Computational Memory Architectures for Autobiographic and Narrative Virtual Agents

Wan Ching Ho

A thesis submitted in partial fulfilment of the
requirements of the University of Hertfordshire
for the degree of

Doctor of Philosophy

The programme of research was carried out in the School of Computer Science,
Faculty of Engineering and Information Sciences, University of Hertfordshire.

October 2005

Acknowledgements

First, I would like to thank my principal supervisor Prof. Kerstin Dautenhahn and my second supervisor Prof. Chrystopher L. Nehaniv for all their help, advice and friendship. This thesis would not have been possible without their support.

I would also like to thank other members of Adaptive Systems Research Group at University of Hertfordshire who have also helped and inspired me. Particularly to Dr. René te Boekhorst, for his efforts on instructing me to carry out statistical analyses for the experimental data.

My family and Gladys have been both encouraging and supportive in all these years, for which I am forever indebted.

Wan Ching Ho

Abstract

This thesis develops computational memory architectures for autobiographic and narrative virtual agents. Humans and many animals naturally possess a sophisticated memory system for reasoning, learning and also sharing information with others. However it has been a difficult challenge to model the characteristics of such a memory system in the research fields of both Artificial Intelligence and Artificial Life. We propose a framework for enhancing reactive autonomous agents to retrieve meaningful information from their dynamic memories in order to adapt and survive in their environments.

Our approach is inspired by psychology research in human memory and autobiographic memory – through remembering the significance of episodic events that happened in the past, agents with autobiographic memory architectures are capable of reconstructing past events for the purpose of event re-execution and story-telling. The memory architectures that were developed are capable of organizing and filtering significant events which originate in agents' own experiences as well as stories told by other agents.

To validate our memory architectures, both simple and complex Artificial Life type of virtual environments with static as well as dynamic resources distribution were implemented that provide events with different levels of complexity and affect the internal variables of the agents. The performance of various types of agents with different memory control architectures are first compared in single-agent experiments. Each agent's behaviour is observed and analysed quantitatively together with its lifespan and internal states measurements. Group performance with and without communication are measured in experiments with multiple autobiographic agents. Results confirm our research hypothesis that autobiographic memory can prove beneficial – resulting in increases in the lifespan of an autonomous, autobiographic, minimal agent. Furthermore, higher communication frequency brings better group performance for Long-term Autobiographic Memory agents in multi-agent experiments. An interface has been developed to visualise agents' dynamic autobiographic memory to help human observers to understand the underlying memory processes.

This research leads to insights into how bottom-up story-telling and autobiography reconstruction in artificial autonomous agents allow temporally grounded behaviour to emerge. This study therefore results in a contribution to knowledge in Artificial Life and Artificial Intelligence.

Contents

Chapter 1

Introduction

"Intelligence is the cognitive ability of an individual to learn from experience, to reason well, to remember important information, and to cope with the demands of daily living." (Sternberg 1994, pages 395-396)

1.1 Motivation

In achieving the ultimate 'dream' of Artificial Intelligence (AI) – to develop intelligent computer programs (so called 'agents') that can compete with human intelligence – research in this field has identified three essential subjects: memory, reasoning and learning, which cannot be studied independently (Schank 1982). Earlier AI research focused on applying computational memory architectures, building large database systems and designing efficient search methods in order to find solutions which mostly match the current problem. As these kinds of system were intensely knowledge-based, they functioned in limited domains and could only be made more general by increasing the storage space or processing power of the hardware. In recent decades, under the new wave of emphasizing the concept of embodiment (so called Embodied AI), researchers started looking at interaction between agents/robots and their virtual/physical environments. Thus 'Nouvelle AI' approaches like neural networks, genetic algorithms, behaviour-based architectures, etc., create a new set of intelligent behaviours for agents/robots in order to enable them to adapt quickly to their environments.

Inspired by psychology research in human memory, which stated that autobiographic memory is a specific kind of episodic memory that may develop in childhood (Nelson 1993), Dautenhahn introduced autobiographic agents, which are agents embodied and situated in a particular environment (including other agents), and which dynamically reconstruct their individual history (autobiography) during their lifetimes (Dautenhahn 1996). Since autobiographic memory particularly focuses on meaningful and significant events for the intelligent agents, it also can be used for:

- Synthesising agents that can behave adaptively (Nehaniv and Dautenhahn 1998a) and in socially intelligent ways (Dautenhahn 1999a).
- Designing agents that appear believable and acceptable to humans.

Furthermore, from the human observer perspective, agents' behaviours are required to be more coherent so that the change from one behaviour to another can be more understandable (Sengers 2003). This coherence which originated with the life story of human beings (Linde 1993) is particularly important for believable agents in narrative story-telling environment.

The research in narrative intelligence aims to develop agents which can have the capacities of story-awareness, story-telling and historical grounding. Concerned with building this kind of narrative agent, the area has been investigated in various directions, such as interactive drama or story-telling (Mateas 1999, Mateas and Stern 2002, Stern 2003, Cavazza, Martin, Charles, Mead and Marichal 2003), social understanding (Dautenhahn and Nehaniv 1998, Dautenhahn 2002, Dautenhahn 2003), autonomous camera agents (Hornung, Lakemeyer and Trogemann 2003) and narrative in virtual environments (Aylett 1999). Researchers have brought fruitful ideas to enhance both story-telling abilities and believability of narrative agents interacting with human users.

The potential of autobiographic agents to create stories from their own experiences and understand stories from others enhances the endurance of events remembered in their autobiographic memory. Nelson (1993) pointed out that in addition to the function of language, humans sharing memories with other people can be seen as narrative story-telling that performs a significant social-cultural function, and both these two functions explain why personal autobiographic memories continue to persist during their lifetime. Autobiographic memory has also been studied as historical grounding for artifacts (Nehaniv 1999), in which recognising, expressing and having a narrative structure are essential in freeing agents from social as well as temporal isolation in interaction with humans or in mediating human-human interaction over networked media.

New web-based technologies for 3D graphics such as VRML, X3D and Java3D provide tools for creating virtually embodied agents and dynamic environments for the need of simulations. There are also some widely applied agent/robot control architectures in the robotic research field, such as the subsumption control architecture (Brooks 1986), that provide fundamentally useful bases for virtual agents to build new extensions of new behaviour, memories and internal states such as some internal and emotional states. Moreover, various memory architectures have been developed in the fields of Cognitive Science, Robotics and Artificial Intelligence, usually for complex tasks such as navigation or case-based problem solving. In contrast to these tasks, our research goal focuses on bottom-up Artificial Life principles, in which we study how behavioural complexity emerges for autobiographic agents that reconstruct and employ their lifetime memories for acting in dynamic virtual environments, interacting with other agents and sharing experience and memories via story telling. This conceptual background is fleshed out in more detail in Nehanive and Dautenhahn (1998a).

1.2 Challenges

Developing computational memory architectures for narrative autobiographic agents in a bottom-up Artificial Life fashion involves both theoretical and technical knowledge from various fields. It is essentially different from building a classical AI expert system which normally contains a sophisticated database and applies techniques like Case-Based Reasoning (Kolodner 1993) for tackling problems by remembering new cases, reasoning and retrieving appropriate cases in the specific domain. The main issue in this thesis is to design and verify the capabilities of autobiographic agents in:

- Remembering significant events experienced during their lifetime for adaptation to the dynamic environments.
- Studying the emergence of a bottom-up narrative structure to understand, reconstruct and tell stories to other agents.

We also expect that sharing stories in a narrative sense for autobiographic agents can result in better 'story qualities' and group adaptation, we expect that in a group of story-telling agents higher lifespans will be observed.

1.2.1 Conventional Memory and Learning Architectures

To create a computational memory architecture for intelligent programs for problem-solving, a huge amount of data needs to be input manually in advance, and then reasoning rules for this database-like memory architecture are incrementally added by a programmer. Some intelligent programs are also capable of remembering new cases and figuring out new rules through new training cases. However, reasoning and learning performed by these programs are still mainly passive, since retrieving the solution is based on the similarity of the stored cases. For example, user input or programmer adjustments are largely required to either rectify the right weight for correct output or remove redundant cases (Leake 1996).

Biologically inspired and evolutionary algorithms, such as neural networks and genetic algorithms, are commonly applied as the learning and adaptation functions for intelligent agents and robots. These algorithms can eventually get what they were supposed to learn, but without a relatively large amount of training data for neural networks or long repeated generations for genetic algorithms, learning is difficult to achieve. Compared to these learning algorithms, the advantage of learning through remembering episodic information is speed. As in some cases, only one observation is needed to be carried out to learn a new task; this is commonly applied to the design of believable characters in a few modern commercial computer games (Cass 2002, Isla and Blumberg 2002).

Research in believable virtual agents has also utilized the human cognitive memory model (Norman and Bobrow 1975) from the field of cognitive science and psychology, such as the study in synthetic vision for autonomous virtual humans (Peters and O'Sullivan 2002). Strategies for memory storage are usually divided into Sensory, Short-term and Long-term memories according to the rehearsal process and the retaining length of time for an item to be remembered. However, imposing a human memory model on virtual agents without taking its real cognitive abilities into account would meet various limitations, for example, only simple items can be remembered, agents are not be able to decide which item is more important to itself or to other agents, and it's hard to create temporal sequences of items in the memory.

1.2.2 Narrative Story-telling Agents

Narrative agents are usually pre-programmed either with temporal and structured stories or with simultaneously selecting story sequences from a large story database when they are interacting with human users in story-telling systems. Therefore they can take advantage of mechanisms used in natural historically grounded systems (Nehaniv 1999) and enhance the friendliness of these systems. While appreciating the success narrative agents bring to story-telling systems or software interfaces, the investigation of how agents themselves can benefit from bottom-up narrative intelligence in the sense of basic survival is missing in the research field. On the other hand, rather than having system evaluations from purely users' perspectives and comments, this thesis presents experiments with quantitative results for studying the emergence of a bottom-up narrative structure to understand, reconstruct and tell stories to other agents.

Humans are naturally experts in narrative as we can even express what we want without using natural language but just by showing appropriate gestures or facial expressions. In contrast, it is very difficult for virtual agents to have the same quality of result. This leads to the problem of behavioural incoherence – human observers can not understand why agents behave in such a way. This phenomenon is described by Sengers as 'schizophrenia' (Sengers 2003). Therefore, narrative virtual agents, especially if they are non-verbal, should be able to show their goals (e.g. what are they going to next) explicitly to human observers, e.g. as realized in this thesis, an interface to show the contents of dynamically reconstructed stories from agents' memory.

1.3 Methodology

In this thesis, we use the term 'agent' as defined by (Franklin and Graesser 1997): "An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

1.3.1 Criteria and Goals

In light of the preceding discussion, we seek an approach to develop intelligent autonomous agents that are capable of remembering significant events, reconstructing events they previously experienced via interacting with their environments and with other agents via narrative story-telling and story-understanding. The desired properties of this approach are as follows:

- Narrative autobiographic agents should be able to remember significant events which contain meaningful information for their surviving in their dynamic memory, and to share this meaningful information through story-telling to other agents.

- As test-beds for running experiments, various types of virtual environment are required in order to have different levels of dynamics and complexities for creating different events for agents to experience.

- Systems should provide different types of interface to show the coherence of agents' behaviours, and the dynamically changing contents of agents' memory as well as stories received from other agents.

The research reported in this thesis contains:

1. Developing several autobiographic memory control architectures, including narrative communicative features in a bottom-up fashion.

2. Validating their performance by running dedicated experiments in various virtual environments with different levels of dynamics and complexity.

1.3.2 Autonomous Artificial Life Agents

We believe that the best way to achieve our goals in the long run is to create autonomous agents which are embodied and situated in the environments by using reactive control architectures. Reactive control architectures provide:

1. Controls with distributed components such as sensors and actuators that react instantly to the environment.

2. Flexibility which allows new components to be added on top of the existing components for creating robust agent behaviour patterns.

In our research, we defined different kinds of internal variables and developed computational memory models as extra components to arbitrate the basic behaviours offered by the reactive architecture, such as wandering and avoiding behaviours.

1.3.3 Narrative Autobiographic Memory Control Architectures

From a bottom-up perspective in developing computational memory architectures for artificial life agents, we start from building a simple and linear memory on the top of a reactive control architecture. After carrying out different types of experiments with this memory architecture in a static environment, we incrementally increase the complexity of the design in order to create a more adaptive architecture, and then we validate this architecture by experimentation in a relatively complex and dynamic environment.

1.3.4 Simulated Virtual Environments

We take the environment design into account seriously because we believe that, as the test-bed for our memory control architectures, the environment is crucial for studying the adaptive behaviours of an embodied system. Therefore we create various virtual environments from simple and static to complex and dynamic for measuring the performances of relevant architectures in both directions, remembering significant events and narrative story-telling features, for our agents. The reason to ultimately have a nature-like virtual environment with dynamic resources distribution is that it provides a rich variety of events for our narrative autobiographic agents to experience, so agents can have various types of events to remember and they have to figure out which one of these events is the most significant one to re-execute or to tell other agents.

1.3.5 Understanding Coherent behaviours

To assist the human observers in understanding agents' behaviours and memory contents which are dynamically changing, we developed Observer Interfaces for different types of memory architecture. The interface for the long-term autobiographic memory architecture is able to show all the candidate events and the target event to be eventually re-executed from the agents' Long-term Autobiographic Memory schemata. Moreover, for the latest architecture which supports narrative storytelling as well as story-understanding, its Observer Interface shows also agents' memory contents which could be partly provided from other agents' stories and mixed into events experienced by themselves.

1.3.6 Disclaimer

As we study virtually embodied Artificial Life agents, some metaphorical terms can be found in this thesis to describe various conditions of our agents, such as 'hungry'. However, using terms does not imply that our agents have a real biological embodiment.

1.4 Contributions and Results

The overall aim of this research is to design an appropriate computational memory architecture for narrative autobiographic agents. This architecture should be able to show the ability of agents reconstructing their own autobiography through experiencing different events and telling as well as accepting stories during the interaction with other agents in the environment.

Through enhancing the performance and the adaptation of a purely reactive agent to survive in its environment, we have progressively developed various computational autobiographic memories as the crucial components embedded into the basic architecture for intelligent agents to broaden their historical grounding and allow a narrative story-telling structure to emerge. We focus on building autobiographic agents which are able to realize 'meaningful' information and reconstruct stories from their own as well as other agents' experiences. These features developed in a bottom-up fashion and from the Artificial Life perspective address problems from:

1. AI systems which are intensely rule-based and lack the capability to determine the significance of information.

2. Learning systems that depend on off-line training data or long reiterated evolutionary or training phases.

The study has been extended beyond spatial problems by emphasizing noncommutative and irreversible sequences of events. Moreover, the Observer Interface developed in this research helps to solve problems arising from agent behaviours that are incomprehensible to human observers.

Advances originating from this study should therefore contribute to knowledge in Artificial Life and Artificial Intelligence.

1.5 Thesis Overview

The thesis is organized as follows:

In Chapter 2 we review previous work primarily in the research fields of human memory and narrative intelligence. Since research in human memory has been carried out for many decades, in this chapter details of different types of human memory and knowledge representation models from various research fields, such as psychology, cognitive science and computer science are illustrated and discussed. Research issues related to applying and modeling memory are also introduced, which include research in:

1. How memory and communication influence animals' foraging behaviour.

2. How the design of characters in computer games benefits from episodic information and social communication.

3. How our approach is different from other approaches to memory and learning in AI.

With regard to narrative intelligence, narrative structures and the criteria for designing narrative agents are discussed. Another main focus of this chapter is the background research of autobiographic memory and its consideration as historical grounding for agent design.

In Chapters 3 and 4 we describe our early research in developing and investigating autobiographic agents for both single-agent and narrative multi-agent experiments in static virtual environments. We have pursued a bottom-up approach from an Artificial Life perspective, and implemented finite-state autobiographic memory into the basic control architecture for purely reactive agents. The mechanisms that autobiographic agents use for remembering their previous action sequences for going back to the particular resources and sharing experiences with other agents are illustrated in detail. The result that autobiographic agents outperform the reactive agents in surviving in the static environments for both single agent and multi-agent experiments is discussed.

Chapters 5 and 6, which form the main body of this research illustrate the essential motivations, design and implementations for long-term autobiographic agents. We show the enhanced design for both narrative autobiographic agents and the dynamic virtual environment. Through experimental results in single-agent experiments, the varied performances of Purely Reactive, Short-term Memory and Long-term Autobiographic Memory agents indicate how agents' adaptivity is influenced by their memory architectures and mechanisms in reconstructing events. Furthermore, the advantages of having narrative story-telling features for autobiographic agents are demonstrated in multi-agent experiments.

In Chapter 7, we describe how we achieve agent behaviours that are more understandable and coherent to observers, by developing interfaces showing the history of agents' internal states, memory contents and reconstructed events.

In Chapter 8, we firstly carry out an in-depth discussion about the design of autobiographic memory control architectures and experimental results. Then we review the contributions of the thesis, conclude the key points of this research and list the possible directions of future research.

Chapter 2

Background

This chapter provides the general background of different areas of our interdisciplinary research. We first review research of human memory in the fields of psychology and cognitive science, then we turn to focus on the characteristics of autobiographic memory and the pioneering work of applying it to agent research. Following that we review the theoretical framework of the role autobiographic memory plays in histories and algebras of time, and the research about how memory influences animals' foraging behaviours. Next we investigate various research work in the field of narrative intelligence. We also study how episodic information and social communication are applied in the design of non-player characters in modern computer games. Then we compare our approach with other approaches to memory and learning and AI. Finally we complete this chapter by putting our research work in perspective relative to prior research on memory and narrative.

2.1 Knowledge Representation and Autobiographic Memory

The way humans and computer programs represent knowledge has been an interesting and important issue in many research areas, such as philosophy, cognitive science, psychology, linguistics and AI. Human natural language is an example of how to represent knowledge in low-level primitives and organize knowledge in high-level structures: a word is the basic unit and a sentence is the basic structure; nevertheless higher-level structures are also available, if needed, for classifying and organizing knowledge, such as paragraphs, sections, chapters, and volumes (Sowa 1999).

In the 1920s, psychologist F. C. Bartlett pointed out a form of mental representation to explain how humans deal with complex structural knowledge – *memory schema* (Bartlett 1932); the theoretical construct of this idea has a strong impact on contemporary theories of knowledge representation in cognitive psychology and cognitive science. Systems in earlier classical AI tend to represent knowledge in a machine-understandable and symbolically-grounded way for computers to perform efficient searches for problem-solving. Embodied AI emphasizes the interaction between agents and their environments, in which the representation of the same object can be very different and lead to different 'meanings' existing in agents' minds. Researchers suggest that these differences in knowledge representation among agents can be negotiated via social interaction (Dautenhahn and Christaller 1996) and social learning (Steels 2003).

This section introduces the previous research on knowledge representations in the memory of both humans and artificial intelligent agents. Since we are particularly interested in the structure of how meaningful episodic events are organized in autobiographic memory, we illustrate and comprehensively discuss one of the autobiographic memory models from human memory research in psychology. This model is then applied in our computational autobiographic memory models.

2.1.1 Interdisciplinary Memory Research

The mysterious capacity of human memory in encoding, storing and retrieving information has gained a vast amount of attention from researchers in different research fields. Apart from focusing on the structures of knowledge representation in human memory – contemporary schema theories, such as Frames (Minsky 1975) and Scripts (Schank and Abelson 1977), whose core components derive from aspects of Bartlett's theory – psychologists and cognitive scientists have also investigated other phenomena of human memory, such as Mental Models (Johnson-Laird 1980), the general stage processing theory (Atkinson and Shiffrin 1968), the length of items in memory retention and forgetting effects.

Memory Processing Theory in Cognitive Science

In the domain of cognitive science, memory processing theory (Atkinson and Shiffrin 1968) is widely recognized as a general approach which views learning as a group of active mental processes for acquiring and remembering knowledge. Although the details of this theory vary from author to author in different studies, the main stages (sensory register, short-term memory and long-term memory) consistently exist in different models. Extracted from Norman and Bobrow (1975), Figure 2.1 illustrates the human memory processing theory in a traditional point of view. It essentially indicates that a stimulus (described as "Physical signals") will be stored permanently

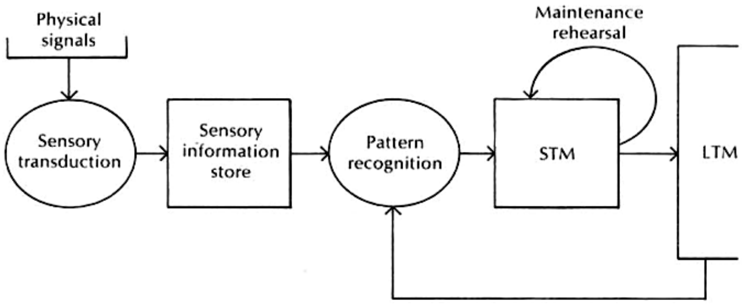

Figure 2.1: The traditional, linear stage theory of memory processing, taken from Norman and Bobrow (1975, page 117). Notably details of the system vary from author to author.

in the long-term memory after it gets captured in the sensory information store and rehearsed several times in the short-term memory. The functionality provided by each type of memory in the memory processing theory figure can be described as follows:

- Sensory information store is also called sensory register. It is a system of receptors which holds sensory information for a very brief period. For example, after we have seen an object, we are able to retain the exact image of this object; however, if we have no use for this retention and we may also shift our attention to other objects, the image of this object can be temporarily held in the sensory information store for a split second. Therefore, only the stimuli we want to remember are moved into short-term memory.

