient actions can be later added to increase the success rate. Moreover, more accurate mental models would also increase correct prediction rates, allowing a research focus on learning rather than tuning scripted behaviour for various scenarios.

4.4.5 Results

We empirically evaluated our approach on a range of kick distance intervals. We developed two versions so that the experimental mental model based prediction (ORPHEUS) can be evaluated separately. Additionally, we tested the naive goalkeeper behaviour, used so far by the RoboCanes team in previous competitions, as a benchmark. The evaluation consists in running a goalkeeper with an attacker that will kick the ball from a random distance within a specified interval, at a given angle.

There would be a large variance in the kicks of the striker, even if the striker kicks from the same distance and angle. Therefore, we reproduce the exact same kicks by assigning a velocity to the ball in the beginning of the kick. Thus, we are able to create a random sequence of kicks and repeat the exact same kicks multiple times using different goalkeepers. Variances in the results are only caused by variance in the goalkeepers behaviour, but not by variances in the kicks.

To better estimate the relationship between distance, shot angle and performance, we devised 2 meter intervals starting from 3 meters away (which is roughly the margin of the goalkeeper's area) to 15 meters away (midfield). The obtained results for each interval (Table 4.6) are the success rates for 100 different kicks repeated 30 times and averaged out (3,000 kicks per interval).

The experiment has been performed on an Intel Core i7-5930K machine with 6 cores, in collaboration with the RoboCanes AI & Robotics Research Group[5], led by Ubbo Visser at the University of Miami, Florida, USA. Results were also verified on two less powerful machines – a Quad Core Intel Xeon and a Dual Core Intel Pentium – and replicate the main experiment. The goalkeeper agent behaviour presented this thesis is an extension to the framework developed by the RoboCanes research group.

Results of distance-based evaluation (Table 4.6) show that our approach brings significant improvements over Naive goalkeeper which was used by the RoboCanes team in previous competitions.

The first important difference between the naive goalkeeper and our approach presented in this thesis is that previously, the naive keeper was instructed to place itself at the middle between the ball position and the goal, while the proposed goalkeeper's behaviour uses the goalkeeper's arc. Midway positioning can be exploited by shooting the ball over the goalkeeper, resulting in easy scoring, illustrated by the constant drop in success rates with respect to distance. This effect is significantly less prominent in the our approach, due to superior initial positioning of the goalkeeper.

Results of the experimental ORPHEUS version of the goalkeeper do not vary in a statistically significant manner from the NR version, which shows that it is feasible to integrate prediction in the goalkeeper without suffering a trade-off.

[5]RoboCanes website: http://robocanes.cs.miami.edu/

Table 4.6: Goalkeeper success rates for 100 random kicks from different distances. The kick angles are chosen randomly from -45 to 45 degrees. Each rate is averaged over 30 repetitions of the 100 kicks (standard deviation shown in parentheses).

Distance	Naive goalkeeper	Nonlinear Regression	ORPHEUS
3-5m	29.8% (2.2)	46.1% (2.8)	45.8% (2.3)
5-7m	22.0% (1.9)	50.4% (2.9)	50.2% (2.3)
7-9m	19.5% (1.9)	48.6% (3.0)	48.0% (3.0)
9-11m	17.8% (1.9)	47.5% (3.4)	46.9% (3.6)
11-13m	17.0% (2.4)	45.0% (3.5)	45.1% (3.8)
13-15m	14.8% (2.0)	43.9% (3.5)	44.8% (3.4)

Table 4.7: Average goalkeeper success rates for 100 random kicks as in Table 4.6, here grouped by angle (using the distance 5-7m).

Angle in degrees	Naive goalkeeper	Nonlinear regression	ORPHEUS
0-20	18.3% (2.1)	43.6% (3.3)	43.4% (2.8)
20-40	21.5% (2.0)	50.0% (2.9)	49.8% (2.7)
40-60	33.1% (2.7)	65.0% (3.3)	64.7% (3.1)
60-80	63.3% (3.6)	76.0% (3.5)	78.1% (3.5)

In order to more extensively evaluate our goalkeepers' performance, we also varied the angle from which the ball is shot and measured success rates (Table 4.7). Results show that the increase in success rates is directly proportional with the shooting angle both our approach and the Naive goalkeeper.

4.5 Discussion

The agent architecture proposed in this thesis was evaluated in three dynamic environments. The challenge was to demonstrate that ORPHEUS is a **generic architecture** for computational mental simulation that **effectively meets the requirements** identified in out study of existing approaches from literature. The outcomes of the evaluation of our architecture are compiled in Table 4.8.