- Short-term memory is also known as Working Memory. It is the information from the the sensory information store we are focusing on at any given time. Due to the limited capacity of short-term memory, approximately only seven items can be kept at once in short-term memory and the length of holding each item is about 20 seconds without maintenance rehearsal (Miller 1956).

- Long-term memory is the permanent store of information, and is virtually unlimited in capacity. However, the length of time taken for recalling different information in long-term memory is very different, as the cue of recalling lies in the familiarity with that information as well as the method of retrieval.

In recent years, some AI research has adopted memory processing theory in developing various components for intelligent agents. One example is in modeling synthetic vision (Peters and O'Sullivan 2002). Various stages derived from memory processing theory are applied not only for storing objects seen by the agent in the virtual environment, but also for filtering the information from the sensor input and determining the length of time for an item to be remembered. Therefore, goals which indicate the focus on specifically interesting objects may be generated for the virtual agents; otherwise a typical synthetic vision strategy 'Proximity Sensing' is applied by the agent to categorize different types of objects in the virtual environment.

Memory Models in Psychology

Psychological studies in human memory for both remembering personal events and creating coherent life stories (Linde 1993) are well established since Bartlett's enlightenment in the research of memory schema (Bartlett 1932). He rejected the thought of considering long-term memory as a warehouse of static and unchanged memories. Instead he showed memory should be often reconstructed based upon world knowledge and schemata by introducing various types of innovative experiments on remembering, such as Repeated Reproduction and Serial Reproduction, in which the same story was recalled on more than one occasion by an individual or a group of subjects. Therefore, Bartlett's work has emphasized the reconstructive view of long-term memory. He proposed that 1) human memory has substantial amount of generic knowledge in the form of unconscious mental structures (schemata), such as rules or scripts that can be used to interpret the world, 2) new information is remembered according to how it adapts into these rules and, therefore, 3) errors in recall may occur when the existing schemata interact with the new information.

A Systematic View of Schema Theory In recent decades, researchers like Alba and Hasher (1983) have investigated systematically the detailed characteristics of schematic memory. They proposed a prototypical schema theory of memory for evaluating four central encoding processes of the schema theory. These four encoding processes include: "selection – a process that chooses only some of all incoming stimuli for representation; abstraction – a process that stores the meaning of a message without reference to the original syntactic and lexical content; interpretation – a process by which relevant prior knowledge is generated to aid comprehension; and integration – a process by which a single, holistic memory representation is formed from the products of the previous three operations" (page 203). They have also found out an extra result during the evaluation of their research, under the assumption of schematic memory, "some details are stored no matter what the extent of a person's prior knowledge and no matter whether that knowledge is activated at encoding" (page 225); therefore they claimed that storing records in human memory is far more detailed than the prototypical schema theories implies.

Different Types of Knowledge in Long-term Memory From a clinical and physiological standpoint, many observations those of such as Tulving (1972) and Cohen and Squire (1980) suggest that there are various sub-categories of long-term

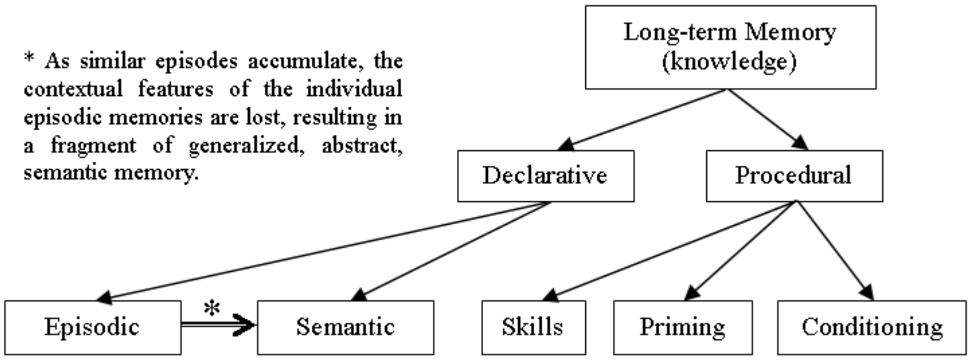

Figure 2.2: Different types of memories exist in human long-term memory, known as classification of knowledge, modified after Tulving (1972, page 382).

memory, as illustrated in Figure 2.2. Autobiographic memory which contains meaningful episodes is not explicitly illustrated in this figure, it exists in the area of episodic memory. Notably the meaningfulness of an episode depends on the pre-existing collection of semantic knowledge.

Human Information-Processing System Based on the memory processing theory in cognitive science, Norman and Bobrow (1975) proposed a human information-processing system from a perspective of schematic memory in order to form a meaningful interpretation of the world for cognitive systems, as illustrated in Figure 2.3. The vast collection of structural memory schemata created by past experience are stored in a long-term storage which can be seen as the combination of short-term memory and long-term memory from the model of memory processing theory. They described how these schemata can be used to actively characterize the declarative knowledge of any experience.

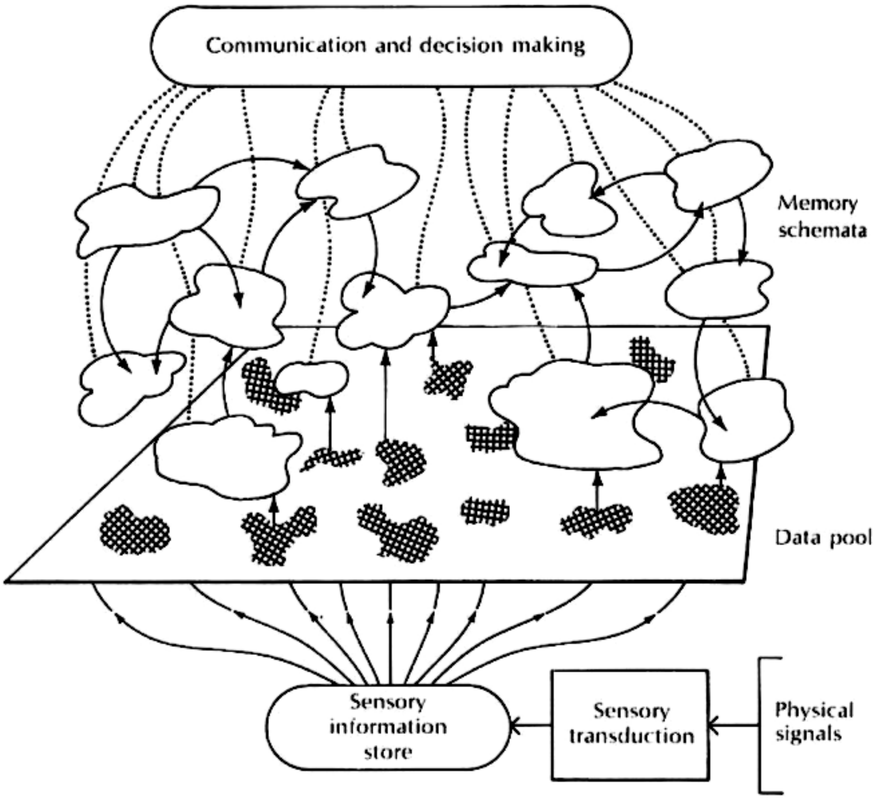

Figure 2.3: The memory schemata view of the human information-processing system, taken from Norman and Bobrow (1975 page 118).

Knowledge Representation in Intelligent Programs

Frames As introduced by computer scientist Minsky (1975), the term *frames* is essentially synonymous with schema theory, except frames have been used as both psychological constructs and as a pioneering model of knowledge representation in artificial intelligence. Minsky was attempting to develop computer programs that would show human-like intelligence, and he decided to build a concrete model as a realization of the theoretical schema theory from Bartlett. In addition to representing generic knowledge, the theory of frames also deals with specific schema-related information which the highly abstracted knowledge representation model derived from schema theory cannot cope with – by having fixed frames and putting default values into empty frame slots for representing a particular instance in the world. For example, to represent a class room in a frame, slots in this frame are filled with default values if no information from the environment can be gathered, such as a white ceiling, fluorescent lights, chairs and tables.

Scripts Schank and Abelson (1977) proposed *scripts* for intelligent computer programs to understand *routine events*. A script which contains general information about a particular, frequently experienced event is similar to a frame. For example, visiting a restaurant has general steps like being seated, ordering the meal and so on. In addition to having these basic features like human episodic memory, scripts are also personally unique and dynamically updated with new experience (Schank 1982).

Based on the research of scripts, Schank (1982) also studied psychologically why humans are reminded of an old experience by a new one and he proposed a model of memory, in which new experiences are stored only if they fail to conform to one's expectations. Moreover, scripts have been investigated for guiding behaviours since they allow humans to make inferences, and to reconstruct incomplete stories as people often recall information that was not in the original story, but was consistent with the script (Bower, Black and Turner 1979).

Case-based Reasoning (CBR) Grounded in the theory of dynamic memory (Schank 1982), case-based reasoning is an AI technique for processing empirical knowledge and a CBR system reuses one or several memorized cases for problem-solving (Kolodner 1993). A case is a set of empirical data and empirical knowledge is knowledge learnt through experience or practice. Nowadays different types of CBR systems have been widely used in software design, medical diagnosis, argumentation in law domains, etc. Typical CBR systems have the following steps in solving problems or dealing with new cases:

1. Abstraction – Converting the input representation of a new case, such as from natural language to an internal representation of the CBR system.

2. Retrieval – Searching through the memory for similar cases and then selecting the sub-set of these cases that will be effectively reused.

3. Reuse – Adapting the cases in problem-solving.

4. Revision – After trying the proposed solution, correcting mistakes if there are any.

5. Memorization – Adding the solved problem as a new case to the memory.

2.1.2 Autobiographic Memory

> *"In the past, it (autobiographic memory) has usually been conceived of in terms of childhood (or infantile) amnesia, the phenomenon, first identified by Freud (1963) and familiar to all who reflect on it, that memories for events from the early years of our lives – before about 3 to 4 years – are not available to adult consciousness, although many memories from later childhood usually are easily called up."* (Nelson 1993, page 8)

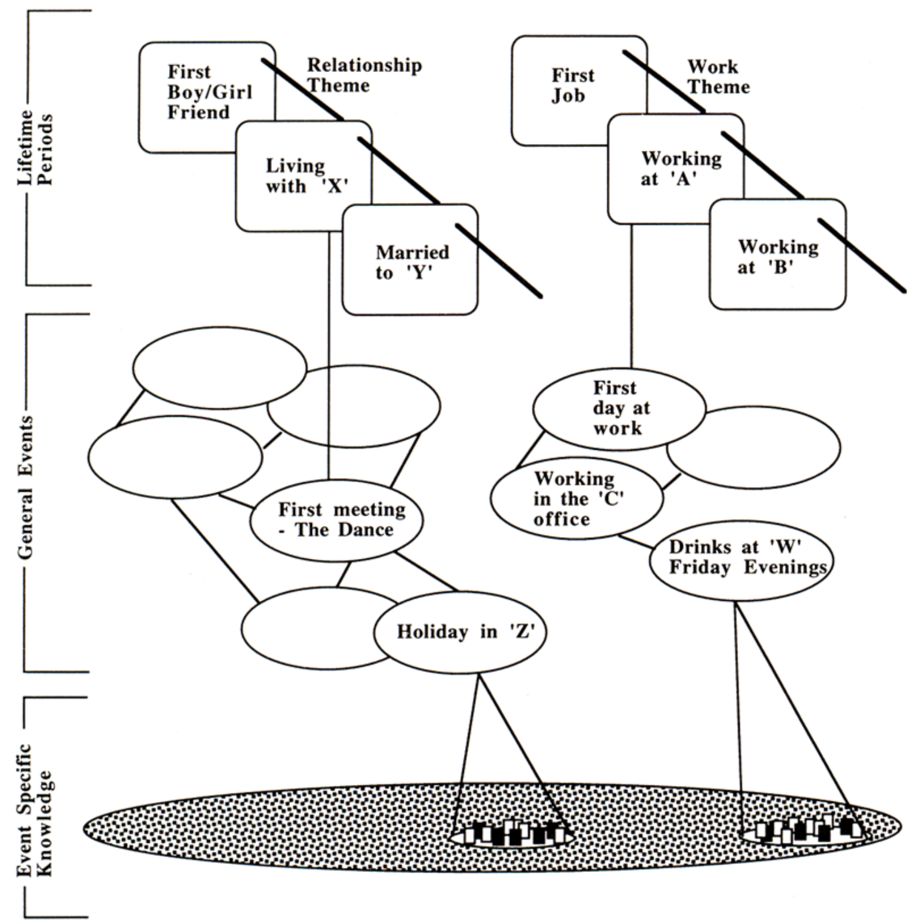

Figure 2.4: Hierarchical knowledge structures in the autobiographic knowledge base, taken from Conway (1996 page 68).

Autobiographic Memory Research in Psychology

Autobiographic memory is a specific kind of episodic memory (Nelson 1993) and contains *significant* and *meaningful* personal experience for a human being. Two features of autobiographic memory are generally defined and accepted by researchers in psychology, as pointed out by Conway, Pleydell-Pearce and Whitecross (2001)):

- Autobiographical memories are mental constructions of the self.

- They very often feature imagery while simultaneously containing abstract personal knowledge (Conway 1990, Conway 1996, Conway and Pleydell-Pearce 2000).

Figure 2.4 extracted from Conway (1996) shows the hierarchical knowledge structures in the autobiographical knowledge base, in which Conway indicated that lifetime periods may themselves be thematically linked together and he showed a work theme and a relationship theme in his past as examples in this figure.

From the perspective of considering humans as social beings, Nelson (1993) in developmental psychology carried out her investigations on how children develop their own autobiographic memory, and she suggested that the primary function of autobiographic memory is sharing memory with other people. Other studies in psychology and cognitive development have also pointed out the cognitive, as well as social, function of autobiographic memory underlying all of human story telling and history-making narrative activities. Kelly and Dickinson studied the autobiographical accounts of the experience of chronic illness and suggested that " 'self' is not a biologistic or psychologistic thing, rather self is autobiographical narrative." (Kelly and Dickinson 1997, page 224).

Neisser (1986) suggested that recalling an experienced event from autobiographic memory is a matter not of reviving a single record but of moving appropriately among nested levels of structure. As the rehearsal process takes place, the memory structure is fundamentally changed by this recall. Moreover, a recall is always reconstructing an event by some other similar events of experience; particularly goals and current situations of the recall do affect how much details can be obtained from the event which is remembered (Neisser 1986).

Self-schemata for maintaining the integrity and gist of past life events have been discussed by Barclay (1986). He argued that events in autobiographic memories change over time since new events occur and many life experiences become repetitious, making any single event indistinguishable from related happenings. Thus inaccurate remembering and forgetting events from autobiographic memory take place because of the merging of episodic memories into more generic event categories representing the semantic features of everyday activities.

Applying Autobiographic Memory in Agent Research

The theoretical term *autobiographic agents* was first defined by Dautenhahn: "we define the concept of an autobiographic agent as an embodied agent which dynamically reconstructs its individual 'history' (autobiography) during its life-time." (Dautenhahn 1996, page 5). Nehaniv and Dautenhahn (1998a) then later introduced the notion of *historical grounding* derived from autobiographic memory. As described in the next sub-section, they stated that historical grounding of autobiographical agents helps developing individualized social relationships and forms of communication, which are characteristic of social intelligence, and it may also lead to more appealing and 'human-like' engaging interactions with artificial agents, making them more pleasant and acceptable to humans.

Dautenhahn and Coles (2001) carried out the first set of pioneering experiments on autobiographic agents, in which Situation-Action-Situation triplets were used as the core of the agents' autobiography. This work compared trajectories and lifetimes of purely reactive (sensory-driven) and post-reactive (memory-driven) control agents, results from the experiments showed that autobiographic memory which is embedded in the control architecture can effectively extend an agent's lifetime.

2.1.3 Histories and Algebras of Time

A feature of memory and remembering is that they provide 'extrasensory' meaningful information by which an agent may modulate or guide its immediate or future behaviour, as indicated by Nehaniv and Dautenhahn (1998a). Nehaniv explains this concept of *temporal horizon* by citing Heidegger (1972) who "saw the state of man as being as situated in the Now, being there in the imminence of the Future in relation to the impinging Past (Nehaniv 1999, page 1). This temporal horizon can allow for planning for future actions and story telling about past or imagined events. The vast temporal horizon means that humans will tend to deal with interaction in a way that makes narrative sense.

Nehaniv and Dautenhahn have also investigated applying global semigroup theory to the design of autobiographic agents (Nehaniv and Dautenhahn 1998b), they stated that structuring historical memories into algebras allows one to construct expressions in 'algebras of time' that can support recording events of fundamental significance to an agent. "Expansions are systematic treatments of histories in algebras of time: recording histories of various kinds can be used to systematically 'expand' algebras of time into larger ones" (Nehaniv, Dautenhahn and Loomes 1999, page 8), this corresponds to an expansion of the temporal horizon. By using an expansion rather than the original algebra of time, which describes only the moment-to-moment transitions of the system, it is possible to express formal stories or histories.

2.1.4 Memory and Communication in Animal Foraging behaviours

In the perspective of ethology, researchers have investigated widely how different types of memory possessed by animals influence their foraging behaviours. Due to space constraints, only a few examples are discussed in this section. Shafir and Roughgarden (1996) studied the foraging behaviour of a lizard, and they found out that measuring the length of time which it takes an animal to begin stabilizing its behaviour may be an effective technique for determining the length of memory. In investigating the existence of declarative and episodic-like memory in animals (Clayton, Griffiths and Dickinson 2000), researchers presented experimental results that demonstrate that some species of birds can perform a food-caching and recovery memory task that depends on episodic-like memory, a type of memory recall that closely resembles episodic memory.

On the other hand, it is increasingly acknowledged that information sharing may improve the performance of animals in a sociobiological point of view. Foraging activities in birds using personal experience and public information are studied by Templeton and Giraldeau (1996). They discovered that when a captive starling could watch another's sampling activities in the experiment, it apparently performed better by adopting the public information to access the patches with foods. Similar collective intelligence is also observable in social insects, such as ants, bees, termites and wasps. Researchers in the field of swarm intelligence (Bonabeau, Dorigo and Theraulaz 1999) have discovered that social insects' behaviours can be viewed as powerful problem-solving systems. They found out that social insects' sophisticated collective intelligence lies in the networks of interactions among individuals and between individuals and the environment, such as *stigmergy* which is a indirect interaction between individual insects. One example of stigmergy is that two individuals (e.g. termites) interact indirectly when one of them modifies the environment (e.g. building their nest by carrying soil pellets from an individual termite) and the other responds to the new environment at a later time (Bonabeau et al. 1999). Compared with our research in developing narrative autobiographic agents which reconstruct stories from other agents or self experiences by applying own memory schemata, this kind of indirect interaction in social insects can be seen as producing explicit 'stories' by modifying the environment individually – the subtle modification of the environment will be interpreted by others as a sign for guiding the cooperative tasks existing among them.

2.2 Narrative Intelligence

In recent years, much high quality and interdisciplinary research can be found in the field of Narrative Intelligence, which established the connection between the research fields of Narrative and Artificial Intelligence, as briefly introduced in Chapter 1. Mateas and Stern have given a elaboration of the birth of narrative intelligence, in which they stated that "The time is ripe for AI to reengage narrative, to explore all the ways in which narrative intersects with intelligence of both the artificial and human varieties" (Mateas and Stern 2002, page 4). Notably researchers in these areas have been generally and mostly interested in building agents which interact with human users with the purpose of relating the agents' stories to users' own lived experience.

As introduced previously in sub-section 2.1.3, the early research from Dautenhahn and Nehaniv (1998) suggested that artificial life agents may be able to construct historical grounding as their own autobiography to understand both themselves and each other, since telling (part of) a plausible autobiographical story to others is more than showing a plausible sequence of episodic events. It includes the construction of a plausible story based on one's goals, intentions and motivations (Dautenhahn 2003). Nehaniv has also pointed out that narratives might be considered stories about the self for an agent and served as a basis for memory, and transmission of such a formal narrative to another agent may be meaningful for the second agent and the behaviour and memories of the second agent can be affected by receiving the narrative (Nehaniv 1999).