	ORPHEUS agents		
	Angry Birds Agent	**Orphy the Cat**	**Nao Robot Goalkeeper**
Space complexity	2D	3D	3D
Time complexity	real time	real time	real time
Time constraints	✓		✓
Embodiment complexity	low	medium	high
Perception noise	✓		✓
Virtual world	✓	✓	✓
Real world			compatible
Environment prediction	✓	✓	✓
Behaviour prediction		✓	✓
Self prediction	✓	✓	✓
Prospection	✓	✓	✓
Navigation		✓	✓
Theory of Mind		Level 1	
Counterfactuals			on self
Believability		✓	

Table 4.8: Challenges identified in our study on related work, that were overcome by the proposed agent architecture, ORPHEUS.

Results show that, **without modifying the agent architecture**, the resulting agents were capable of satisfy the entire range of the competences that were pursued in the context of the computational use of mental simulation. Therefore, we are able to conclude that ORPHEUS is a **generic** approach which is suitable for agent cognition in complex, dynamic, real time environments with time constraints.

The agents obtained by instantiating our architecture were able to **adapt** to their environment, **imagine** solutions to obtain their goals and act accordingly. Their decision making made use of **contextual** information such as the interdependency of **environmental and behavioural** aspects, projections in **counterfactual scenarios** and **beliefs of other entities**.

Finally, in a nature inspired scenario, the agent displayed **believable behaviour** measured by its resemblance to the agent's biological counterpart.

Contents of Chapter 5

5 Conclusion **91**

5.1 Discussion . . . 93

5.1.1 Contributions of Orpheus . . . 94

5.1.2 Limitations of Orpheus . . . 95

5.2 Future Work . . . 96

5.2.1 Nao Soccer Lab . . . 96

5.2.2 Learning . . . 97

5.2.3 Planning . . . 98

5.2.4 Agent's Goals . . . 99

5.2.5 Emotions . . . 99

5.3 Publications . . . 101

Chapter 5

Conclusion

Summary of Chapter 5

This final chapter aims to review the contributions of this thesis, discuss the level to which objectives have been attained and identify how the architecture could be coupled with existing algorithms to improve the resulting agent's behaviour. Finally, we provide a preliminary plan regarding near and distant future research on computational mental simulation, and how we envision building improved agents using our approach.

The aim of this thesis is to propose a generic agent architecture for computational mental simulation that is suitable for complex, dynamic environments in which both physical and behavioural aspects must be taken into account in the agent's anticipation.

This thesis began with an introduction of our goals and objectives and the approach we take in order to achieve them: drawing inspiration from studies in cognitive science on mental simulation. Our focus has been on the human's anticipation ability and capability to adapt while interacting. First, the reader was provided with insight on the dynamic context to which we apply our approach and the efforts that have been made by cognitive science research in studying the mechanisms behind human decision making. We discussed mental simulation and its applicability to intelligent agents, and provided an overview of what can be achieved via mental simulation, therefore presenting our motivation to implement an agent that functions based on this paradigm.

The first result of our research consisted in a survey of computational approaches to mental simulation. Current implementations of the mental simulation paradigm in related works, taken together, computationally address many aspects suggested by cognitive science research. Agents are able to find solutions to non-trivial scenarios in virtual or physical **environments** by treating decision-making as a search problem in the mental simulation space. Existing systems also learn new **behaviour** by imitation of others similar to them and model the behaviour of different others with the help of specialised models, culminating with the collaboration between agents and humans. Approaches that use models of **self** are able to mentally simulate interaction and to learn about their own physical properties. **Multiple mental simulations** are used to find solutions to tasks, for truth maintenance and contradiction detection. However, individual approaches do not cover all of the contexts of mental simulation and they rely on techniques which are only suitable for subsets of obtainable functionality. The main drawbacks of current approaches are the **inflexibility** to context change and the **lack of a generic computational framework** for mental simulation. Our survey spanned over four perspectives on the functionality of state of the art artificial intelligence applications, while linking them to cognitive science research results. Finally, it identified the main gaps in existing literature on computational mental simulation.

Our answer to the shortcomings identified in the current state of the art materialised into a generic mental simulation agent architecture: *"*ORPHEUS*: Reasoning and Prediction with Heterogeneous rEpresentations Using Simulation"*. We provided a detailed conceptual description of our architecture, followed by its implementation in the `C++` programming language. The conceptual description was structured into three main aspects: perception, imaginary world for mental simulation and an abstract world for knowledge representation and learning. We explained how the agent observes the environment and constructs its inner imaginary analogue of the world which functions based on a variety of abstract models. We also discussed how our architecture can

enable an agent to perform mental simulation in virtual and real environments with the purpose of achieving the discussed cognitive functions and managing both environmental and behavioural anticipation simultaneously. Then, the discussion on the implementation delved into greater detail of how the conceptual framework is integrated within the software architecture, resulting in a **generic** approach to building **autonomous** agents capable of **online** learning and **believable** behaviour. Furthermore, in response to the shortage of available software resources in literature, we distributed our agent architecture as **Open Source**[1] along with fully functional demo applications[2], to be used freely in other projects concerning agents with mental simulation capabilities.

Finally we evaluated the **genericness** our agent architecture by instantiating it into three applications, in the order of increasing complexity. First, an agent that solved puzzles in the complex 2-dimensional virtual environment of the "Angry Birds" game. Here, the agent received input from the game in the form of snapshot images of the scene, being required to extract knowledge about the objects composing the scene and take the **appropriate actions** to win the game. Evaluation consisted in three participations in an international competition, resulting in 8$^{\text{th}}$ and 4$^{\text{th}}$ place against other autonomous agents, and 1$^{\text{st}}$ place (best agent) in a "man vs machine" event. Second, we subjected our proposed architecture to an adaptability experiment that included environmental and behavioural aspects inside a dynamic 3-dimensional virtual environment. Here, our agent was able to **adapt** its behaviour based on changes in its environment and exhibited *believable* behaviour such as hiding behind obstacles, without any semantic knowledge of its real counterpart or any strategy. Finally, we increased the complexity of the environment and the agent's embodiment and introduced noisy perception, in the context of the "RoboCup 3D Simulation" competition. Our agent played the role of the goalkeeper of a humanoid robot soccer team, obtaining improved results in comparison to another approach that was used in the competition.

5.1 Discussion

Our research focused on a novel approach to computational mental simulation which extends the current state of the art in this domain. In this section we discuss how our proposed agent architecture improves upon the existing approaches (Subsection 5.1.1), and what matters still remain to be addressed (Subsection 5.1.2).

[1] ORPHEUS source code: https://bitbucket.org/polceanum/orpheus
[2] ORPHEUS demos: https://bitbucket.org/polceanum/orpheus.demos

5.1.1 Contributions of Orpheus

This thesis focused on modelling an agent architecture that is able to encompass the applications of mental simulation in existing computer science literature into an unified framework, while remaining a pertinent analogue through the prism of cognitive science studies on human cognition and brain functions.

Existing approaches to computational mental simulation found in literature are constrained to function in relatively specific setups, with few exceptions which do not provide access to the details of their implementation. There exists no **generic** rely heavily on models created by experts which tend to be difficult to obtain for contexts with higher degree of complexity. Existing implementations are generally used in controlled scenarios and are not designed to be fully **autonomous**. Although approaches exist that function in real time, **online** prediction and learning are used only for specific tasks. The mechanism of the simulation paradigm, such as continuous imagination-reality comparison in complex environments and imaginative behaviour, are not yet fully exploited. Due to either context simplicity or specific functionality, **believability** is only achieved to a relatively low level.

The main contribution of our research – Orpheus – provides an agent with the ability to perform mental simulations in a generic way, based on an analogical representation system (as opposed to fully symbolic or sub-symbolic approaches), along with the mechanisms required to manage, synchronise, learn and apply the actions, emulated within its imaginary world of mental simulations, to its environment.

The **genericness** of Orpheus has been demonstrated by subjecting it to three evaluation scenarios. The first use case of the proposed architecture was a continuous 2-dimensional environment that featured complex physical interactions between objects with time constraints. Second, we have increased the environment complexity to three dimensions, continuous actions (performing a certain action for an arbitrary period of time) and additional environmental and behavioural challenges. Finally, the complexity of the agent's embodiment was increased, with stricter time constraints. The results showed that the cognitive functions provided by Orpheus makes it suitable for complex environments.

In order for an agent to be **autonomous**, it must account for a continuous cycle of perception, cognition and action. One of the main design principles of Orpheus is to integrate these three aspects into the same framework by using analogical mental simulations. We envision cognition within our architecture as an autonomous mentally simulated world of entities that are synchronised with reality through perception, with which the agent's imaginary self can interact in multiple ways, resulting in alternate scenarios that materialise into the agent's real actions.

Making **online predictions** requires high perception frame rate, efficient

memory management and multiple execution threads so that the available time can be used to its maximum. ORPHEUS satisfies all these requirements and also enables mental models to be trained in real time with perception input, allowing the agent that uses it to perform **online learning**.

In this thesis, we assessed the level of **believability** of our agent based on behavioural traits of the animal it impersonated. The results (detailed in the Orphy the Cat application) showed that the agent's behaviour resembled that of its biological counterpart in the real world, without having any knowledge of which animal it was impersonating.

5.1.2 Limitations of Orpheus

Learning: The first, most important aspect that requires further investigation in our approach is applying online learning. Currently, our architecture was only tested with offline learning for predicting the effects of body movements and, in an incipient stage, with online learning of cyclical trajectories. In order for the agents that use ORPHEUS to improve their adaptability and decision making process, more effort is required to address the development of more advanced mental models capable of fast online learning. Additionally, learning must also be performed in the context of controlling the course of actions performed in the agent's mental simulations; *i.e.* using learning algorithms at the level of the action interfaces of mental models in ORPHEUS. This topic has not been addressed in our current approach, but constitutes an important step to obtaining more efficient and adaptable agents.