2.2.1 Narrative Structure

The structure of narrative has been widely investigated in different research fields such as mythology, linguistics and computer science. Some decades ago, structuralists and formalists defined the basic functions and story units for narrative structure, e.g. a story unit called *lexeme* which was invented by Barthes (1966). The term is a synonym for 'reading unit' – a single lexeme could be a page of a book, a web page, a short video or any other unit of narrative designed to be understood at once. Nowadays, *lexia* (the plural form of *lexeme*) are widely used as the basic story unit which is generated in modern story-telling systems (Mateas and Stern 2002, Cavazza et al. 2003) in different narrative structures. Story-telling systems basically create a story with vast amount of lexia and ask the user to make choices to navigate among these lexia while experiencing the story. Practical narrative structures are necessary to be generated by these systems since the number of lexia required grows exponentially with the growth of the number of choices in the story.

Linde (1993) formally developed another narrative structure for applying to life stories of humans. Linde's narrative structure provides a clear picture regarding different features of narrative and how these features can fit together in a simple personal experience. This structure can be interpreted as follows:

- Abstract – Type of story.
- Orientation (optional) – Abstractions about who, when, where and what happened.
- Narrative – Detailed descriptions of the event(s) that happened in the story.
- Evaluation – Evaluation of the whole story or a part of the story.

Based on Linde's study, a few years ago computer scientist Goguen (2001) proposed a systematic *Labov narrative structure* which adopts an Extended Backus-Naur Form (EBNF) notation to formalise a piece of narrative as a sign system. Extended BNF adds list and optional operators to standard BNF. Each extended BNF statement defines the syntax of a part of the source language. Goguen expects that stories represented via this narrative structure can be understood by computer programs. With Labov narrative structure, the following is an example of a summarised narrative in a general syntactic expression, in which *Abs* is abstract, *Ornt* is orientation, *Eval* is evaluation, *Cls* is narrative clause and *Coda* is for summarising the narrative and marking its end.

$$[(< Abs > + < Ornt > + < Abs >< Ornt >)[< Eval >]](< Cls > [< Eval >]) * [< Coda >]$$

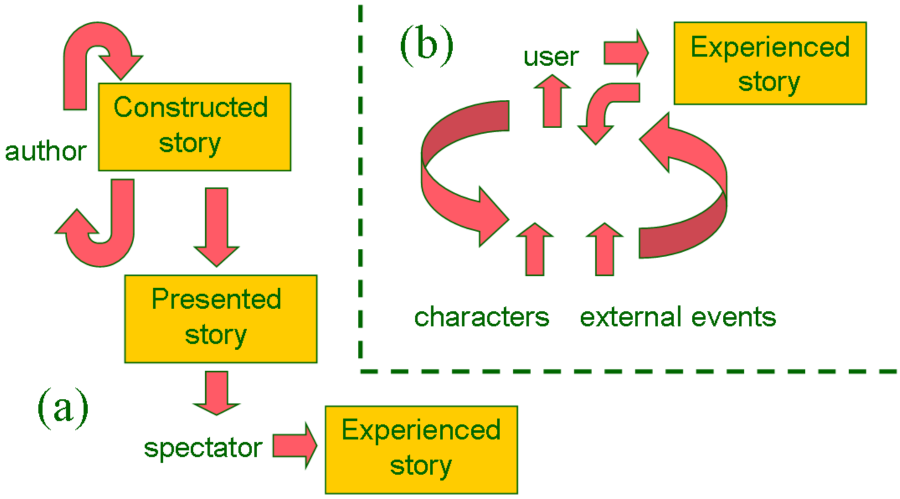

Figure 2.5: Two different narrative interactions between user and system: (a) typical narrative form such as literature and cinema; (b) real-time and interactive narrative such as interactive drama in theater and modern narrative story-telling systems. Taken from Aylett (2005, page 18).

In this example of story structure, [...] indicates either zero or one instance of whatever is enclosed, where + indicates exclusive or, and where juxtaposition of subexpressions indicates concatenation.

Researchers investigating *emergent narrative*, such as (Aylett 1999), stated that narrative structures for real-time and interactive narrative are very different and incompatible with certain narrative forms such as literature or cinema. The author is writing, telling and displaying the story, and meantime the reader is reading or viewing it, which is impossible for literature. Real-time and interactive narrative also make certain constraints to the richness of the story contents. Extracted from Aylett(2005), Figure 2.5 illustrates the central ideas of typical narrative form and real-time interactive narrative in regard to the interactions between user and system.

2.2.2 Origins of Narrative Intelligence

From the theories and work in narrative psychology which was founded by psychologist Bruner (1991), a detailed interpretation and discussion for applying the characteristics of narrative to the design of intelligent agents is specified by Sengers (2000). From Sengers' elaborations, two features of narrative, *intentionality* and *breaches* are particularly interesting in our research work for developing narrative autobiographic agents.

Sengers pointed out that intentionality is a critical issue since in a narrative, what actually happens is less important than what the characters feel or think about what has happened. It means that when people watch autonomous agents, they are interested in not only what the agent does, but also how the agent's choice was derived. Therefore, she stated that the agent architecture should be organized in a way that reasons for behavioural change are explicit and continuously expressed.

Breaches in narrative indicate that a story should contain something unexpected, some problem to be resolved, some unusual situation, etc. Sengers interpreted breaches as an enhancement of intentionality making the agent do something unexpected. Regarding autobiographic memory, it is about significant events – not routine matters which happen everyday and can be encrypted by scripts (Schank 1982). These events occur unexpectedly to the agent, and have outstanding impact such as recurring the same type of events in agents' memory. Scripts are not 'worth telling' unless they include the 'unusual', breaches, violations to the script which make a story interesting (Dautenhahn 2003).

2.2.3 Story Telling and Autobiographic Memories

As first discussed by Dautenhahn (1997), story-telling provides 'empathic resonance' through transferring own experiences or receiving experiences between autobiographic agents. A degree of empathic re-experiencing of (the internal state of) others since agents who received experiences may also reconstruct aspects of an individual's history of others. From a bottom-up perspective, when receiving stories from other agents, the receiver (an autobiographic agent) can selectively choose the information to store into its own autobiographic memory from different senders. As a certain level of trust of a specific sender (another autobiographic agent) can be built up after the receiver received many experiences from different senders and each experience has brought different levels of benefits, such as avoiding dangers to the receiver.

Consequently, story-telling brings changes to the agents' own memory and also the attitudes toward others. In addition to this, the usefulness of stories which the agent received from others will be known only after the agent re-experienced them later; it is the same idea as the *Evaluation* element in the narrative structure (see previous sub-section 2.2.1), but applying it to other agents' stories.

Understanding a story from another agent is neither simply matching the explicit contents of the story to the agents' own stories in its memory, nor using high-level cognitive processes from the perspective of human understanding. The agent has to employ a low-level abstraction process for recognizing the meaning and the contents of this story from its own accumulated stories, and then the agent will not directly merge this story with other old stories or store it into its own memory, but shape it to the agent's own story figure, i.e. using its own memory schemata to fit the story into its memory.

2.3 Memory and Communication in Computer Games

The design of the behaviours of non-player characters (NPCs) in computer games is dominated by two major techniques: Finite State Machine (FSM) and Scripting. FSM treats each behaviour of an NPC as a state and a limited number of states are pre-defined by the game designer; the actual state of the NPC is made to switch with seeming intelligence, depending on criteria also pre-defined by the designer (Cass 2002). Scripting is another type of technique defined by using a high-level language. It allows game designers to set the overall goals and playing methods of the game without hard-coding the control of game events (Cass 2002). Scripting is also good to craft an illusion of AI. A typical scripted scene is that a fixed sequence of actions or dialogs will be triggered when the condition has been matched in the game, such as an NPC answering a question selected by the player.

In recent years, researchers have applied concepts from embodied AI to the design and development process of computer games, such as in the following aspects (Champandard 2003):

- Embodied agents that are actually situated in realistic worlds: Embodied systems are mostly affected by their immediate surroundings. Only local information is collected by using sensors, similar to how humans or animals interact with their environment. Therefore the efficiency of the game can be improved when each agent needs to remember less information about the world.

- behaviour-based control architectures are suitable for intelligent control: Their reliability and simplicity, often providing a foundation for more elaborate techniques. Subsumption control architecture (Brooks 1986) is an good example since it has distributed components that react instantly to the environment and it is easier for development, as NPCs are usually built incrementally and validated by experimentation.

- Learning techniques can be used for the simulation of adaptive behaviours: To create intelligent capabilities for NPCs, these techniques have varied essential characteristics: 1) *teaching* requires a set of examples provided by human players to help agents in managing and understanding what to do; 2) *imitation* provides a demonstrator who is normally a human player for the agent to copy and learn the appropriate behaviours; 3) *shaping* organizes consecutive tests for the agent to learn, after the agent is able to perform simple tasks, more complex ones are given; 4) *trial and error* allows the agent to try out all possible approaches on its own by placing it in a new environment, and in this case the agent is expected to learn from the successful attempts.

In developing character-based AI for computer games, Isla and Blumberg (2002) indicated that one relationship that has yet to be explored thoroughly is the one between learning and explicit memory formation. Nowadays only a few behaviour simulation systems have made explicit use of episodic memory as a learning mechanism, where learning means individual adaptation processes that occur throughout a character's brain. One advantage of using episodic memory for learning, compared to other mechanisms such as reinforcement, neural networks and genetic algorithms, is speed. The game character can form usable hypotheses for making decisions or selecting behaviours to execute in the future, after just one observation of users or other agents.

There is also an outstanding computer game released in year 2001, *Black and White*, in which characters seem to apply episodic information and social communication in situations including a) when a character watches the player's actions and attempts to divine the intent behind them, this technique is called 'empathic learning'; and b) agents' behaviour will be controlled by their membership in overlapping social groups (Cass 2002). Since each group will have its own set of needs and attitudes toward other groups, its members reduce demands on memory and processing power by sharing information among themselves.

2.4 Alternative Approaches to Memory and Learning in AI

The utilisations of memory in learning mechanisms in AI, such as *reinforcement learning* and *neural networks*, are very different compared with the approach of episodic memory learning in our research.

Reinforcement learning (Sutton and Barto 1998) is a method by which an agent learns a policy based on a reward (a real valued signal) it receives following the actions it takes. These rewards can be used to teach it any task which can be represented as a Markov Decision Process. In the best known algorithm, Q-learning (Watkins 1989) it builds a look-up table in memory as it explores its environment. This table averages the immediate rewards plus the discounted utility of the successor state for each action in each state. The optimal policy is obtained by taking the greedy action (the action with the highest value) in each state.

An artificial neural network is a computational model that is inspired by the brain's information processing ability. It is constructed from a number of units (artificial neurons) which are interconnected by links. Each link has a numeric weight associated with it. Weights are considered as long-term storage in neural networks and learning usually occurs by updating the weights (Russell and Norvig 1994). Artificial single- or multi-layer neural networks are advantageous, especially in pattern recognition and classification tasks. *Recurrent neural networks* (RNNs) take some of the outputs and feed them back to the inputs or to the hidden layer in order to create a dynamic memory, so the networks are responsive to temporal sequences (Elman 1990). In recent years autonomous agents' dynamic properties have been studied extensively. Dynamic neural network models such as *continuous-time recurrent neural networks* (CTRNNs) which are used in conjunction with genetic algorithms (GAs) to produce robot controllers, have attracted a substantial amount of interest. An overview of CTRNNs can be found in Beer (1995). Early experiments by using CTRNNs on tasks such as visually-guided orientation, object discrimination and accurate pointing are studied in Beer (1996). Experimental results showed that the internal dynamics allows an agent to make use of the recent history of interaction with the environment for its behaviour execution on simple cognitive tasks.

Different from the above mentioned approaches, in this thesis, we focus on *investigating how significant events in autobiographic memory affect agents' adaptation to the environment as well as the narrative story-telling between agents through a highly abstracted and symbolically-grounded approach.* The reasons are as follows:

1. We are *not* studying the emergence of a capacity to represent episodic and autobiographic memory as a primary focus of research, for which neural networks might be more suitable.

2. The representation of episodic memories of the agents should be amenable to examination by the experimenters and their relationship to meaningful events for the agent should be accessible.

3. Communication of specific and/or abstracted episodic histories should be possible between the agents.

Since neural weights and their transmission are especially unsuitable for requirements (2) and (3) above, we take another approach in developing narrative autobiographic agents. The visualization of memory contents enables monitoring the learning process of memory agents, such as how an agent manipulates past experiences and why a particular event in the memory gets retrieved. This feature becomes more advanced when narrative communications take place between agents since story-telling and story-understanding can be observable.

In the research field of behaviour-based robotics, Arkin pointed out that, in the context of computational models used to express brain behaviour, schema theory and neural networks are two mainstream forms fully compatible in building behavioural models: "Schema theory is a higher-level abstraction by which behaviour can be expressed modularly. Neural networks provide a basis for modeling at a finer granularity, where parallel processing occurs at a lower level" (Arkin 1998, page 41).

The approach to modeling autobiographic memory taken in this thesis, strongly inspired by schema theory, meets all three requirements listed above and is suitable to address the research questions relevant to this work.

2.5 Summary

> *"Instead, today, many AI researchers aim toward programs that will match patterns in memory to decide what to do next. I like to think of this as 'do something sensible' programming. A few researchers – too few, I think – experiment with programs that can learn and reason by analogy.*

These programs will someday recognize which old experiences in memory are most analogous to new situations, so that they can 'remember' which methods worked best on similar problems in the past." (Minsky 1982, page 4)

As the research fields of human memory and narrative intelligence are well established and both of them are interdisciplinary, the vast amount of literature from these two research fields have brought us fruitful ideas to develop narrative autobiographic agents in a bottom up fashion. Particularly the theory and the experiments of remembering from Bartlett's intellectual achievements (Bartlett 1932) and the model describing hierarchical knowledge structures in the autobiographic knowledge base from Conway (1996), have inspired us very much about how events could be organized in a computational autobiographic memory model. Aside from the essential characteristics introduced in the previous sections, nested structure (Neisser 1986) and self-schemata (Barclay 1986) of autobiographic memory also produce other interesting features, which are possible to apply into bottom-up design of narrative autobiographic agents.

Story telling and understanding in narrative, as high-level cognitive processes which seem to involve human consciousness, are highly challenging tasks in building intelligent agents and have yet to be established in AI and other research fields. For that reason, this research investigates, from the bottom-up approach, the nature of narrative intelligence for Artificial Life agents by making use of stories that are natural and meaningful to them.

General characteristics of the human memory system especially autobiographic memory illustrated in this chapter will be taken into consideration in the architecture design for narrative autobiographic agents. Different types of memories can be seen as non-sensory data from the perspective of narrative and historical temporal grounding, which helps an agent escape from the present in its perception-action cycle (Nehaniv et al. 1999).

Chapter 3

Basic Autobiographic Memory Architectures

The first set of experiments conducted as part of this thesis focused on studying autobiographic agents, from the bottom-up approach and Artificial Life perspective, by implementing finite-state autobiographic memory into the basic control architecture for a *Purely Reactive* agent. Different static virtual environments with resource+ allocations are created for a single agent or multiple agents to 'survive' in the environment. In order to find out the location of the resources by using the local information of the sensory data, autobiographic agents are able to remember their previous action sequences for going back to the particular resources. Therefore, both studies from (Ho, Dautenhahn and Nehaniv 2003) and (Ho, Dautenhahn, Nehaniv and te Boekhorst 2004) show that autobiographic agents and agents that share experiences outperform the reactive agents in surviving in the static environments and in both single agent and multi-agent experiments. This chapter illustrates the essential motivations, design and implementations from the earlier work. The first work (Ho et al. 2003) has been presented at an international workshop, Intelligent Virtual Agents 2003 (IVA 2003) at Schloss Irsee, Germany in September 2003; the second work (Ho et al. 2004) has also been presented in an international conference, the 8th Conference on Intelligent Autonomous Systems (IAS-8), at Amsterdam, Netherlands in March 2004.

3.1 Motivation

In our research regarding the study of single autobiographic agents, we have been pursuing a bottom-up, Artificial Life perspective, developing minimal memory-based control architectures, and investigating whether and how these control architectures benefit the survival of agents in their virtual environments.

3.1.1 Study of Single Autobiographic Agents

In the research area of intelligent virtual agents (Aylett and Luck 2000), virtually embodied agents should be able to achieve a certain level of autonomy, which means they need to maintain their own internal variables at an adequate level, besides responding to and reacting to different types of stimuli from their environments by their virtual sensors and virtual actuators. To achieve these goals, a robust control system is necessary to control the agents. The subsumption architecture (Brooks 1986) is one of the solutions dedicated to behaviour-based agents, since it decomposes the problem by defining layers of competing behaviours and defining relationships between those behaviours. This leads to a control architecture that can select an action to execute immediately based on the current value of the sensors and internal variables of the agent, even though the agent does not possess any complex internal knowledge representations, such as an overall map of the environment. Furthermore, we implemented principles of the agent programming language PDL (Steels 1992), which updates sensory quantities using the latest reading from sensors and which then sends new values of the action parameter quantities to the actuators. PDL was previously designed to specifically support dynamical, life-like behaviour for autonomous agents (Dautenhahn 1999a). The details of PDL can be found in Appendix A.

Sophisticated virtual agents are likely to require memory designs appropriate to sustaining interactive experiences with other agents (including humans) and their environments over time. The environments studied in the present work are 'ecological', i.e. virtual worlds that contain food resources and where agents need to address simple tasks in order to survive. Similar environments have been used widely in Artificial Life research (Cañamero, Avila-Garcia and Hafner 2002, Cuperlier, Laroque, Andry and Gaussier 2004). The goal of this biologically motivated approach is to develop and thoroughly test and compare generic control architectures for autobiographic agents. The present work describes the implementation and results from the first phase of this project where we used very simple virtual environments in order to examine differences between the performances of agents with different autobiographic or reactive control architectures. Based on this research, later work will address autobiographic memory in more complex tasks and more complex environments (see Chapters 5 and 6).

3.1.2 Study of Multiple Autobiographic Agents

This work extends the study of autobiographic memory in autonomous agents (Ho et al. 2003) by studying multiple autobiographic agents in a static virtual environment with different conditions on the communication of episodic memories. For the purpose of this work, a 'communication' is defined as a sequence of events (perceptions and actions) that some agent has acquired in its memory through experience and that is being transmitted to another recipient agent. In this way, an agent can share another agent's episodic memory.

All 'communications' in this chapter will be receptions of episodic information of this kind. In contrast, ethologists define communication as a signaling by one organism that when responded to by another confers some advantage (statistically speaking) to the signaler or its group (Burghardt 1977). Whether and how communications (in our sense) of memories actually do confer some advantage as required by the ethological definition will be a primary concern of our research.

Previously we have discussed how animal foraging behaviours can benefit from memory and communication in many cases (see Sub-section 2.1.4 in Chapter 2). As autobiographic agents, humans are constantly telling and re-telling stories about themselves and others to establish social relationships and connectedness: "telling (part of) a plausible autobiographical story to others is more than showing a plausible sequence of episodic events, it includes the construction of a plausible story based on one's goals, intentions and motivations" (Dautenhahn 1999b, page 4). From the perspective of minimal Artificial Life agents we are concerned with a) in what situations communications of memories should occur, b) what kinds of information should be exchanged, c) how costs attached to communications impact the results, and d) how the episodic information could benefit a single agent and the whole multi-agent system consisting of minimal autobiographic agents.

In our work we are pursuing a bottom-up, Artificial Life perspective, studying communications in the context of an *Event-based* mode of the *Trace-back* memory control architecture derived from our previous work with single autonomous agents (Ho et al. 2003). We now investigate the effectiveness of this control architecture for the overall survival of multiple agents in their virtual environments. Results of this study can be found in sub-section 4.2.2. The environment studied in this work (see Figures 3.15 and 3.16) is similar to the one described in the previous sub-section 3.1.1, except that there is only one type of resource (*Food*) in the environment, and thus each agent has only one internal state (*Glucose*) to maintain. Agents need to carry out similar tasks to survive (e.g. finding food, avoiding obstacles).

3.2 Design and Implementation

In this section, the detailed implementation of different virtual environments and agent memory control architectures in our earlier study are illustrated. These basic memory control architectures consolidate the relatively sophisticated Long-term Autobiographic Memory architectures we are going to show in later chapters.

3.2.1 Single Autobiographic Agents

The web-based 3D technology VRML provides a set of comprehensive definitions of geometric shapes for building a suitable wide range of virtual environments and 3D objects involved in the related experimental simulations. To achieve a certain level of autonomy for virtual agents, control architectures for agent behaviours are implemented by using JavaScript and applying timed constraint programming techniques (Saraswat, Jagadeesan and Gupta 1994). In each time step, the script node which acts as the brain of the agent routes necessary information to modify the shapes of

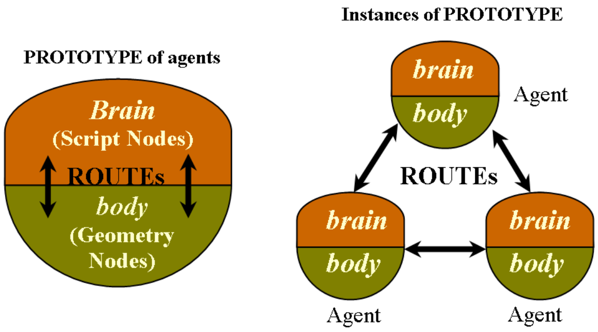

Figure 3.1: Simple design of a virtual autonomous agents using VRML (left), the design for multi-agent systems is shown on the right.

the agent's body and the objects in the environment, so that autonomous behaviour and continuous movement can be carried out. Figure 3.1 illustrates a simple design using VRML to implement virtual autonomous agents, which can dynamically change the body (geometry) in terms of position, shape, or movement by calculating the agent's own status.