Believability evaluation: Second, due to its subjectivity, believability is generally difficult to assess in controlled contexts. A "strong" believability test has been proposed by Turing (1950) in his paper "Computing Machinery and Intelligence" but, to date, no machine has passed the original Turing test. However, some variants [Arrabales et al. (2012)] of the Turing test showed that believability can, to some extent, be observed in an objective way. To this end, ORPHEUS requires more extensive evaluation before we can determine that it directly leads to believability in any scenario.

Planning: Third, from the efficiency point of view, more advanced planning techniques should be coupled with our agent architecture. For instance, in our applications, we used a naive random technique to sample the mental simulation state space; our agents' planning abilities could be improved by integrating general purpose search algorithms and/or heuristics. To this end, our approach to computational mental simulation appears to fit well with Monte Carlo Tree Search algorithms which rely on a "simulator" in order to build plans.

5.2 Future Work

Based on the results presented in this thesis, one of the directions in which we have begun to extend our research is towards an evaluation laboratory for studying robot performance and believability (Section 5.2.1). In addition, we intend to extend the capabilities of mental simulation enabled agents by integrating new learning models (Section 5.2.2) and automated planning techniques (Section 5.2.3). Finally, we envision new applications and possible extensions of ORPHEUS, towards agents that are able to create their own goals (Section 5.2.4) and be able to express emotions and affective behaviour (Section 5.2.5).

5.2.1 Nao Soccer Lab

As a result of our collaboration with the RoboCanes[3] research group at University of Miami in Florida, USA, we have built a smaller scale robot laboratory (Figure 5.1) intended for research on goalkeeper behaviour embodied in the physical Nao humanoid robot.

Figure 5.1: Nao Robot Soccer Lab, Centre Européen de Réalité Virtuelle (CERV), Brest, France. **Left**: initial tests; **Right**: goalkeeper evaluation session.

Additionally to the integration of our proposed architecture on the physical robot, our research also includes preliminary evaluations, similar to the approach with the virtual replicas of Nao, aiming to obtain a testing protocol (Figure 5.2) to compare the performance of different behaviours.

Work on believability evaluation of agent behaviours is also set to begin, with a PhD thesis proposal supervised by C. Buche, in the context of CIFRE (Conventions Industrielles de Formation par la REcherche), within the IHSEV

[3]RoboCanes website: http://robocanes.cs.miami.edu/

Figure 5.2: Nao goalkeeper evaluation interface by graduate students at the CERV.

team of the CNRS laboratory Lab-STICC, situated at the European Centre for Virtual Reality (CERV) in Brest, France.

5.2.2 Learning

Our research now focuses on developing additional mental models that learn and are more generic. In contrast to existing approaches, our architecture supports seamless model replacement, therefore the aim is to obtain a repository of lightweight regression models which can be applied to the majority of entities in the environment for the agent to use. Non-linear regression algorithms such as KRLS [Engel et al. (2003)] seem to be well suited for cyclical behaviour prediction in the context of mental simulations, and reinforcement learning is a promising approach to improving our agent's efficiency by informing action sequence generation in the imaginary world. We aim to extend the repository of mental models capable of learning, so that the agents that use our architecture include more contextual information into their decision making process. We have observed in the evaluation scenarios that the quality of decisions is directly influenced by the capability of the mental models to predict the behaviour of observed entities, which was to be expected. Therefore, extending the current learning capabilities (Figure. 5.3) using our framework is a crucial step towards more efficient behaviour of our agents.

The process of building the imaginary world – *i.e.* reconstructing reality in mental simulations – can be interpreted as an *imitation learning* problem. Therefore, we intend to evaluate new mental models that are based on im-

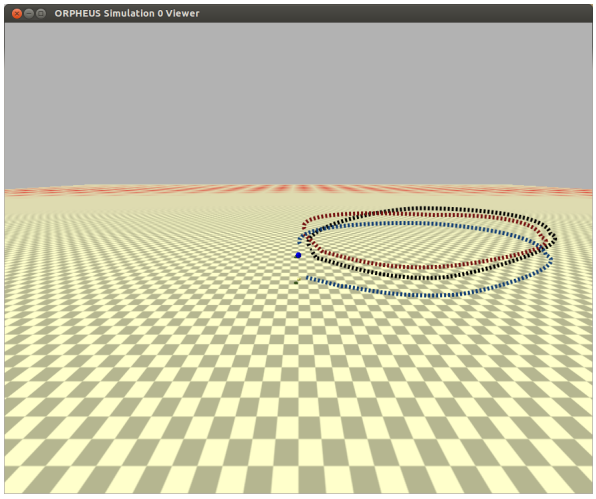


Figure 5.3: Preliminary real time behaviour learning results in the "Orphy the Cat" application presented in this thesis. Traces show mental simulations of the learned trajectory of a bird, after 5 to 10 seconds of observation, in which three KRLS mental models collectively learn different aspects of the bird's behaviour.

itation learning techniques in existing literature [Peters and Schaal (2008); Grollman and Jenkins (2010); Calinon et al. (2012, 2014)]. ORPHEUS allows heterogeneous mental models to be used simultaneously for predictions, and we expect that this capability will be especially useful as the complexity of the agent's environment increases. Imitation learning is particularly interesting due to its stability in the mental simulation context, which other methods lack. By stability, we mean that the output state of one step of an imagined scenario can be reintroduced as input, without having the mental simulation diverge into chaos. Existing imitation models, similarly to instance based learning, exhibit a high level of resilience to the mental simulation divergence problem, and may eventually provide a more robust solution to the creation of the imaginary world.

5.2.3 Planning

Currently, our agents' planning capabilities are limited in terms of the strategy they use to mentally simulate scenarios which are more useful to their goals. To this end, further research will address the integration of heuristic search algorithms such as Monte Carlo Tree Search, with the aim of optimizing the agent's decision making process.

Another particularly interesting research topic is on what we call "hierarchical mental simulation". By this, we refer to exploiting the fact that our architecture supports mental simulations with variable scale of space and time

– *i.e.* small/large scale physical interaction and small/large time steps – with the aim of investigating the advantages of performing mental simulations of an event at multiple scales simultaneously. This could lead to less error-prone predictions using the large scale models, while still benefiting from the detailed interaction models at the small scale. Moreover, we are interested in studying whether larger scale models could be obtained automatically through temporal operations on the small scale models. Additionally, the inverse of this approach may also lead to improvements, as plans could be "sketched" at high level and then be broken down into sub-plans which can be solved through random sampling in mental simulations.

Small scale mental simulations could also enable embodied agents to locally adapt their movements to novel conditions; *i.e.* the agent would possess "low-level creativity". For example, if the scenario is small enough such as touching a button with a finger, it may be feasible to perform a local search in the space of mental simulations to find a solution if there is another object blocking access to the button.

5.2.4 Agent's Goals

In decision making, before searching for a way to achieve a goal, one must have a sense of what the goal is. Most research has been focused on how goals can be achieved, but the problem of creating goals has been relatively neglected [Oettingen et al. (2001)]. A definition of goal may be formulated as a favourable state of an individual and its environment. In theory, such a scenario can be obtained by creating mental simulations of certain elements ("imagination" mechanism) and finding a configuration which is found favourable to the individual. However, there is no objective definition of what is favourable or unfavourable, but most models assume that goals are inputted by human experts after which the agent searches for a solution to achieve the given goal. One may assume that the favourableness of a given state may partly be inherited (through evolution) while the rest can be learned (from previous experience, or from observing others). In the ideal case, an agent would be able to formulate its own goals.

5.2.5 Emotions

Much effort has been put into modelling human emotions into agents to achieve believability. Approaching emotions alone has yielded insight into how facial expressions, body poses and other cues contribute to expressing and understanding emotions. However, from an evolutionary point of view, there should exist a function of emotions [Plutchik (2001)]. Furthermore, in order to express such emotions via various cues, the emotions must exist beforehand. Therefore the emotional system can be interpreted in two interconnected but differentiated parts: the emotion core represented by how the individual ac-

tually feels and the communication of emotions, where the first part precedes the second. The emotion core is also dependent on the characteristics of a given situation, either external or internal.

One may assume that it is a self-regulatory mechanism which identifies the state of the system and produces a response that would trigger other processes to cope with the situation that caused the system to be in the respective state. Moreover, emotions can manifest themselves without direct sensory input from the environment but from mental images, which suggests that the supposed mechanism takes as input the model of the environment, not direct sensory input. The mental image itself is not required to reflect reality to trigger emotional response.

Given this description of emotions, an agent that possesses the capability of building mental images of real or imaginary scenarios could, through a certain mechanism, produce emotions in response to what is happening, has happened or might happen. These emotional responses may afterwards be communicated to others. It is also interesting to note that emotion communication can also function in the absence of emotional states, behaviour which is observable in various types of deception. However, when the emotional content of communication is truthful, the agent may be able to express affective behaviour towards more efficient communication with human users.

We have started preliminary research into the field of affective computing during this thesis, in collaboration with the Affective Social Computing Laboratory[4] at Florida International University, USA, directed by Christine Lætitia Lisetti.

[4]ASCL website: http://ascl.cis.fiu.edu/

5.3 Publications

The research conducted during this thesis generated a number of publications regarding both conceptual and technical results, as following:

Masterjohn J., **Polceanu M.**, Jarrett J., Seekircher A., Buche C., Visser U., *"Regression and Mental Models for Decision Making on Robotic Biped Goalkeepers"*, RoboCup'2015 Symposium, LNCS/LNAI. Springer, 2015. [5]

Polceanu M., Parenthoen M. and Buche C., *"ORPHEUS: Mental Simulation as Support for Decision-Making in a Virtual Agent"*, 28th International Florida Artificial Intelligence Research Society Conference (FLAIRS'15), pages 73-78, AAAI Press, 2015.