Figure 3.1 also shows that duplicating the original prototype of agents can lead to multi-agent systems, in which agents of comparatively similar architecture facilitate the communication phase with routing information between them for cooperative goals. Studies in multi-agent experiments will take advantage of this multi-agent option (see Sub-section 3.2.2 in this chapter and Sub-section 5.2.2 in Chapter 5). In this section only one virtual agent in the environment is being studied at a time.

Reactive and Autobiographic Memory Architectures

We study three types of agents, which are based on a subsumption control architecture, namely *Purely Reactive*, *Trace-back* and *Locality*. All agents have a finite lifetime. The survival of an agent depends on maintaining homeostasis for three

Environmental Resource	Internal Variable	behaviour Execution
Nest	Energy	Homing
Water	Moisture	Drinking
Food	Glucose	Eating

Table 3.1: Relationships between resources in the environment, agents' internal variables and agents' behaviour executions.

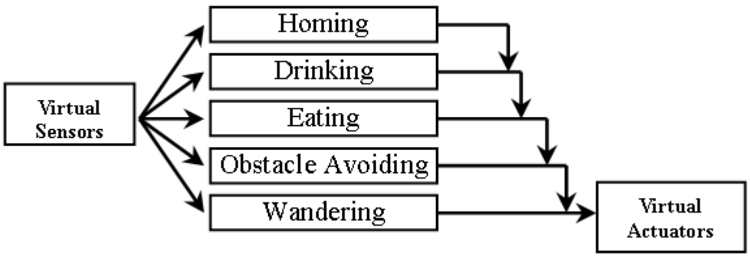

Figure 3.2: behaviour hierarchy which is based on the subsumption architecture for a Purely Reactive agent.

internal variables: Energy, Moisture and Glucose. Each internal variable is initialized with a maximum value at the start of each experimental simulation run. Each translation or rotation of the agent will reduce each internal variable by a certain value. When one of the internal variables drops below a threshold, which is half of the maximum value, then the agent will start searching around for a specific resource located in the environment. Relationships between resources in the environment, agents' internal variables and agents' behaviour executions are in Table 3.1.

If the agent is not able to detect the resource which it needs, and if the value of a particular internal variable is less than a particular minimum value, then the agent will die. The experimental parameters (thresholds etc.) that allow the agents to live in the virtual environment, but eventually die, were determined in initial tests.

Purely Reactive The architecture of the *Purely Reactive* agent includes five layers and higher-level behaviours which inhibit or override lower-level behaviours. The agent usually wanders around in the environment by executing the bottom layer in the architecture. At the same time, PDL control which allows the description of parallel processes that react to sensor readings by influencing the actuators is employed to increase or decrease the velocity of the agent in order to achieve a certain level of continuous, natural movement. For example, the agent would slowly increase its velocity after it finished an obstacle avoidance behaviour and when it started again to explore the environment. When the agent encounters an object, which can be any kind of resource, obstacle or one of the boundaries of the environment (walls), then the agent avoids the obstacle or the wall by generating a random direction rotating its body. This behaviour will also be triggered in case the agent encounters a resource object, in case the internal variable which needs that particular resource is higher than the corresponding threshold. Figure 3.2 shows the control architecture for the *Purely Reactive* agent. The system design of the *Purely Reactive* architecture can be found in Sub-section B.1.2 in Appendix B.

Trace-back On the basis of the design of a *Purely Reactive* architecture, autobiographic agents with memory *Trace-back* or *Locality* possess a memory module on top of the subsumption architecture. Each type of autobiographic agent has a unique mechanism for making memory entries as the remembering process, and using the memory as a tracing process. In the case of the *Trace-back* mechanism, the agent has a finite number of circularly-ordered memory entries. Introduction of new entries depends on the way of making memory entries for the agent (*Event-based* or *Time-based*). In the *Event-based* mode, making new memory entries occurs when

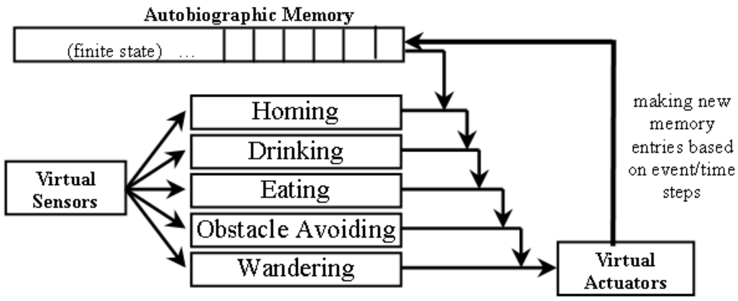

Figure 3.3: Memory architecture of Trace-back mechanism.

the agent encounters a different object and changes its current behaviour; in the *Time-Based* mode, a new memory entry is made every fixed number of time steps. Information of each memory entry includes the current time step of the simulation, the behaviour which is being executed, the direction the agent is facing, the object encountered by the agent (if any), how far the agent has walked (distance) since last encountering an object, as well as the current internal variable. This information is inserted at the current position of the index into the finite circular memory. Figure 3.3 illustrates the memory architecture of the *Trace-back* mechanism, Figure 3.4 shows examples of memory entries made by executing the *Event-based* mode and the *Time-based* mode. The system design of the *Trace-back* autobiographic memory architecture can be found in Sub-section B.1.3 in Appendix B.

The memory trace back process will be triggered if one of the internal variables of the agent is lower than the threshold, and if the corresponding resource can be found at any entry, which indicates that the agent has previously encountered the resource. Once trace back has started, the agent will simply 'undo' all previous behaviours. As indicated in Figure 3.4, the agent will execute in reverse order, the action to undo each step starting from the current entry to the target resource.

Target resource in memory

Current entry in tracing process.

	Time	Directions	Behaviors	Objects	Distance	States
0	0	-.7511			0	Happy
1	640	2.107	Avoiding	Wall	39.8349	Happy
2	673	-1.3763	Wandering	Nothing	.134	Happy
3	675	1.4882	Avoiding	Wall	.2004	Happy
4	679	-1.3276	Wandering	Nothing	0.0696	Happy
5	686	-2.2367	Avoiding	Wall	.7014	Happy
6	877	2.4528	Wandering	Nothing	.2136	Happy
>> 7	1604	-1.1472	Avoiding	Food	72.8456	Happy
8	1747	-.4031	Wandering	Nothing	.2804	Happy
9	2241	2.0818	Avoiding	Obstacle	49.4988	Happy
10	2343	-2.4002	Wandering	Nothing	.2033	Happy
11	2587	2.5197	Avoiding	Obstacle	24.4487	Happy
12	2796	1.7867	Wandering	Nothing	.2302	Happy
13	3278	-2.0802	Avoiding	Wall	48.2964	Happy
14	3411	-.8651	Wandering	Nothing	.3417	Happy
15	3511	-2.1553	Avoiding	Obstacle	10.0199	Happy
16	3576	-2.1237	Wandering	Nothing	.2696	Happy
17	4021	-2.1186	Drinking	Water	44.589	Thirsty
18	4042	-2.1185	Drinking	Water	1.4027	Happy
19	4177	.2419	Avoiding	Water	.1001	Happy
20	4322	.7675	Wandering	Nothing	.7538	Happy
> 21	4837	.7675	Wandering	Nothing	51.6029	Hungry

	Time	Directions	Behaviors	Objects	Distance	States
0	0	-1.7508			0	Happy
1	100	-1.7508	Wandering	Nothing	.1209	Happy
2	200	-1.7508	Wandering	Nothing	2.2099	Happy
3	300	-1.7508	Wandering	Nothing	4.21	Happy
4	400	-1.7508	Wandering	Nothing	6.2099	Happy
5	500	-1.7508	Wandering	Nothing	8.2099	Happy
6	600	2.0029	Avoiding	Obstacle	.2722	Happy
7	700	2.0029	Wandering	Nothing	.8388	Happy
8	800	2.0029	Wandering	Nothing	9.9269	Happy
9	900	2.0029	Wandering	Nothing	10.0199	Happy
10	1000	2.0029	Wandering	Nothing	10.0199	Happy
11	1100	2.0029	Wandering	Nothing	10.0199	Happy
12	1200	2.0029	Wandering	Nothing	10.0199	Happy
13	1300	.2234	Avoiding	Wall	4.0704	Happy
14	1400	-.7239	Avoiding	Wall	.8005	Happy
15	1500	-.7239	Wandering	Nothing	6.3126	Happy
16	1600	-.7239	Wandering	Nothing	10.02	Happy
17	1700	-.7239	Wandering	Nothing	10.02	Happy
18	1800	-.7239	Wandering	Nothing	10.02	Happy
19	1900	-.7239	Wandering	Nothing	10.02	Happy
20	2000	-.7239	Wandering	Nothing	10.02	Happy
21	2100	-.7239	Wandering	Nothing	10.02	Happy
22	2200	2.8499	Avoiding	Food	4.6916	Happy
23	2300	2.8499	Avoiding	Food	0	Happy
24	2400	2.8499	Wandering	Nothing	6.7134	Happy
25	2500	2.8499	Wandering	Nothing	10.02	Happy
26	2600	2.8499	Wandering	Nothing	10.02	Happy
27	2700	2.8499	Wandering	Nothing	10.0199	Happy

Figure 3.4: Trace-back memory entries made by executing *Event-based* mode (left) and *Time-based* mode (right). The left figure shows a situation where the agent is Hungry (the last column State of current entry) and is starting the Trace-back process since the resource Food was found in its memory (the fifth column object of entry No.7). It will successively undo its actions leading back toward the situation in entry 7. The right figure is similar, but a time-based memory entry has been made every 100 time steps (see the values under the second column Time).

This mechanism has a close connection to the group-theoretic notion of inverse in mathematics (Nehaniv and Dautenhahn 1998a). Thus, the agent will execute the reverse of each action step-by-step starting with the most recent action, using the information specified in Direction and Distance. The trace back process will be completed once the agent has executed all memories entries and has reached the target resource. At this moment, the agent will start sensing around for the resource. Note, there is a possibility that the resource is not available at this location since the actual rotation value in each entry might have been slightly distorted by noise. Also, the trace back process will be terminated if the agent collides with any of the

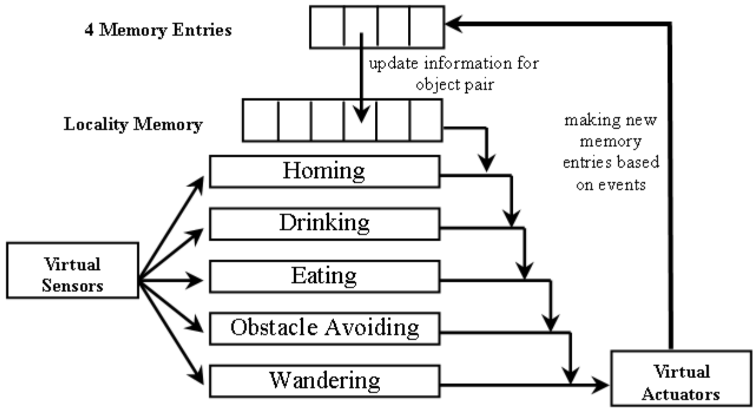

Figure 3.5: Memory architecture of Locality mechanism.

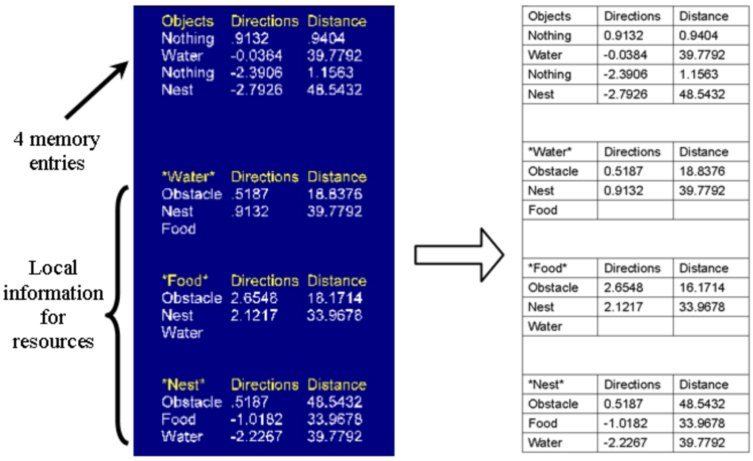

Figure 3.6: Memory entries made by Locality mechanism. The left figure is captured from a running experiment. For the purpose of easy viewing, the table at the right represents the same data. Values of Direction are shown in VRML rotation unit. In VRML, a π value (3.14159...) is used to represent 180°. For example, if the value of VRML rotation unit is 1.0, this value is equal to approximately 57.30°.

objects in the environment (e.g. due to accumulated errors caused by noise).

Locality In contrast to the *Trace-back* mechanism, the *Locality* memory agent has only four memory entries for remembering the most recent behaviours using the *Event-based* mechanism. In addition, the agent maintains information relevant for traveling from one type of object to any other given type of object (*Locality* memory). Making or replacing entries of long-term *Locality* memory occurs exactly when a particular pair of objects has been encountered. The reason for having four memory entries at the top of architecutre (see Figure 3.5) is as follows:

1. Each complete action occupies two entries. In the first entry the agent rotates its body to avoid an object; in the second entry it travels in a straight line to depart from that object.

2. Information showing the relationship between a pair of useful objects contains two contiguous actions.

Three 3 times 2 tables represent all relevant information for the possible specific pair of objects; each row of the table contains information of Direction and Distance, specifying what the agent did while traveling between two objects (Figure 3.6). The tracing memory process occurs when the agent is looking for a specific resource and the entry of the object leading the agent to that resource contains the required information.

The *Locality* memory architecture of the agent is shown in Figure 3.5, and a sample of memory entries is shown in Figure 3.6. The system design of the *Locality* autobiographic memory architecture can be found in Sub-section B.1.4 in Appendix B.

Virtual Environment and Virtual Agent

As the experimental test-bed, in order to assess the impact of autobiographic memory without unnecessary complexity, we designed simple, large size and static environments with different object distributions with basic geometric shapes which are generated in real-time from VRML. In addition to being able to study the consequences of autobiographic memory influencing an agent's behaviours, we are also concerned with system performance since VRML is essentially a description-based language for 3D scenes and objects distributed on the web. Rather than using existing simulation tools such as: Swarm (Center for the Study of Complex Systems at the University of Michigan 2005), Webot (Cyberbotics 2005), or other physics engines, for modeling the environment and agents, we create our own virtual environment by using VRML. This approach has advantages as follows:

- Easier to design the environment and behaviours of agents: VRML provides a easy and direct way to describe 3D objects. Thus objects in the environment can be built easily, transformed and made as prototypes for further uses. Agents' behaviours can be programmed and updated dynamically through internal VRMLScript or many other languages like Java, C and C++. Moreover, standard VRML is platform-independent. Therefore theoretically simulations built by using VRML can be run on different types of machines and operating systems.

- Achieving better performance through reducing executions of redundant libraries: As experiments carried out in this research are typical Artificial Life simulations, these experiments don't require representations of factors from physics, such as friction and gravity used in physics engines.

- Flexible to create program interfaces: Highly flexible program interfaces can be created as extra components by Java or other programming languages for various purposes, e.g. observing the change of internal variables of agents, or for illustrating the memory contents dynamically.

The virtual environment consists of various types of resources including food, water and nests, as well as obstacles and walls which serve as the boundaries of the environments. All resources are *unlimited*. Experimental virtual environments can be categorized according to the distribution of objects. For the first type of autobiographic agents using the *Trace-back* control architecture, objects are uniformly and symmetrically distributed in the environment. For the second type of autobiographic agents using the *Locality* control architecture, similar types of objects are grouped together in certain areas which can be easily found by the agent in the initial stage of making memory entries. These two types of environments are illustrated in Figures 3.7. The system design of this environment can be found in Sub-section B.1.1 in Appendix B.

Figure 3.8 and 3.9 illustrate the related metric information for both environments used in this study.

To increase the level of realism of both the simulated virtual environments and agents' autobiographic memories, we introduced environmental noise – a random noise signal ranging between $-5°$ and $5°$ from a Gaussian distribution with a mean of zero to slightly alter the Direction value when an agent is retrieving an entry from its memory. The detailed experimental settings can be found in section 4.1 of Chapter 4.

The virtual agent has a highly abstracted body, which includes the head, and the tail for indicating the agent's current direction. A partly transparent area of

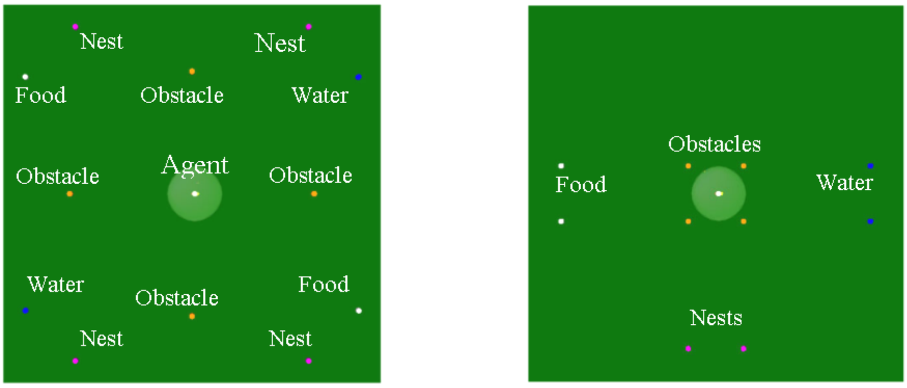

Figure 3.7: The left figure is the view of looking down from above, showing the object distribution in the first type of virtual environment where *Trace-back* control architecture is used. The right figure is the object distributions in the second type of virtual environment using *Locality* control architecture.

circular shape shows the effective range provided by the virtual sensor of the agent. The appearance of the virtual agent is shown in Figure 3.10.

3.2.2 Multiple Autobiographic Agents

With the same technical implementation described in sub-section 3.2.1 for both virtual environment and agent architectures, variable numbers of autobiographic agents were evaluated for the ability to survive under different experimental conditions related to *when* communication occurred (never, only when a potential recipient is hungry, or whenever one agent encounters another one) and whether or not cost is incurred. Note that the environment has only one type of resource (*Food*) and thus agents in this particular study needs to maintain only one internal state (*Glucose*). The *Food* resource is not limited so the agents are not in competition. The agents' task was to maintain their glucose level within a homeostatic range while exploring the environment and thus not remaining near a resource permanently. Maintaining

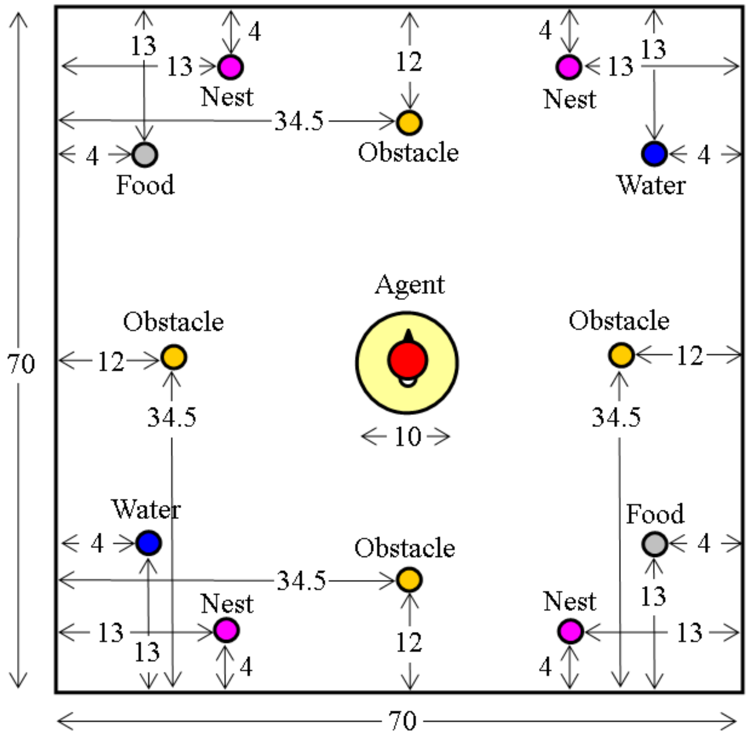

Figure 3.8: Information represented in VRML distance unit for environment size, agent's sensory range and object locations. This is the first type of virtual environment where *Trace-back* control architecture is used. The agent needs to find food, water and nest to replenish the decreasing internal variables.

the glucose level, together with sensing and reacting to objects and obstacles, allows an agent to 'survive' in its environment. In this context, survival of an agent depends on its ability in monitoring and maintaining its glucose level. This ability can also be enhanced to allow the agent to remember past action sequences which lead to reward from the environment. The idea of an agent maintaining essential variables in homeostatic ranges in order to 'survive' in its environment has been widely adopted in AI and robotic research (Meyer 1995, Cañamero et al. 2002, Cuperlier et al. 2004) and particularly Artificial Life simulations (Lowe, Cañamero, Nehaniv

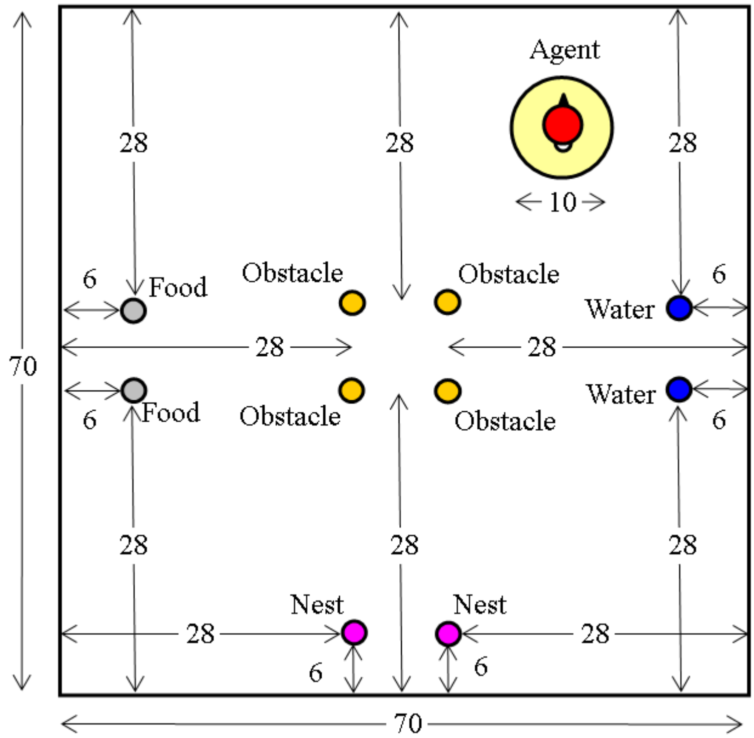

Figure 3.9: Information represented in VRML distance unit for environment size, agents' sensory range and object locations. This is the second type of virtual environment where *Locality* control architecture is used.