Polceanu M. and Buche. C., *"Towards A Theory-Of-Mind-Inspired Generic Decision-Making Framework"*, International Joint Conference on Artificial Intelligence (IJCAI), Symposium on AI in Angry Birds, 2013. [6]

Polceanu M., *"MirrorBot: Using Human-inspired Mirroring Behavior To Pass A Turing Test"*, IEEE Conference on Computational Intelligence in Games, pp. 201–208, 2013.

[5] Joint paper with the RoboCanes team at the University of Miami, FL, USA.
[6] Our AIBirds competition entry obtained 4^{th} place in 2014 and 8^{th} place in 2013.

Bibliography

Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2002). Imitation with ALICE: learning to imitate corresponding actions across dissimilar embodiments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(4):482–496.

Anderson, J. R. (1983). The architecture of cognition.

Anderson, J. R. (1993). *Rules of the mind*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA.

Arrabales, R., Muñoz, J., Ledezma, A., Gutierrez, G., and Sanchis, A. (2012). A Machine Consciousness Approach to the Design of Human-Like Bots. In Hingston, P., editor, *Believable Bots*, pages 171–191. Springer Berlin Heidelberg.

Atkeson, C. G. and Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In *International Conference on Robotics and Automation*, pages 3557–3564. IEEE Press.

Balla, R.-K. and Fern, A. (2009). UCT for tactical assault planning in real-time strategy games. In *IJCAI*, pages 40–45.

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332.

Bergen, B. (2005). Mental Simulation in Spatial Language Processing. In *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*.

Berthoz, A. (1997). *Le sens du mouvement*. O. Jacob.

Berthoz, A. (2003). *La décision*. Odile Jacob.

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science (New York, N.Y.)*, 314(5802):1118–21.

Breazeal, C., Gray, J., and Berlin, M. (2009). An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots. *The International Journal of Robotics Research*, 28(5):656–680.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and autonomous systems*, 6(1):3–15.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):1–43.

Buche, C., Chevaillier, P., Nédélec, A., Parenthoën, M., and Tisseau, J. (2010). Fuzzy cognitive maps for the simulation of individual adaptive behaviors. *Computer Animation and Virtual Worlds*, 21(6):573–587.

Buche, C. and De Loor, P. (2013). Anticipatory behavior in virtual universe, application to a virtual juggler. *Computer Animation and Virtual Worlds (CAVW)*, 24(2):111–125.

Buche, C., Parenthoën, M., and Tisseau, J. (2002). Learning by imitation of behaviors for autonomous agents. In *3rd International Conference on Intelligent Games and Simulation (GAME-ON 2002)*.

Buchsbaum, D., Blumberg, B., Breazeal, C., and Meltzoff, A. N. (2005). A simulation-theory inspired social learning system for interactive characters. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 85–90.

Buckner, R. L. and Carroll, D. C. (2007). Self-projection and the brain. *Trends in Cognitive Sciences*, 11(2):49–57.

Byrne, P., Becker, S., and Burgess, N. (2007). Remembering the past and imagining the future: a neural model of spatial memory and imagery. *Psychological review*, 114(2):340.

Byrne, R. M. (2002). Mental models and counterfactual thoughts about what might have been. *Trends in cognitive sciences*, 6(10):426–431.

Calinon, S., Bruno, D., and Caldwell, D. G. (2014). A task-parameterized probabilistic model with minimal intervention control. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3339–3344. IEEE.

Calinon, S., Li, Z., Alizadeh, T., Tsagarakis, N. G., and Caldwell, D. G. (2012). Statistical dynamical systems for skills acquisition in humanoids. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 323–329. IEEE.

Carruthers, P. (1996). Simulation and self-knowledge: a defence of theory-theory.

Cassimatis, N. L., Trafton, J. G., Bugajska, M. D., and Schultz, A. C. (2004). Integrating cognition, perception and action through mental simulation in robots. *Robotics and Autonomous Systems*, 49(1–2):13–23.

Castillo, E. (1997). *Expert systems and probabilistic network models*. Springer.

Chersi, F., Donnarumma, F., and Pezzulo, G. (2013). Mental imagery in the navigation domain: a computational model of sensory-motor simulation mechanisms. *Adaptive Behavior*.

Churchill, D. and Buro, M. (2013). Portfolio greedy search and simulation for large-scale combat in starcraft. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pages 1–8. IEEE.

Costa, P. and Botelho, L. (2013). Learning by Observation of Agent Software Images. *Journal of Artificial Intelligence Research*.

Curtin, R. R., Cline, J. R., Slagle, N. P., March, W. B., Ram, P., Mehta, N. A., and Gray, A. G. (2013). MLPACK: A scalable C++ machine learning library. *Journal of Machine Learning Research*, 14:801–805.

Dautenhahn, K. and Nehaniv, C. L. (2002). The agent-based perspective on imitation. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in animals and artifacts*, pages 1–40. MIT Press, Cambridge, MA, USA.

Decety, J. and Grèzes, J. (2006). The power of simulation: Imagining one's own and other's behavior. *Brain Research*, 1079(1):4–14.

Dennis C. Turner, P. B. (2000). IV: Predatory behaviour. In *The Domestic Cat: The Biology of Its Behaviour*, pages 152–154. Cambridge University Press.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural computation*, 12(1):219–245.

Dufossé, M., Hugon, M., and Massion, J. (1985). Postural forearm changes induced by predictable in time or voluntary triggered unloading in man. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, 60(2):330–4.

Engel, Y., Mannor, S., and Meir, R. (2003). The kernel recursive least squares algorithm. *IEEE Transactions on Signal Processing*, 52:2275–2285.

Frith, U. and Happé, F. (1999). Theory of Mind and Self-Consciousness: What Is It Like to Be Autistic? *Mind and Language*, 14(1):82–89.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54.

Ge, X., Gould, S., Renz, J., Abeyasinghe, S., Keys, J., Lim, K.-W., Mubashir, Z., Wang, A., and Zhang, P. (2013). Angry Birds Basic Game Playing Software, version 1.22.

Ge, X. and Renz, J. (2013). Representation and Reasoning about General Solid Rectangles. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13), Beijing, China*.

Goldman, A. (2005). *Imitation, Mindreading, and Simulation*, volume 2, pages 79–93. Cambridge: MIT Press.

Gray, J. and Breazeal, C. (2005). Toward Helpful Robot Teammates: A Simulation-Theoretic Approach for Inferring Mental States of Others. In *Proceedings of the AAAI 2005 workshop on Modular Construction of Human-Like Intelligence*.

Grezes, J. and Decety, J. (2001). Functional anatomy of execution, mental simulation, observation, and verb generation of actions: a meta-analysis. *Human brain mapping*, 12(1).

Grollman, D. H. and Jenkins, O. C. (2010). Incremental learning of subtasks from unsegmented demonstration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 261–266. IEEE.

Gross, J., Schmitz, F., Schnitzler, I., Kessler, K., Shapiro, K., Hommel, B., and Schnitzler, A. (2004). Modulation of long-range neural synchrony reflects temporal limitations of visual attention in humans. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, pages 13050–5.

Hanratty, T., Hammell II, R. J., Yen, J., and Fan, X. (2007). Utilizing Concept Maps to Improve Human-Agent Collaboration Within a Recognition-Primed Decision Model. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IAT '07, pages 116–120, Washington, DC, USA. IEEE Computer Society.

Happé, F. (2003). Theory of Mind and the Self. *Annals of the New York Academy of Sciences*, 1001(1):134–144.

Hegarty, M. (2004). Mechanical reasoning by mental simulation. *Trends in cognitive sciences*, 8(6):280–285.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247.

Hsu, F.-H. (2002). *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press.

Kahneman, D. and Tversky, A. (1981). The simulation heuristic. Technical report, DTIC Document.

Kavšek, B. and Lavrač, N. (2006). APRIORI-SD: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583.

Kennedy, W., Bugajska, M., Harrison, A., and Trafton, J. (2009). "Like-Me" Simulation as an Effective and Cognitively Plausible Basis for Social Robotics. *International Journal of Social Robotics*, 1(2):181–194.

Kennedy, W. G., Bugajska, M. D., Adams, W., Schultz, A. C., and Trafton, J. G. (2008). Incorporating mental simulation for a more effective robotic teammate. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*, AAAI'08, pages 1300–1305. AAAI Press.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

Kleiman, D. and Eisenberg, J. (1973). Comparisons of canid and felid social systems from an evolutionary perspective. *Animal Behaviour*, 21(4):637–659.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer.

Konno, H. and Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management science*, 37(5):519–531.

Kormushev, P., Calinon, S., and Caldwell, D. G. (2013). Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, 2(3):122–148.

Kosslyn, S. M. (1980). *Image and mind*. Harvard University Press.

Kunze, L., Dolha, M. E., and Beetz, M. (2011a). Logic programming with simulation-based temporal projection for everyday robot object manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3172–3178.

Kunze, L., Dolha, M. E., Guzman, E., and Beetz, M. (2011b). Simulation-based temporal projection of everyday robot object manipulation. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 107–114. International Foundation for Autonomous Agents and Multiagent Systems.

Laird, J. (2012). *The Soar cognitive architecture*. MIT Press.

Laird, J. E. (2001). It Knows What You're Going To Do: Adding Anticipation to a Quakebot. In *Proceedings of the fifth international conference on Autonomous agents*, pages 385 —- 392.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64.

Langley, P., Laird, J. E., and Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2):141–160.

Markman, K. D., Gavanski, I., Sherman, S. J., and McMullen, M. N. (1993). The mental simulation of better and worse possible worlds. *Journal of Experimental Social Psychology*, 29(1):87–109.