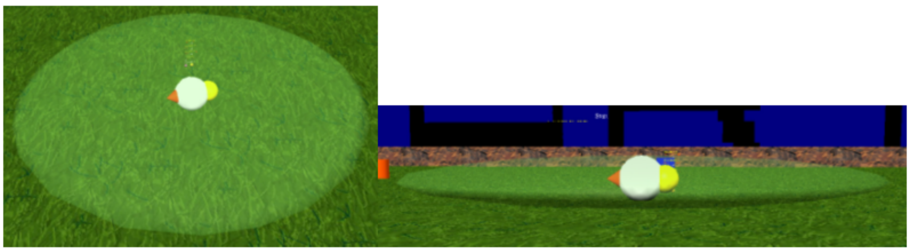

Figure 3.10: An abstract model of a virtual agent with a round-shaped sensing area. The yellow sphere is the head, and the red cone is the tail of the agent.

and Polani 2004).

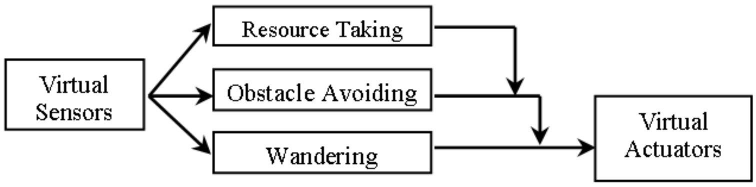

Figure 3.11: behaviour hierarchy which is based on the subsumption architecture for a *Purely Reactive* agent.

Improved Autobiographic Memory Architecture

We mainly study and improve two architectures of agents from the previous Sub-section 3.2.1, which are based on a subsumption control architecture (Brooks 1986), namely *Purely Reactive* and *Trace-back*. The design of both *Purely Reactive* and *Trace-back* autobiographic memory architectures in this work is similar to the designs and implementations in Sub-section 3.2.1, except the agents need to maintain only one internal variable. The architecture of a *Purely Reactive* agent is illustrated in Figure 3.11. The system design of the *Purely Reactive* architecture can be found in Sub-section B.2.2 in Appendix B.

The autobiographic architecture with a memory *Trace-back* process in this work is designed to be more efficient and easier to be shared between agents. An *Event-based* memory entry making mode is used in the design, in which the introduction of new entries occurs each time the agent experiences an event, i.e. encounters either an object or agent, and/or changes its current behaviour. There are essentially two improved features, firstly each time when the agent encounters the resource, all memory entries are cleared and the next memory entry will be made at the first location in the memory table. The index is then set to zero. Information of each memory entry includes the current direction which the agent is facing, the kind of object encountered by the agent (if any), and how far the agent has walked

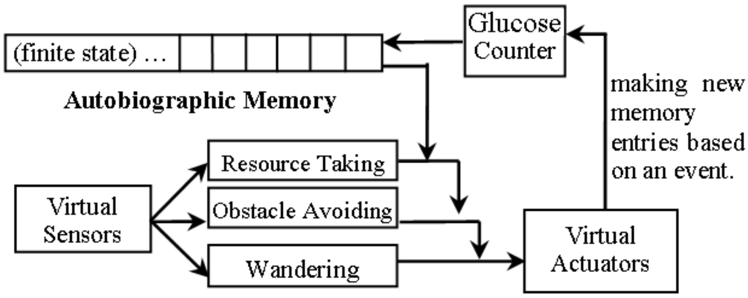

Figure 3.12: Memory architecture of the *Event-based* Trace-back mechanism.

(Distance) since the last event. This information is inserted at the current position of the index into the finite memory table. A glucose counter which is the second improved feature, is used for calculating the total cost that would be incurred if the agent were to trace-back through the memory entries, undoing remembered actions, while trying to obtain the resource. The value of this counter will be increased when a new memory entry is made. Then, a comparison process will be executed immediately for checking if this value is larger than the current internal variable of the agent. If this value is larger, it means the agent would not be able to successfully finish the tracing process and go back to the position of the resource which the agent previously encountered, so all entries in the memory table will be cleared in this situation. Figure 3.12 illustrates the memory architecture of the *Trace-back* mechanism. Figure 3.13 shows samples of memory entries made by executing the *Event-based* mode.

With the memory trace back process, agents will 'undo' all previous behaviours, as described in the previous sub-section 3.2.1. In multi-agent experiments, accumulated noise makes the agent sometimes unable to finish the *Trace-back* process since it collides with another object rather than the target one. In addition, other agents running in the environment also create the effects of 'disturbances'. These effects

Index	Direction	Distance
1	1.3525	53.2689
2	-0.5896	0
3	3.0587	15.7635
...	...	...

Figure 3.13: Autobiographic memory sample entries. During the Trace-back process, inverse action sequences are executed for each memory entry in reverse order, e.g. for an agent with autobiographic memory entries as above, to return to the previously encountered resource, Index 1 should be the last action sequence to be undone.

are particularly likely in the multi-agent context which could therefore obstruct the Trace-back process of the autobiographic agents. In a multi-agent context, the same control architectures are running on all agents.

Agent Communications of Episodic Memories Effective communication of episodic memories is the fundamental research issue addressed in this work. Communications can only come from agents which have already acquired an autobiographic memory, i.e. where at least one memory entry has been made.

Communications can occur in two ways: 1) an agent will always receive memory information from a perceived, nearby agent, or 2) an agent perceiving another will receive memories only if it is ‘hungry’, i.e. if the internal variable of the agent is lower than a certain threshold. Nevertheless, based on a general assumption that any memory is more beneficial than no memory an agent replaces its own memory entries by the information from other agents when either its memory table is empty, or if the action sequences that can be copied from the other agent’s memory would take less glucose to get back to the resource, thus giving a better path to the resource that can later be used in the *Trace-back* process. These two conditions suggest that only potentially useful episodic information (useful as evaluated from the perspective

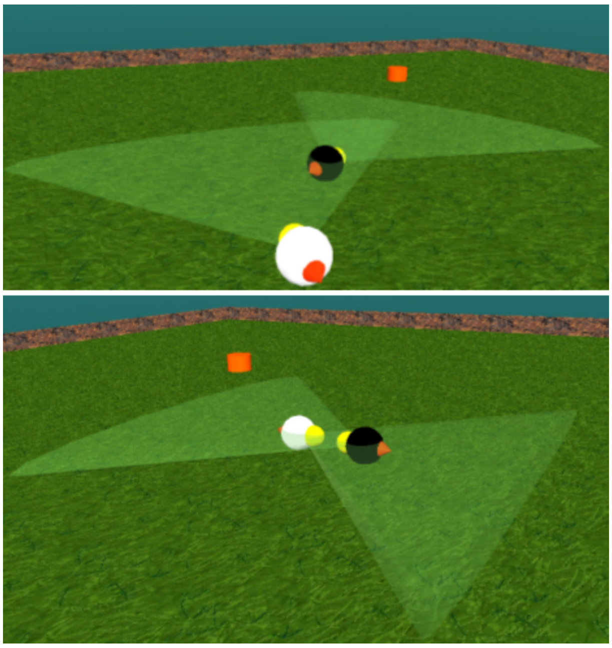

Figure 3.14: Two screen captures showing two autobiographic agents communicating with each other. Situations can be either 1) one is sensed by another (upper situation), or 2) two of them are facing each other (lower situation).

of the agent copying and later using this information) will be copied.

An agent can communicate with another only if both of them are close enough and, at least, one is in the sensory range of another one. Figure 3.14 shows two sample situations when communication between two agents occurs.

Communication costs represent a certain value that will be deducted from a recipient agent's internal variable for each memory entry being transmitted from one agent to another.

The system design of the communicative *Trace-back* autobiographic memory architecture can be found in Sub-section B.2.3 in Appendix B.

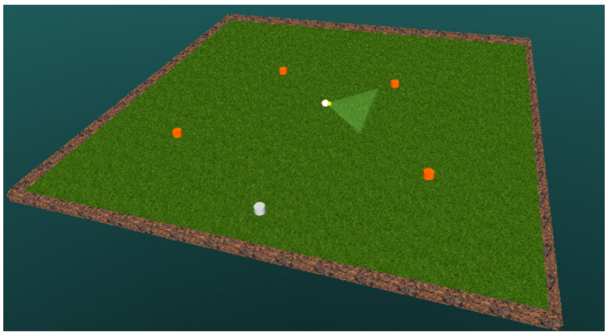

Figure 3.15: Resource and objects' distribution in the virtual environment. (White cylinder represents the resource and others represent obstacles.)

Virtual Environment and Virtual Agent

In order to assess the impact of communications between autobiographic agents, we designed a simple, large size and static environment with irregular object distribution with basic geometric shapes which are generated in real-time from VRML. The virtual environment consists of one resource, four obstacles and walls which serve as the boundaries of the environment. The resource in the environment is *unlimited*. All experimental simulations were running in the same virtual environment in order to make proper comparisons of the results generated from single agent or multi-agent experiments. The overview of the object distribution in the virtual environment is illustrated in Figure 3.15. Obstacles and walls are static, similar to environments shown in Figures 3.8 and 3.9, but the presence of other agents introduces dynamic aspects to the environment. Figure 3.16 shows the related metric information for this environment. The system design of this environment can be found in sub-section B.2.1 in Appendix B.

Basic cylinder and cone shapes are used to construct the body and the sensing area of the virtual agents, as shown in Figure 3.17. The simple design of the virtual

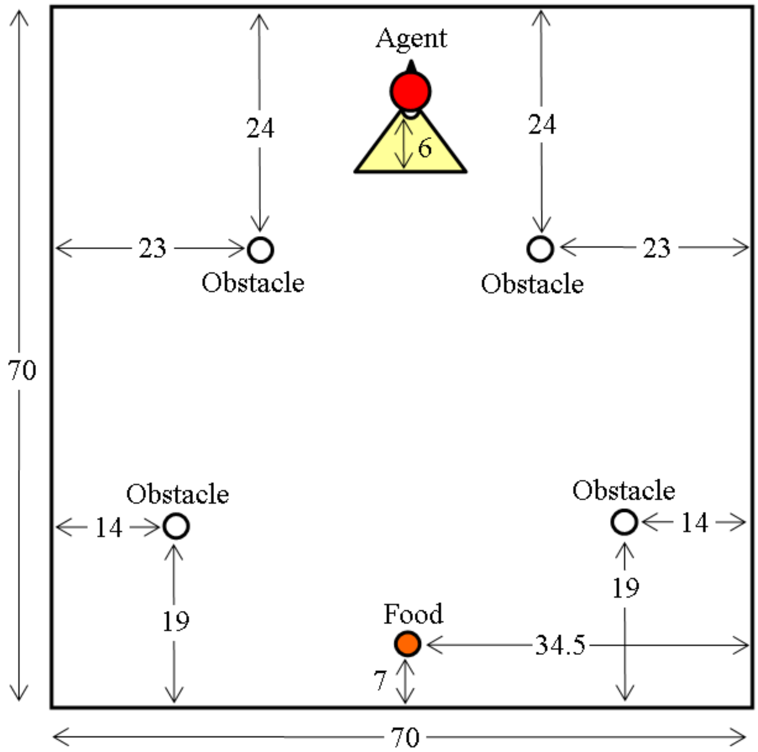

Figure 3.16: Information represented in VRML distance unit for environment size, agents' sensory range and object locations. This environment is used in experiments with multi-agent .

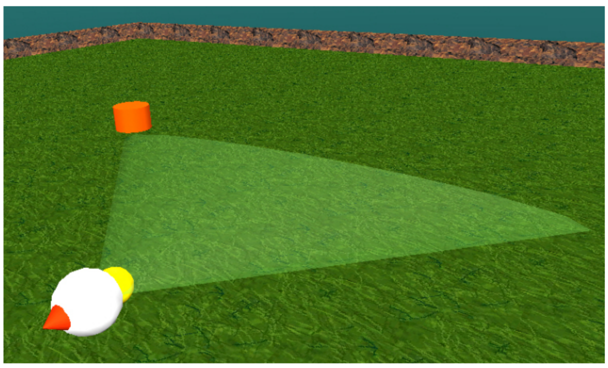

Figure 3.17: A virtual agent and its sensing area (triangle).

environment and the plain appearance of the virtual agent are necessary in consideration of the performance of current PC hardware, as the speed of the simulations can be noticeably decreased in a multi-agent context. The sensor of the virtual agent is modeled as effectively working within a certain range and angle.

Chapter 4

Experiments for Basic Autobiographic Agents

4.1 Single-agent Experiments

An experimental framework has been used for studying different possible mechanisms for developing autonomous autobiographic agents from an agent-centered viewpoint. We investigate the hypotheses that, for appropriate contexts, different control architectures of minimal autobiographic agents with a finite-sized memory can be advantageous to an autonomous, reactive agent (Dautenhahn and Coles 2001). As illustrated in Section 3.2.1, control architectures for autobiographic agents are categorized as *Trace-back* and *Locality*. The main characteristic of memory functioning in the *Trace-back* architecture is that a memory entry is continuously and directly made based on sensory information while the agent is wandering around in its environment. This is similar to a short-term memory since earlier entries will be replaced when all memory entries are filled. Moreover, *Event-based* and *Time-based* are two inherently different mechanisms used for making memory entries with the Trace-back control architecture. The *Event-based* mechanism is based on the agent's encounters with various objects in the environment, in contrast to the *Time-based* mechanism, where agents make memory entries at regular, fixed periods of time (periodically after a certain number of time steps). Comparatively, a 'long-term' memory approach is implemented in the *Locality* architecture. Here, only information which is meaningful (leading to the increase of at least one of the internal variables) to the agent will be remembered as 'long-term' memory entries.

The lack of unique object identifiers in the virtual environment produces a limiting factor. As a consequence, this will inhibit the agents' ability to recognize specific targets in unfamiliar environments. Similarly, the unavailability of global coordinate information of distinct objects restricts the agents' resource tracking system. An important factor which must be taken into account in our experiments is noise generated from the environment which influences the precision of the agents' actuators. Noise is likely to decrease the usefulness of the autobiographic memory in the process of remembering. We therefore hypothesise that noise will generally impair the agents' survival in the environment.

4.1.1 Experimental Settings

Experiments for examining the survival of autobiographic agents in static environments can be separated into two categories, namely *Trace-back* memory agents versus a Purely Reactive agent, and *Locality* memory agents versus a *Purely Reactive* agent. Table 4.1 shows the design for experimental runs for Trace-back memory agents. Each condition has 50 runs and takes 2 to 3 hours for execution time on a desktop computer (Pentium 4, 2 GHz, 512 MB main memory); there are 25 condi-

Conditon	**Mem-Size**	**Time-Step**	**Identifier**
PR	0	NIL	1
EB mem20	20	NIL	2, 14
EB mem50	50	NIL	3, 15
EB mem100	100	NIL	4, 16
TB 50 mem20	20	50	5, 17
TB 50 mem50	50	50	6, 18
TB 50 mem100	100	50	7, 19
TB 100 mem20	20	100	8, 20
TB 100 mem50	50	100	9, 21
TB 100 mem100	100	100	10, 22
TB 200 mem20	20	200	11, 23
TB 200 mem50	50	200	12, 24
TB 200 mem100	100	200	13, 25

Table 4.1: Experimental runs for *Trace-back* memory agents, compared with *Purely Reactive* agents (PR). Different parameters for *Event-based* (EB) and *Time-based* (TB) are shown in the table, experiments with identifiers 1 to 13 are without noise interference, experiments in identifiers 14 to 25 are with noise interference. MemX is the length of the circularly-ordered memory, Time-step is the period between making entries for the TB architectures. All conditions listed, except for PR (condition 1), are tested with (14-25) and without noise (2-13). In total this results in 25 different experimental conditions.

tions for different types of agents which are uniquely parameterized. For experiments with noise interference, a random noise signal ranging between $-5°$ and $5°$ from a Gaussian distribution with a mean of zero was applied to turning angles.

For a comparison of the *Locality* memory agent and the *Purely Reactive* agent, another two sets of experiments of 50 runs for each type have been carried out in the second environment (see Figure 3.7).

An agent will expire after 9000 timesteps of life span if it does not perform any action in an experimental run for both sets of experiments.

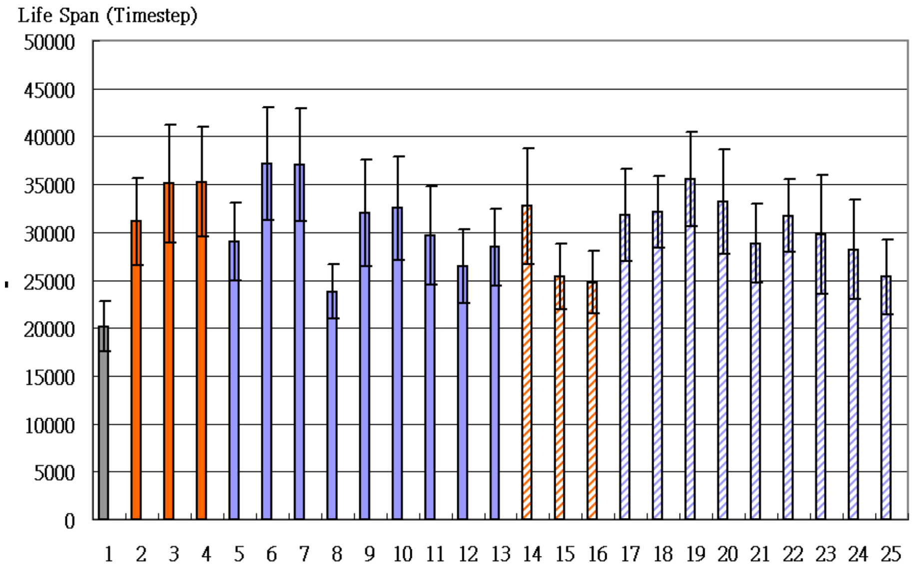

Figure 4.1: Experimental results showing life span of *Purely Reactive* agents and *Trace-back* memory agents. Identifiers are as in Table 4.1. Experimental conditions 14-25 are subjected to Gaussian noise. Error bars show confidence values of the results. Note that the result for PR agent only applies to this experimental study because of the specific settings of both internal variables and environmental resources in this study.

4.1.2 Results

Figure 4.1 and 4.2 illustrate the results of the average lifetime for each set (50 runs) of the experiments. Confidence values are shown as error bars.

4.1.3 Evaluation

The experimental results show that both approaches of autobiographic memory effectively extended the agents' lifetime in nearly all cases. The *Locality* memory agent performs far better than a *Purely Reactive* agent in the specific static environment that we used. The general results without noise interference, both for *Time-based*

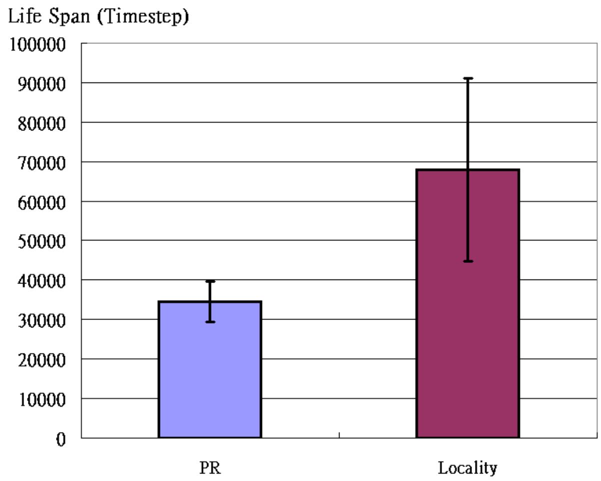

Figure 4.2: Experimental results showing life span of *Purely Reactive* agents vs. Locality memory agents. Error bars shown as confidence values of the results. Note that the result for PR agent only applies to this experimental study because of the specific settings of both internal variables and environmental resources in this study.

and *Event-based* mechanisms of *Trace-back* memory, indicate that agents with a larger number of memory entries have longer lifetimes (except in the *Time-based* mode when the number of time steps of making memory entry equals 200; this exception might be explained by the fact that a very large memory may be cluttered with unimportant events.) With noise interference, this trend is basically reversed. Since a larger number of memory entries allows the accumulation of more noise disturbances, incremental deviations of rotation values can cause the agent to get 'lost' in the environment when memory rotation values are retrieved from memory and applied to the actuators.

Comparing with *Time-based* mode, we speculate that *Event-based* mode is able to capture an event constructed by a sequence of actions more precisely. Particularly in a larger environment, it is necessary for agents to remember only important actions when the capacity of their memory is limited. Therefore, *Event-based* mode is taken for further developments including experiments in multi-agent context (see Section 4.2) and the design of Long-term autobiograhpic memory agents (see Chapter 5).

Furthermore, experiments also reveal a disadvantage of the *Trace-back* mechanism, especially when the agent has a large number of memory entries. In case a useful resource was encountered a long time ago, and the memory entry for this resource occurred much earlier in comparison to the current entry, then the agent could easily die halfway through the trace back process. This might be remedied by cost-benefit analysis by the agent to determine whether it would pay off to use an old memory.

The *Locality* mechanism seems to be an approach for using long-term memory for the agent to remember only information which is significant to its survival in the environment. Nevertheless, in the experiments described in this chapter, object distribution and environmental exploration are limited when applying this approach, and only a static environment has been studied.

Due to differnt environmental conditions for two separate sets of experiment, results for *Purely Reactive* agents in both Figure 4.1 and 4.2 are not the same. Object distributions in the environment for these two sets of experiments are shown in Figure 3.7.

4.1.4 Lessons Learnt

The comparison of the agents' lifetimes shows significant differences between Purely Reactive agents and Autobiographic agents. Our results indicate that, even in simple static environments, autobiographic memories increase the adaptability (in terms of survival) of the agent. We hypothesise that more complex, dynamic and/or social environments will generally require more elaborate autobiographic memory structures. Therefore, in the next chapter (Chapter 5) we designed more sophisticated memory architectures, and a more complex environment with dynamic resource allocation, i.e. the existence of resources will be made less predictable by introducing time varying features to the environment, such as 'seasons'. In order to further investigate the potential of the *Trace-back* memory approach, an improved version of the current *Trace-back* (renamed *Short-term Memory Trace-back*) and a new *Long-term Autobiographic Memory Trace-back* mechanism for autonomous agents are developed and evaluated in the next chapter.

4.2 Narrative Multi-agent Experiments

Based on the experimental framework in the previous section (Section 4.1), which studied different possible mechanisms for developing autonomous autobiographic agents from an agent-centered viewpoint, in this section we investigate the hypotheses that

1. In a multi-agent context, environmental dynamics are complicated by multiple agents wandering around in the same environment. As a result, the interference of encountering other agents is expected to adversely affect the performance of both autobiographic agents and purely reactive agents.

2. Autobiographic agents which communicate memories under appropriate conditions should generally outperform autobiographic agents which do not share memories.

3. If communication of memories has a cost, we hypothesise that communications will be less beneficial than when communication is cost-free.

4. If costs are incurred for the communication of memories, we expect selective strategies for when to communicate to be beneficial to autobiographical agents.

At the outset, we are investigating communications taking place in different situations. Agents acquire potentially useful information of short paths to the resource either 1) any time they have the opportunity to copy memories of another agent – they are sufficiently close to another agent, or 2) if, in addition, the agent's internal variable is lower than a certain threshold ('hunger'). An important factor which must be taken into account in our experiments is if communication costs are incurred each time information is transmitted from one agent to another, due to time and energy factors that impact information exchange among physical agents such as animals or robots.

Although simulation environments in general can provide an agent with full information about events and features in the environment, we deliberately take a bottom-up, agent-centered approach: We do not use any unique object identifiers in the virtual environment so that an agent is not able to identify or recognize specific targets. Furthermore, global coordinate information of distinct objects does not exist for the agents in searching the resource.

4.2.1 Experimental Settings

Experiments for examining the survival of autobiographic agents in the virtual environment can be generally classified into three main types according to use of autobiographical memory and communications: Purely Reactive Agents (*PR*), Autobiographic Agents with no communication (*Mem NoComm*) and Autobiographic Agents with communication (*Mem Comm*). The last type (*Mem Comm*) is again split into two categories that differ in the way the agents communicate: 1) at any time (*Mem Comm*); or 2) only when an agent is hungry (*Mem Comm Hungry*). Each of these two categories can be used in two different conditions: no communication costs attached (*NoCost*), or communication costs attached (*CommCost*). In total, this results in 26 different experimental conditions, each of them tested in 50 runs which take approximately three hours for running on a desktop computer (Pentium 4, 2 GHz, 512 MB main memory). In order to ensure robustness of the results, in each of the 50 runs the agents were initialized with different internal variables and random locations. The 26 experimental conditions are labeled according to Table 4.2

An agent will expire after 13333 timesteps of life span if it does not perform any action in an experimental run.

4.2.2 Results

Figures 4.3 to 4.7 show five comparisons of different experimental sets. In each figure, the y-axis represents the lifespan of the agents and the x-axis represents the number of agents in each set of experiments. Each point in the figure is the average lifespan of 50 runs of each agent in a specific experiment; therefore, multiple points appear in the same column of multi-agent experimental conditions. For clearly illustrating the result, each fitted curve in the figures corresponds to the average value of those points in a particular set of results.

Experiment result for single *Purely Reactive* agent in Figure 4.3 is different from results shown in previous single-agent experiments (see Figure 4.1 and 4.2). It is

Conditon	**PR**	**Mem NoComm**	**Mem Comm NoCost**	**Mem Comm Hungry NoCost**	**Mem Comm CommCost**	**Mem Comm Hungry CommCost**
SA	Fig. 4.3	Fig. 4.3 Fig. 4.4				
MA-2	Fig. 4.3	Fig. 4.3 Fig. 4.4	Fig. 4.4 Fig. 4.5 Fig. 4.6	Fig. 4.5	Fig. 4.6 Fig. 4.7	Fig. 4.7
MA-3	Fig. 4.3	Fig. 4.3 Fig. 4.4	Fig. 4.4 Fig. 4.5 Fig. 4.6	Fig. 4.5	Fig. 4.6 Fig. 4.7	Fig. 4.7
MA-4	Fig. 4.3	Fig. 4.3 Fig. 4.4	Fig. 4.4 Fig. 4.5 Fig. 4.6	Fig. 4.5	Fig. 4.6 Fig. 4.7	Fig. 4.7
MA-5	Fig. 4.3	Fig. 4.3 Fig. 4.4	Fig. 4.4 Fig. 4.5 Fig. 4.6	Fig. 4.5	Fig. 4.6 Fig. 4.7	Fig. 4.7

Table 4.2: Conditions for experimental runs, whose results appear in Figures 4.3 to 4.7. In the first column of the experimental conditions, SA stands for single agent and MA-X for multiple agents where X is the number of agents allocated in the environment for each experimental run.

because the agent has different internal variables in this study and this set of multi-agent experiments runs on a different environment, as shown in Figure 3.15. The environments for single-agent experiments can be found in Figures 3.8 and 3.9.

4.2.3 Evaluation

Experimental results in Figure 4.3 reconfirm our previous hypothesis in Section 4.1, namely that autobiographic memory effectively extends the lifespan of a Purely Reactive agent which is based on the subsumption control architecture, in both single-agent and multi-agent environments. The agents' average lifespan for both Autobiographic and Purely Reactive agents generally decreases as the number of agents in the environment increases, particularly when the number of agents is more

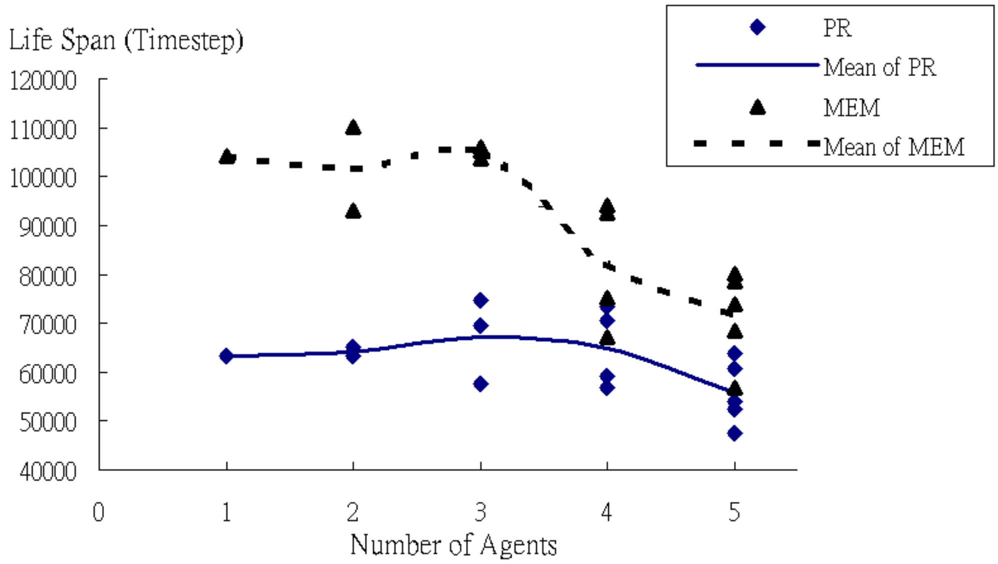

Figure 4.3: Comparing the average lifespan of Purely Reactive agents (PR) and Autobiographic agents (MEM) without communication of memories. Note that the result for single PR agent only applies to this experimental study because of the specific settings of both internal variables and environmental resources in this study.

than three in multiple autobiographic agent experiments, supporting our initial research hypothesis 1 in this section. Since more complicated environmental dynamics are created by an increased number of agents, we find higher probabilities of an agent being disturbed by other agents wandering around in the environment during the Trace-back process when there are more agents.

We hypothesised in the multi-agent communication experiments involving the sharing of memories that agents should be able to receive better paths from the current location to the resource. However, no two agents can occupy the same space and have the same perspective on the environment due to their embodiment. This gives rise to the *displacement problem*, the phenomenon that a neighbouring agent's

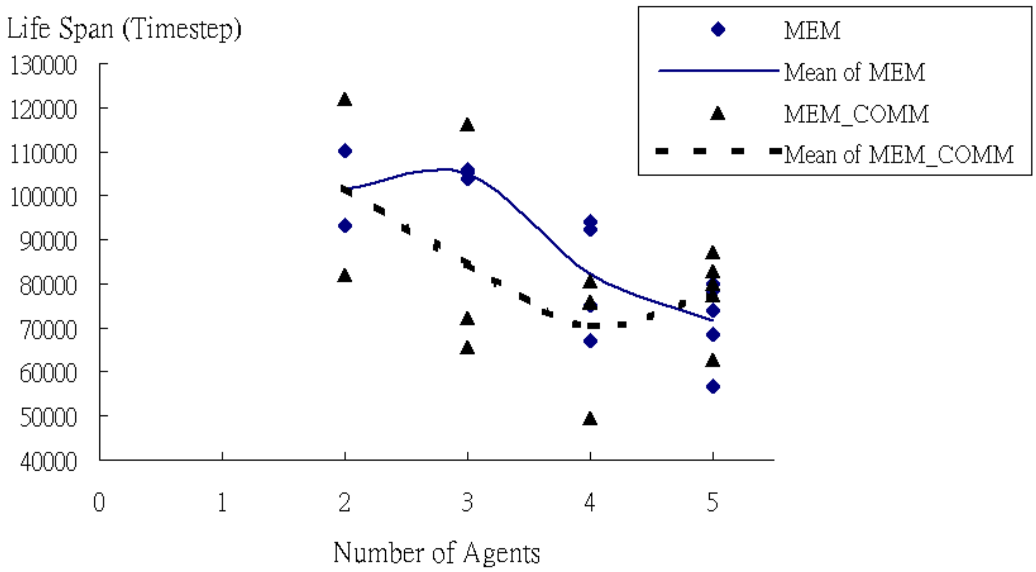

Figure 4.4: Comparing the average lifespan of Autobiographic agents (MEM) vs. Autobiographic agents with communication of memories (MEM COMM). Results of a single agent in MEM are ignored for the comparison with MEM COMM, since communication only happens when two or more agents exist in the environment.

location slightly differs from one's own, which result in a slight error in reusing other agents' paths and leading to lower performance.

From the observation of experiment, we assume that *displacement problem* could be a possible explanation for the results shown in Figure 4.4. This problem reveals the necessity of consideration of embodiment issues in applying memory sharing communication behaviours to autobiographic agents. One solution to this problem could be to allow agents to correct for the different location and perspective of others. Figure 4.4 also indicates that the increasing frequencies of communications in the multi-agent communication experiments with five agents can compensate fairly well for the negative effect of interference resulting in decreasing lifespans in the experiments of Autobiographic agents without communications.

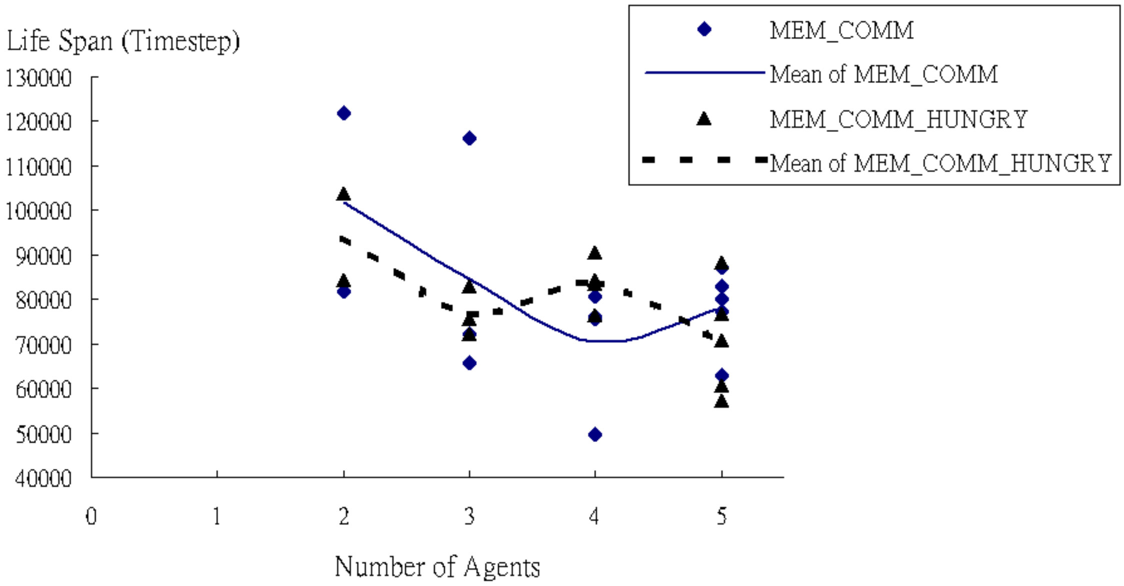

Figure 4.5: Comparing the average lifespan of two types of Autobiographic agents with different communication motivations: communicate memories to other agents any time (MEM COMM) vs. communicate only when recipient is 'hungry' (MEM COMM HUNGRY).

Figure 4.5 illustrates that there was no significant distinction in the multi-agent communication experiments related to the two different motivational conditions (communication any time versus only when the recipient agents are hungry). Similarly, Figure 4.6 shows no significant difference between Autobiographic agents which incurred costs versus no costs for communicating. Possibly the results would be different if the displacement problem had been solved. Nevertheless, when both factors of communication motivations and costs are combined in the experiments whose results are shown Figure 4.7, then there is a tendency showing that when communication incurs costs, agents that communicate with others only when the recipient is hungry perform better than those agents that can communicate any time. This tendency reveals that seeking to use others' episodic memories only when hungry

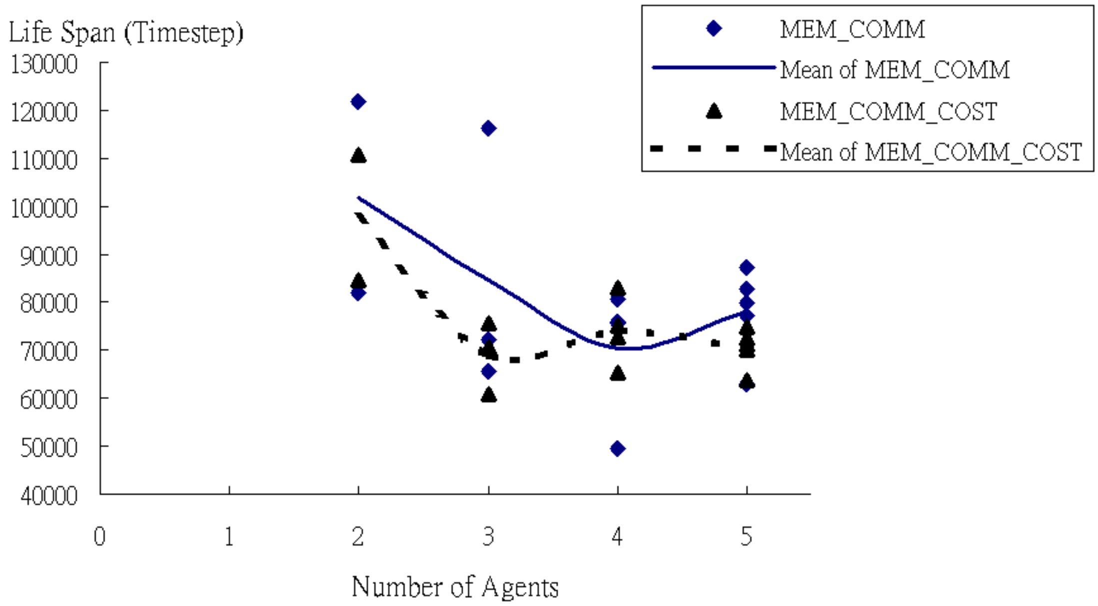

Figure 4.6: Comparing the average lifespan of two conditions of Autobiographic agents with communications: communicate memories with other agents without applying the communication costs (MEM COMM) vs. communicate with other agents applying the communication costs (MEM COMM COST).

prevents unnecessary costs from being incurred and thus extends the lifespan of Autobiographic agents.

4.2.4 Lessons Learnt

Our study provides experimental evidence that within our framework autobiographic agents effectively extend their lifespan by embedding an Event-based memory which describes agents' previous action sequences as compared to a Purely Reactive subsumption control architecture. Multi-agent environmental interference dynamics result in decreasing average lifespan of agents. Although we cannot entirely rule out the possibility that various specific conditions of agent communications might

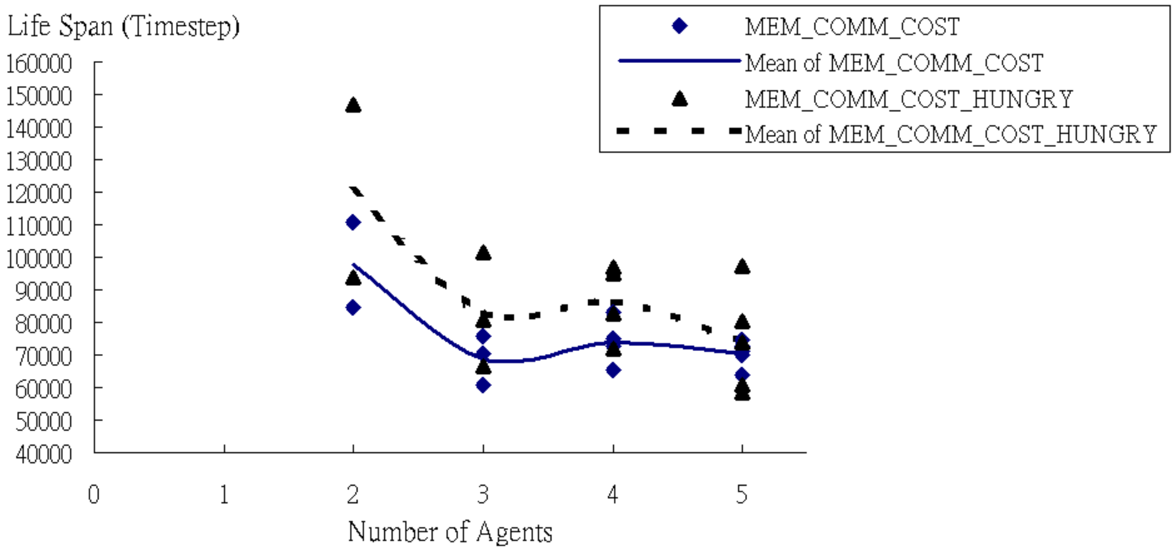

Figure 4.7: Comparing the average lifespan of two types of Autobiographic agents with costs applied to communications and different communication motivations: communication with costs (MEM COMM COST) vs. communicate only when the recipient agents are 'hungry' and applying costs (MEM COMM COST HUNGRY).

be able to improve the overall lifespan of the agents in the environment, some appropriate combinations of factors, e.g. communication motivation and cost factors, result in improved performance.