Meltzoff, A. N. (1995). Understanding the intentions of others: Re-enactment of intended acts by 18-month-old children. *Developmental Psychology*, 31(5):838–850.

Meltzoff, A. N. (2002). Imitation as a mechanism of social cognition: Origins of empathy, theory of mind, and the representation of action. In *Blackwell handbook of childhood cognitive development*, pages 6—-25.

Meltzoff, A. N. (2007). 'like me': a foundation for social cognition. *Developmental science*, 10(1):126–134.

Meltzoff, A. N., Prinz, W., and Ebrary, I. (2002). *Elements of a developmental theory of imitation*. Cambridge studies in cognitive perceptual development. Cambridge University Press, Cambridge, {U.K}.

Moulton, S. T. and Kosslyn, S. M. (2009). Imagining predictions: mental imagery as mental emulation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521):1273–80.

Negnevitsky, M. (2005). Rule-based expert systems. In *Artificial Intelligence: A Guide to Intelligent Systems*, pages 25–54. Pearson Education, 3rd edition.

Oettingen, G., Pak, H.-j., and Schnetter, K. (2001). Self-regulation of goal-setting: Turning free fantasies about the future into binding goals. *Journal of Personality and Social Psychology*, 80(5):736–753.

Pearl, J. (2009). *Causality*. Cambridge university press.

Peters, J. and Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697.

Pezzulo, G., Rigoli, F., and Chersi, F. (2013). The mixed instrumental controller: using value of information to combine habitual choice and mental simulation. *Frontiers in psychology*, 4.

Plutchik, R. (2001). The Nature of Emotions. *American Scientist*, 89(4):344–350.

Poli, R. (2010). The many aspects of anticipation. *foresight*, 12(3):7–17.

Premack, D. and Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(04):515–526.

Rao, R. P. N., Shon, A. P., and Meltzoff, A. N. (2004). A Bayesian Model of Imitation in Infants and Robots.

Reynolds, C. W. (1987). Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34.

Risk, N. A. and Szafron, D. (2010). Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 159–166. International Foundation for Autonomous Agents and Multiagent Systems.

Rizzolatti, G. (2005). The Mirror Neuron System and Imitation. *Kodaly Envoy*, 66(5):557–560.

Rochat, P. and Passos-Ferreira, C. (2009). From Imitation to Reciprocation and Mutual Recognition. In Pineda, J. A., editor, *Mirror Neuron Systems*, Contemporary Neuroscience, pages 191–212. Humana Press.

Roese, N. J. (1997). Counterfactual thinking. *Psychological bulletin*, 121(1):133.

Roese, N. J. and Olson, J. M. (1995). *What might have been: The social psychology of counterfactual thinking.* Lawrence Erlbaum Associates, Inc.

Roy, D., Hsiao, K.-y., and Mavridis, N. (2004). Mental imagery for a conversational robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(3):1374–1383.

Rumelhart, D. E. and Norman, D. A. (1983). Representation in memory.

Seed, A. and Byrne, R. (2010). Animal tool-use. *Current Biology*, 20(23):R1032 – R1039.

Shepard, R. N. and Cooper, L. A. (1986). *Mental images and their transformations.* The MIT Press.

Stoecker, J. and Visser, U. (2012). Roboviz: Programmable visualization for simulated soccer. In Röfer, T., Mayer, N., Savage, J., and Saranlı, U., editors, *RoboCup 2011: Robot Soccer World Cup XV*, volume 7416 of *Lecture Notes in Computer Science*, pages 282–293. Springer Berlin Heidelberg.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press.

Svensson, H., Morse, A., and Ziemke, T. (2009). Representation as Internal Simulation : A Minimalistic Robotic Model. In *Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society*, pages 2890–2895. University of Skövde, Incentive level 3.

Taylor, S. E., Pham, L. B., Rivkin, I. D., and Armor, D. A. (1998). Harnessing the imagination: Mental simulation, self-regulation, and coping. *American psychologist*, 53(4):429.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, LIX:433–460.

Ustun, V. and Smith, J. S. (2008). Mental simulation for creating realistic behavior in physical security systems simulation. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 879–885. IEEE.

Van Hoeck, N., Ma, N., Ampe, L., Baetens, K., Vandekerckhove, M., and Van Overwalle, F. (2012). Counterfactual thinking: an fMRI study on changing the past for a better future. *Social cognitive and affective neuroscience*, page nss031.

Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *Evolutionary Computation, IEEE Transactions on*, 11(2):151–180.

Visser, U. and Burkhard, H.-D. (2007). Robocup: 10 years of achievements and future challenges. *AI Magazine*, 28(2):115.

Wilkins, D. E. (2014). *Practical planning: Extending the classical AI planning paradigm*. Morgan Kaufmann.

Williams, D. (2010). Theory of own mind in autism: Evidence of a specific deficit in self-awareness? *Autism : the international journal of research and practice*, 14(5):474–94.