With the aim of producing a generic adaptive control architecture with the most efficient usage of autobiographic memory to enhance the survival of autonomous agents, the Trace-back process in the current experimental framework needs to be further improved. For example, Event-based memory making mechanisms inspired by the episodic memory of humans can be manipulated by not only using 'undo' in a trace-back process (see Section 4.1), or by re-enacting previous experiences, such as 'day-dreaming' in Dautenhahn and Coles (2001), but also by regularly applying consistency checks whereby agents can validate their location, e.g. through recognizing local landmarks such as obstacles, environment boundaries, as well as responding appropriately to unexpectedly encountered object or agents. These issues together with the use of autobiographical memory in more complex environments and for more complex tasks are addressed in the later research, described in Chapters 5 and 6.

Chapter 5

Long-term Autobiographic Memory Architectures

After the early studies on basic memory architectures for narrative autobiographic agents (Chapter 3 and 4), we continue to pursue our research goal in developing a generic and robust computational memory architecture for autobiographic agents. In this and the next two chapters (Chapter 6 and 7), we show that our recent studies from Ho, Dautenhahn and Nehaniv (2005) and Ho, Dautenhahn and Nehaniv (submitted) with the enhanced computational autobiographic memory architectures and a complex and dynamic environment again confirm our research hypothesis that autobiographic memory can prove beneficial – indicating increases in the lifespan of an autonomous, autobiographic, minimal agent. Furthermore, higher communication frequency brings better group performance for Long-term Autobiographic Memory agents in multi-agent experiments. The research with single-agent experiments (Ho, Dautenhahn and Nehaniv 2005b) has been presented at an international conference, IEEE Congress on Evolutionary Computation 2005 (CEC 2005) in Edinburgh, UK in September 2005; the research with multi-agent experiments (Ho, Dautenhahn and Nehaniv submitted) has been submitted to an international journal: Cognitive Systems Research.

In this chapter we first describe the overall motivation for our further studies in developing computational memory architecture for narrative autobiographic agents. Next we introduce agent embodiment within a large and complex virtual environment, which dynamically changes its conditions and resources distribution in order to generate various types of events and sequences of sufficient complexity for the experiments (Section 5.2.1). Then we focus on three main agent control architectures (Section 5.2.2): Purely Reactive (PR), Short-term Memory (STM) and Long-term Autobiographic Memory (LTM). For each of them we illustrate the design concepts of the architecture in detail. Finally, for the LTM architecture, we specify its main features, including Event Specific Knowledge (ESK), Event Reconstruction (ER), Event Filtering and Ranking (EFR) processes and particularly aspects for narrative story-telling and story-understanding.

5.1 Motivation

We extend our previous work (Chapter 3 and 4) in investigating the performance of different autobiographic memory control architectures which are developed based on a basic subsumption control architecture for Artificial Life autonomous agents surviving in a dynamic virtual environment. In our previous work we showed how autonomous agents' survival in a static virtual environment can benefit from autobiographic memory, using a kind of communication of experiences in multi-agent experiments. In this chapter we extend the existing memory architecture by enhancing its functionalities and introducing Long-term Autobiographic Memory, which is derived from the inspiration of human memory schema – categorical rules or scripts that psychologists in human memory research believe all humans possess to interpret the world (see Section 2.1 in Chapter 2). A large-scale and dynamic virtual environment was created to compare the performance of various types of agents with various memory control architectures: Purely Reactive (PR), Short-term Memory (STM), Long-term Autobiographic Memory (LTM) and Short-term and Long-term Autobiographic Memory (STM+LTM).

Narrative story-telling in a bottom up fashion is another important feature which we develop for autobiographic agents in this chapter. As we have learnt an important lesson from the early study of communicative autobiographic agents (Chapter 4), we take the essential characteristics of story-telling and story-understand into consideration when we are designing the Long-term Autobiographic Memory architectures. In addition to directly reusing all the information from a original story told by another agent, a communicative agent with Long-term Autobiographic Memory is able to selectively choose part of this story which is useful to be combined with an agent's own experience in order to reconstruct a new story for locating a resource. Reconstructing a story in this manner is called *Mixed Reconstruction*, as described in detail in Chapter 7 when the *Observer Interface* is introduced. Sharing stories in a narrative sense for autobiographic agents can result in better 'story qualities', and group adaptation. In the next chapter (Chapter 6) we show that in a group of story-telling agents, higher lifespans are observed.

5.2 Design and Implementation

In this section, we show the enhancements of the design of memory control architectures for autobiographic agents in Chapter 3. In order to validate our new architectures, a large and dynamic virtual environment is created to generate a rich variety of events for agents to experience.

5.2.1 The Complex Environment and Agent Embodiment

In order to create rich possibilities of temporal sequences of events for examining the performance of our agent control architectures and the utility of narrative storytelling features between LTM agents, a large, dynamic and complex 'nature-like' virtual environment has been created using VRML and Java programming languages. This environment is fairly different from other simple and flat agent test-beds since it has various types of resources, most of them dynamically distributed on different kinds of landforms. Figure 5.1 shows the design of the environment in an iconic way. Figure 5.2 illustrates the virtual environment model from two different perspectives. Figure 5.3 demonstrates the related metric information for this environment.

The temporal richness of events generated by the complex environment particularly includes the following two algebraically non-trivial characteristics (Nehaniv and Dautenhahn 1998b): 1) *non-commutativity* – a sequence of events, can be order-dependent, with different effects depending on the specific sequence in which they happen; 2) *irreversibility* – some events cannot be 'undone' ('undo' means trying to realize the previously encountered situation by following actions in reverse order).

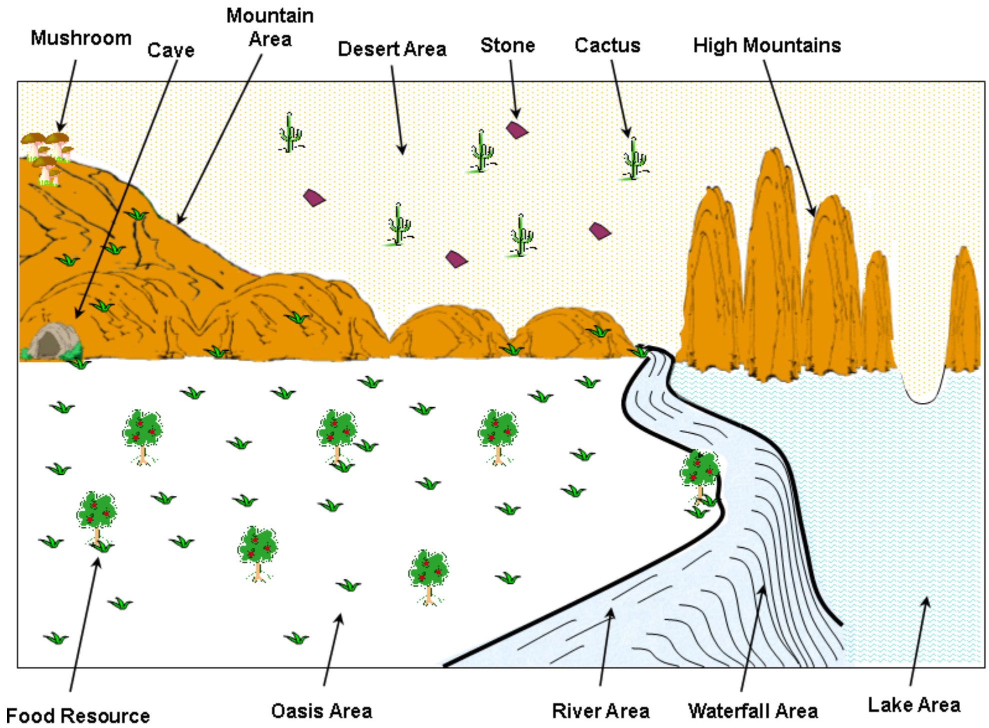

Figure 5.1: The design of the dynamic environment.

Environment Structure

To create this richness of temporal events, each area in the environment has its unique features, illustrated as follows:

- *Oasis* – this is generally a warm and flat area, which has three *Apple Trees* in the summer.

- *Desert* – a hot and flat area which efficiently provides body heat to the agents and has *Stones* and *Cactuses*. Cactus is the only resource for agents to increase their *moisture* in the winter. To crush the Cactus, agents need to pick up a Stone; this is a realization of *non-commutativity* (crush, then pick-up is NOT

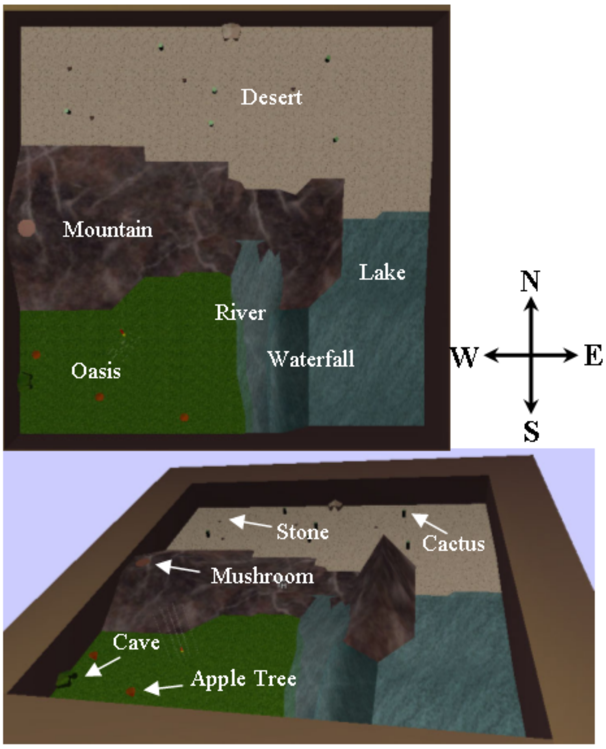

Figure 5.2: The simulated dynamic virtual environment viewed from two different perspectives. Object locations and landform boundaries are slightly different from the original design shown in Figure 5.1.

the same as pick-up, then crush), and agents are able to change the Stone distribution in the environment by randomly carrying or laying down the Stone after they have consumed a Cactus.

- *Mountain* – located between the desert and oasis areas; some edible *Mushrooms* exist permanently on top of the mountain, however, climbing up the mountain takes an extra amount of internal energy from the agents.

- *River* – in the summer, it provides water resource to the agents and is located next to the oasis. Agents are able to swim in the river, but they cannot swim

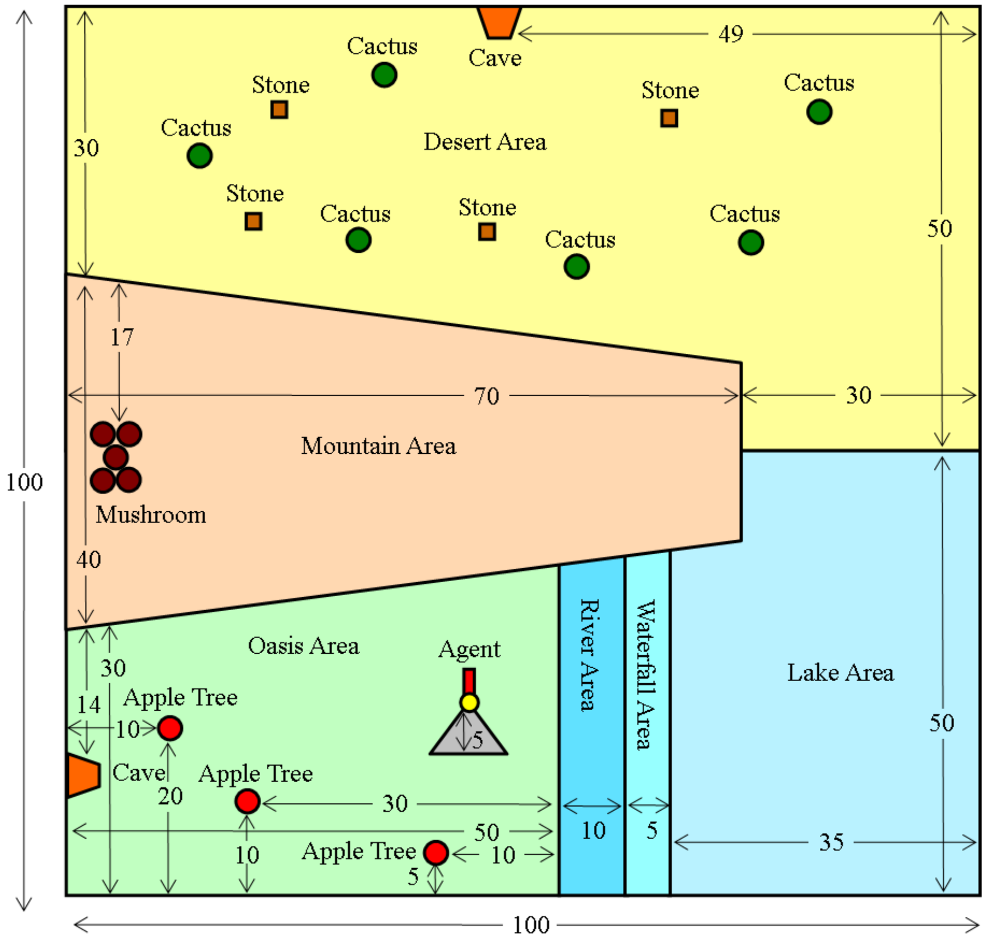

Figure 5.3: Approximate information represented in VRML distance unit for environment size, agents' sensory range and object locations. Note that the positions of Cactus and Stone in the Desert area are randomly generated in each experimental run, and Cave has the function with *Nest* we used in our earlier experimental study.

towards the north since it is against the current.

- *Lake* and *Waterfall* – these provide another source of moisture and environmental complexity. The waterfall connects to the upper river and the lake. Once agents enter the waterfall area, they will be picked up by the down-

	Oasis & Mountain	**Desert**	**River, Waterfall & Lake**	**Resource allocation**	**River accessibility**
Summer	Cool	Hot	Cool	Oasis - Cave, Apple Tree, River, Waterfall, Lake - Water, Mountain - Mushroom, Desert - Cave, Cactus	Flowing (Agents cannot pass)
Winter	Cool	Warm	Cold	Oasis - Cave, Mountain - Mushroom, Desert - Cave, Cactus	Frozen (Agents can pass)

Table 5.1: Environmental heat, resource allocations and river conditions in the dynamic environment

stream current and then fall into the lake area. The passage going to the lake area by passing through the river and waterfall areas can be seen as realizing *irreversibility*, since an agent is not able to either go back to the river from the waterfall or go back to the waterfall from the lake. Agents have to search for the north exit in the lake area to go to the desert area.

- *Cave* – there are two caves in the environment for agents to regain their energy, one located in the oasis area and the other one located in the desert area. Cave has the function with *Nest* we used in our earlier experimental study.

Alternatively, two seasons *Summer* and *Winter*, have been simulated in the environment to have a higher level of environmental dynamics (Table 5.1). Each season has the same duration but different effects on a) the level of temperature in different areas of the environment, b) dynamic resources allocation and c) the accessibility of the river.

The system design of this large and dynamic environment can be found in Subsection B.3.1 in Appendix B.

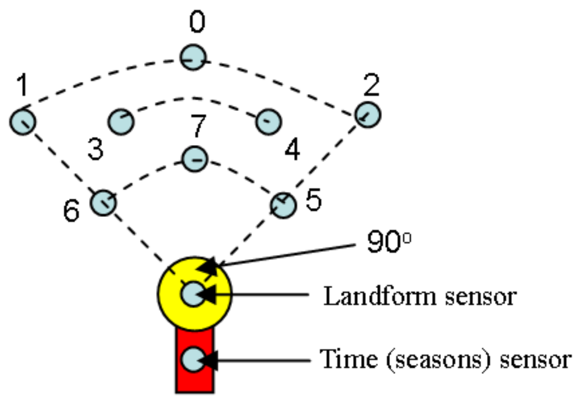

Figure 5.4: Hit-Ray sensors 0 - 7 for sensing both objects and Landforms, the agent body has a landform sensor and a time sensor.

Agent Embodiment

All agents in the dynamic environment are virtually embodied with the same body size and sensors. They are equipped with nine external sensors: seven Hit-Ray sensors (Blaxxun 2004) form a 90 degree fan-shape for detecting the objects, landforms, as well as the environment heat from different types of landforms; the agent body has a landform sensor and also a time sensor for sensing the current season of the environment. Figure 5.4 shows the distribution of these sensors.

All agents are designed to have a finite lifespan and are required to wander in the environment as their basic behaviour. The survival of an agent depends on maintaining homeostasis for its four internal variables, namely *glucose*, *moisture*, *energy* and *body temperature*. Internal variables *glucose*, *moisture* and *energy* are initialized close to a maximum value at the start of each experimental simulation run and can be increased by taking different types of resources in the environment. Variable *body temperature* is initialized to be acceptable, that is half of the maximum value, and needs to be maintained between maximum and minimum values by regularly wandering in different areas in the environment. Each translation or rotation of the agent will reduce the internal variables *glucose*, *moisture* and *energy* by a certain value. When the internal variables *glucose*, *moisture* and *energy* drop below a threshold, which is half of the maximum value, then the agent begins searching around for resources dynamically located in the environment. When *body temperature* goes beyond the acceptable range – lower than 30% or higher than 70% of the maximum value, the agent needs to move to an appropriate area to maintain *body temperature* until it comes back to the acceptable range again. If the value of one of the internal variables (*glucose*, *moisture* and *energy*) is less than a particular minimum value, or *body temperature* reaches the minimum or maximum value, then the agent will die. The experimental parameters (thresholds etc.) that allow the agents to live in the virtual environment, but eventually die, were determined in initial tests.

The relationship between internal variables and various types of resources in the environment are shown in Table 5.2.

5.2.2 Agent Memory Control Architectures

We aim to develop appropriate autobiographic memory architectures on top of a basic subsumption control architecture in order to a) enhance the agents' performance in surviving in a dynamic environment and b) share meaningful information as stories between autobiographic agents. To achieve the first goal, we designed and implemented three different control architectures: Purely Reactive (PR), Short-term Memory (STM) and Long-term Autobiographic Memory (LTM). In addition to these three architectures, in the experimental Chapter (Chapter 6) we also investigate the

Internal variables	Relevant resource (effects)
Glucose	Apple Tree (+ 100%) Mushroom (+ 100%) Cactus (+ 10%)
Moisture	Apple Tree (+ 100%) Cactus (+ 10%)
Energy	Cave (+ 100%) Cactus - touch without stone penalty (- 10%)
Body temperature (in each simulation time step)	Desert - Summer (+ 0.0015%) Desert - Winter (+ 0.0005%) Oasis - Summer (- 0.00025%) Oasis - Winter (- 0.0005%) Mountain - Summer (- 0.00025%) Mountain - Winter (- 0.0005%) Water Area* - Summer (- 0.0005%) Water Area* - Winter (- 0.0015%) **Water Area = River, Waterfall and Lake*

Table 5.2: Relationships between agents' internal variables and different resources and contexts in the environment.

fourth type, which is built by combining STM and LTM into one architecture in order to broaden the agents' temporal horizon (Nehaniv and Dautenhahn 1998a, Nehaniv 1999, Nehaniv, Polani, Dautenhahn, te Boekhorst and Cañamero 2002) by taking advantage of more sophisticated memory control algorithms. For both STM and LTM architectures, their distinctive memory layers of control algorithms are built on top of the PR architecture with the aim of inhibiting the execution of behaviour from the PR architecture optionally. In order to carry out the second goal for narrative story-telling between Long-term Autobiographic Memory agents (LTM), we established the 'communication protocols' for story-telling and story-understanding features between LTM agents and enhanced the Event Filtering and Ranking (EFR) processes in selecting events to be re-executed among events experienced by the agent itself and told by other agents.

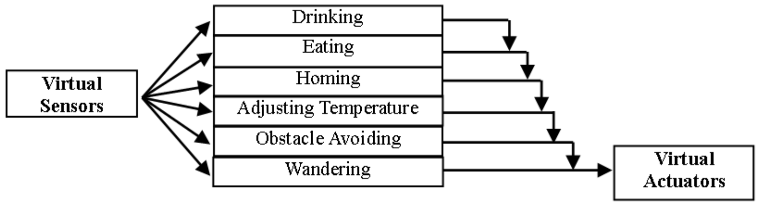

Figure 5.5: behaviour hierarchy which is based on the subsumption architecture for a Purely Reactive (PR) agent. It addes the behaviour 'Adjusting Temperature' comapred with Figure 3.2.

Purely Reactive Architecture (PR)

Similar to the previous study in Chapter 3, we define a PR agent in this chapter as an agent who makes its decisions for executing behaviour totally based on its internal variables and sensory inputs. Therefore we designed and implemented the PR agent by using a basic subsumption control architecture (Brooks 1986), as illustrated in Figure 5.5. The architecture of the PR agent includes six layers. Higher-level behaviours inhibit or override lower-level behaviours. The agent usually wanders around in the environment by executing the bottom layer in the architecture. Descriptions in detail for the basic behaviours can be found in the previous study in basic memory control architectures (see Sub-section 3.1.1 in Chapter 3).

The system design of the PR architecture can be found in Sub-section B.3.2 in Appendix B.

Short-term Memory Architecture (STM)

As our goal is to improve the design of the *Trace-back* architecture in the previous study (see Sub-section 3.1.1 in Chapter 3), we develop STM as a memory module – an adjustable one-dimensional array – on top of the subsumption architecture

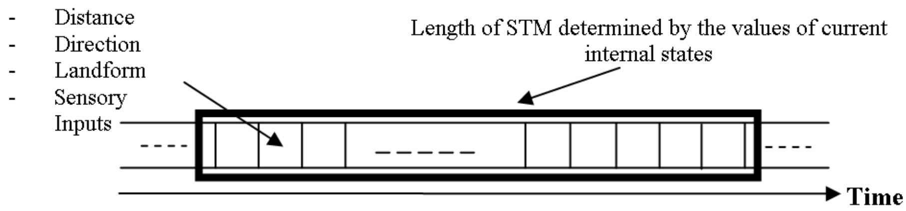

Figure 5.6: Short-term Memory (STM) with information indicating contents of each entry and the change of its length.

with *Event-based* memory entry making mode and *Trace-back* process. *Event-based* memory entry making mode means that introduction of new entries occurs each time the agent experiences an event, i.e. encounters either an object or agent, enters a new area, or changes its current behaviour. Each memory entry includes the current *Direction* the agent is facing, the kind of *Object* encountered by the agent (if any), the current *Landform* the agent now locates on, and how far the agent has traveled (*Distance*) since the last event. This information is inserted at the current position of the index into the memory table, which has finite length restrained by the current internal variables but indefinite index number. The abstract model of STM is shown in Figure 5.6.

Similar to the Trace-back process we introduced in the early study (for details, see Sub-section 3.1.1 in Chapter 3), the STM *Trace-back process* will be triggered if one of the internal variables of the agent is lower than the threshold and the table of STM entries has at least one useful entry which indicates that the agent has previously encountered a relevant resource or a landform. During the Trace-back process, we have also introduced a random noise signal ranging between $-5°$ and $5°$ from a Gaussian distribution with a mean of zero to slightly alter the Direction value when the agent is retrieving an entry from its STM. We introduced this noise in order to account for the imperfection of memory retrieval. Therefore, there are possibilities that the resource is not available at this location since 1) some resources in the environment are dynamically distributed; or 2) the actual rotation and distance value in each entry might have been slightly distorted by accumulated errors created by the noise during the Trace-back process. As a consequence of these accumulated errors the agent might not be able to finish the Trace-back process, which is terminated if the agent collides with any other object or agent in the environment.

After an agent performs a Trace-back process, the result will be either: target is found or target is not found; in both situations, those undone entries will be cleared and the agent will start making new entries from that point. When an STM agent faces the environmental dynamics, such as unstable resource distributions and the flowing direction of the river and waterfall in summer, these sometimes cause the agent to fail in executing the Trace-back process, where the agent will erase all the memory entries in its STM. An STM agent cannot remember an unlimited number of entries. The number of entries in STM is determined by estimating the costs of executing the Trace-back process of undoing all existing entries. If the cost for one of the internal variables is higher than the current value, then the length of STM will be shrunk by deleting the earliest entries since the agent is not able to afford the cost of doing the Trace-back, as illustrated in Figure 5.6. The processes of erasing undone entries and dynamically shrinking the length of STM can be seen as an improvement from the previous work (Ho et al. 2003, Ho et al. 2004) as in Chapters 3 and 4.

The system design of the STM architecture can be found in Sub-section B.3.3 in Appendix B.

Long-term Autobiographic Memory (LTM)

Inspired by human long-term memory (Alba and Hasher 1983) and autobiographic memory models from related research in psychology (Conway 1992), we developed a more sophisticated LTM architecture, which addresses our fundamental research issue: autobiographic memory. In this LTM architecture, we are interested in investigating how the Event Reconstruction (ER) process in LTM can be beneficial when the agent recalls all possible past events from its Event Specific Knowledge (ESK). Also, we are proposing a method for how an event can be eventually selected from numerous reconstructed events in the filtering and ranking processes.

Event Specific Knowledge (ESK) An LTM agent surviving in the dynamic environment has a long list of 'histories', which contains records of situations, called Event Specific Knowledge (ESK). Similar to the Event-based memory entry making mode in STM, each record in ESK is a situation of a particular moment when the agent tries to remember the event context – in this case, the objects and the landform of its surrounding environment and its internal variables; the name of each field in a LTM record and sample entries are shown in Figure 5.7. Due to the special acceptable range of *Body Temperature* an agent needs to maintain, the change of this internal variable is not to be remembered in ESK by an LTM agent.

Some records, which individually describe special situations about the environment, are noticed by the LTM agent as environmental rules. These records have their unique combinations of various keys: *Condition 1, Condition 2, Match Key* and *Search Key*, where *Search Key* indicates a resource that can be obtained if *Condition 1* and *Condition 2* hold in the area specified by *Match Key.* By recognizing and remembering these environmental rules, an LTM agent can enhance the preci-

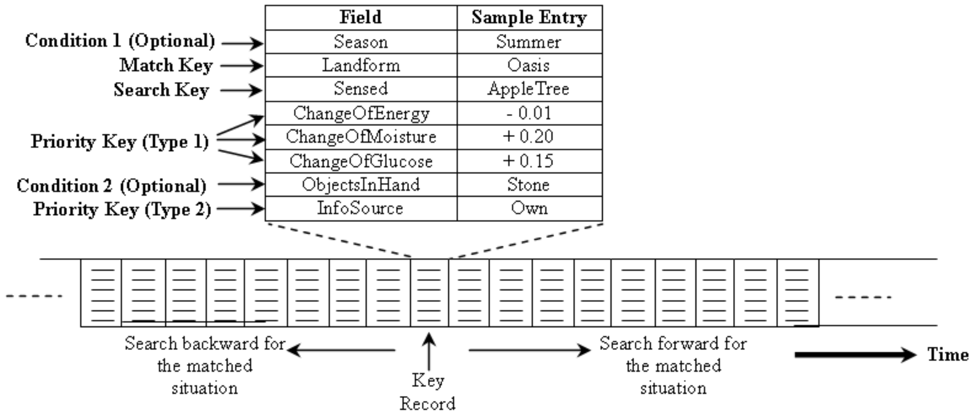

Figure 5.7: Event Specific Knowledge (ESK) of Long-term Autobiographic Memory (LTM).

sion when filtering out events in the Event Filtering and Ranking (EFR) processes. More details are provided in Sub-section 5.2.2.

Event Reconstruction (ER) Process The Event Reconstruction process proceeds as follows: if one of the internal variables of the LTM agent is outside acceptable range, the agent will search all records in its LTM and retrieve at least one relevant event. In order to form groups of events taking place in different periods of time, and also regarding different types of resources or landform, the ER process retrieves a certain number of records from ESK and reconstructs each event by using the 'meaningful' *Search Key* (Figure 5.7). Meaningful here refers to the fact that *Search Key* contains the resource which is needed by the agent in that moment to fulfill one or more than one of its internal variables. Then it will recognize the possible sequence of how an event should be organized - a *Redo* event can be used to repeat a previous situation by executing actions in the original order, in contrast an *Undo* event matches situations that happened in the past which can be reached

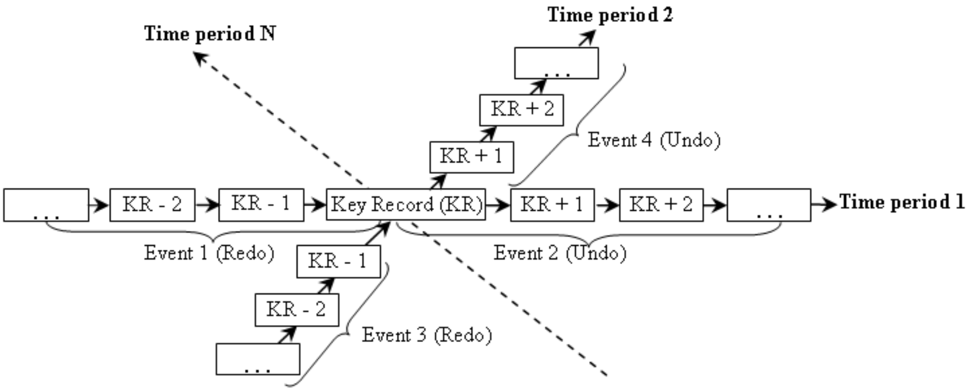

Figure 5.8: Result from Event Reconstruction (ER) process – autobiographic memory schema.

again by executing inverses of actions in reversed order, i.e. undoing each action.

Deciding the appropriate length of each event, which means how many records are related to a specific event, is one of the important processes during Event Reconstruction process. The *Key Record* (Figure 5.7) contains the appropriate *Search Key* to indicate one of the target resources for satisfying the current internal needs of the agent. The length and the final situation of a *Redo* or *Undo* event are recognized by checking the *Match Key* in the *Key Record* to find out the situation that is most appropriate to the current one. Checking the *Match Key* can be done in both directions, searching backward for a *Redo* event and forward for an *Undo* event according to the time. Figure 5.8 shows, after all possible events have been reconstructed by records from ESK, an autobiographic memory schema dedicated for satisfying a specific internal variable.

With regard to the dynamic virtual environment introduced in Section 5.2.1, all possible events which are generated by the environment and can be remembered by the agent in its LTM are classified in Table 5.3.

Possible Event	Effect to Internal Variable	Target (Resource, Object or Location)	Environmental Condition
Looking for Apple Tree	Glucose(I), Moisture(D)	Apple trees in Oasis area	Summer only
Looking for River	Moisture(I)	River	Summer only
Looking for Lake	Moisture(I)	Lake	Summer only
Looking for Cave	Energy(I)	Cave in Oasis area and Desert area	
Looking for Mushrooms	Glucose(I)	Mushrooms	Climbing up Mountains (Energy (D))
Eating Cactus	Glucose(I), Moisture(I)	Cactus in Desert area	With a stone in hand
Hurt by Cactus	Energy(D)	Cactus in Desert area	No stone in hand
Picking Stone	Stone(Picked)	Stone in Desert area	
Location of Mountain area	Glucose(I) (from Mushroom)	Mountain Area	
Location of Oasis area	Glucose(I), Moisture(I) (from Apple Tree)	Oasis Area	
Location of Desert area	Glucose(I), Moisture(I) (from Cactus), Stone (picks up)	Desert Area	
River water flow	Energy(D), Moisture(D), Glucose(D) (gets stuck)	River	Summer only
Irreversible Waterfall	Energy(D), Moisture(D), Glucose(D) (gets stuck)	Waterfall	

Table 5.3: Possible events for Long-term Autobiographic Memory (LTM) agent to remember (I: Increase, D: Decrease).

Event Filtering and Ranking (EFR) Processes After an LTM agent survives for a certain period of time and wanders around different areas in the environment, its ER process is able to produce groups of events when it needs to retrieve an appropriate event from its LTM. Therefore in the next stage we add Event Filtering and Ranking (EFR) processes to 1) filter out inappropriate events by applying environmental rules learnt from situations when the agent was surviving in the dynamic environment, and then 2) rank the remaining events by measuring their significance to the agent.

The first step of EFR processes is searching for the *instantaneous context*, where the situation the agent is currently facing fully matches the target situation specified by the *Key Record*; in this case, the agent will directly execute the *LTM Trace* behaviour (*Redo*ing a sequence of actions with length zero) and just wander around in the same area and wait for the target object to appear. If the current situation does not match any target situation, in the second step of EFR processes some

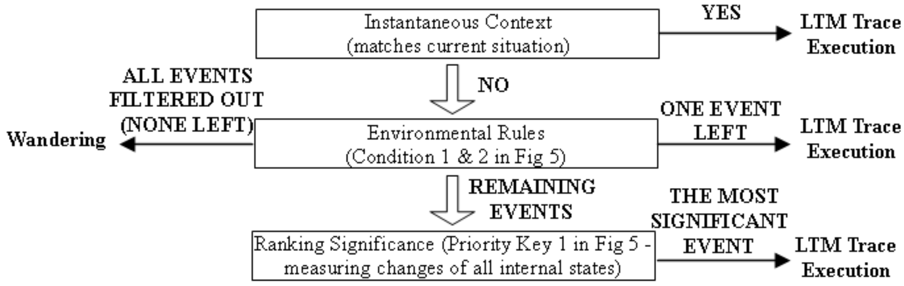

Figure 5.9: Event Filtering and Ranking (EFR) processes.

events which are inappropriate to the current situations will be filtered out by using environmental rules (shown as *Condition 1 & 2* in Figure 5.7). If there is more than one event left after the filtering process, a ranking process will choose the most significant event (shown as *Priority Key (Type 1)* in Figure 5.7) to do the *LTM Trace*. The most significant event is calculated by measuring the total change of internal variables *glucose*, *moisture* and *energy*. EFR processes are illustrated in Figure 5.9.

To execute an *LTM Trace* (either a *Redo* or *Undo* event), the agent will try to achieve the next situation from the current situation, until it reaches the target one. For example, once an LTM agent wandering in the *Oasis* area needs to find *Cactus* in the *Desert* area, this agent follows the reconstructed event experienced in the past, which indicates that in order to reach the desert area, the agent will need to go first to the mountain area, and then to the desert area. Before it can consume the cactus, the event also indicates that the agent should have a *Stone* to crush the *Cactus*; therefore it searches for a *Stone* after it reaches the desert area.

The system design of the Long-term Autobiographic Memory architecture can be found in Sub-section B.3.4 in Appendix B.

Narrative Story-telling for Long-term Autobiographic Memory Architecture

Due to the flexible design offered by LTM architecture – remembering qualitative situations needed for reconstructing events, we experimentally defined new 'communication protocols' intended for approaching emergent narrative story-telling and understanding between multiple LTM agents. This differs from copying the entire memory contents as atomic communications from other agents as presented in (Ho et al. 2004) and in Chapters 3 and 4.

The way communication is established between two LTM agents can be described as 'telling the best story' from the perspective of a story-teller and 'receiving as own experience' from the perspective of a story-receiver. 'Telling the best story' means that, for example, after agent A receives a story request from agent B about finding resources in fulfilling at least one internal variable, and if agent A has encountered any related useful resource previously, then agent A will become the story-teller in this case. As a story-teller, agent A then handles this request from agent B as its own demand, so it tries to execute ER and EFR processes – retrieving meaningful information and reconstructing events from ESK, filtering and ranking these events according to their significance. If finally an event gets successfully generated, agent A will then offer this event as its best story to agent B. Afterward, as a story receiver, agent B accepts the story reconstructed by a sequence of situations from agent A and remembers it as own experience – regarding this sequence of situations as new situations experienced by itself and stores it in its ESK. Although this sequence of situations is specifically transferred from agent A to agent B and it certainly has useful information for agent B to find out a necessary resource in order to fulfill one or more agent B's internal variables, it does not have any priority to be recalled

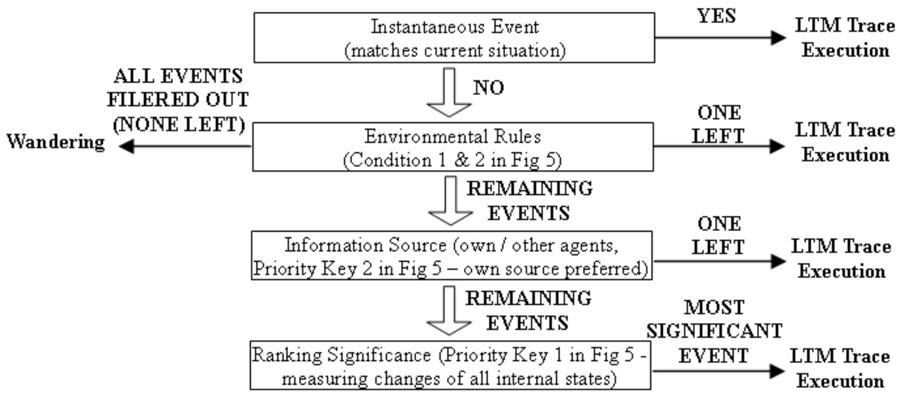

Figure 5.10: Advanced Event Filtering and Ranking (AEFR) processes for communicative LTM agents.

later when agent B needs to recall useful information from its LTM.

By having the ability to accept other agents' stories, communicative LTM agents now can store not only their own experiences but also others' in their own ESK. There is no displacement problem. While the amount of useful information represented in situations is vastly increased, this in fact creates the additional complexity in selecting which event should finally be re-executed from reconstructed events for *LTM Trace* process. Therefore, EFR processes of communicative LTM agents need to be advanced in order to filter and rank various reconstructed events from different agents. Compared with Figure 5.9 from the previous sub-section, Figure 5.10 shows the Advanced Event Filtering and Ranking (AEFR) processes for communicative LTM agents which now in addition include an extra process 'Information Source'. If there are events existing in the autobiographic memory schemata generated by the ER process of the agent and they are reconstructed from their own experiences and other agents' experiences, this process selects events completely reconstructed by agent's own experiences.

The system design of the Long-term Communicative Autobiographic Memory architecture can be found in Sub-section B.3.6 in Appendix B.

Chapter 6

Experiments for Long-term Autobiographic Agents

In this chapter, we carry out two separate sets of experiments to measure the performance of different types of agent control architectures we developed in the Chapter 5. In single-agent experiments, we examine four different architectures: *Purely Reactive* (PR), *Short-term Memory* (STM), Long-term Autobiographic Memory (LTM) and Short-term Memory and Long-term Autobiographic Memory (STM+LTM) essentially by two quantitative measurements: lifespan and maintenance of internal variable. We also give examples to show how architectures with Long-term Autobiographic Memory (LTM) can effectively maintain the internal states in the acceptable ranges. Moreover, different control architectures influenced by the dynamic environmental conditions are discussed, and the combination of Short-term Memory (STM) and Long-term Autobiographic Memory (LTM) produces the best result in average lifespan coping with dynamic environmental conditions in a large-scale virtual environment. In multi-agent experiments, lifespan together with overall communication and storying receiving frequencies are measured to show that communicative LTM agents can benefit from narratively structured communications.

With the Observer Interface we developed this work, which will be introduced in the next chapter(Chapter 7), in single-agent experiments each agent's behaviours can be observed and analyzed. Therefore, the behavioural patterns generated from each agent control architecture are provided to give further explanations for experimental results. In experiments with multiple communicative LTM agents, each agent's dynamically changing LTM contents are able to be monitored in order to analyze:

1. New events which are generated and organized from the Event Reconstruction (ER) process.

2. The effectiveness of reusing events from other agents.

Similar to the experimental setting in the previous work (Chapter 4), unique object identifiers and global coordinate information are unavailable to all types of agents in the virtual environment. Consequently it is difficult for agents to 'track down' the resources.

6.1 Single-agent Experiments

By extending the previous experimental studies (see Chapter 4), in this set of experiments we measure not only the life span of a single agent surviving in the dynamic environment, but also show how well a memory control architecture helps an agent to maintain its internal variables in the acceptable range.

6.1.1 Experimental Settings

To measure the performance of four types of agent architectures: PR, STM, LTM and STM+LTM running in the dynamic virtual environment, we carried out 10 experimental runs for each architecture; each run takes approximately 20 minutes on a Pentium 4, 2 GHz PC with 512 MB Ram. For the fourth type STM+LTM control architecture, we have arranged the STM to have higher priority to execute its *Trace-back* process than *LTM Trace* in the sense of decision making. The system design of the STM+LTM architecture can be found in Sub-section B.3.5 in Appendix B. The starting position for all agents in the experiments is in the center of the oasis area. At the beginning of each experimental run the agent performs a random rotation.

Apart from the main measured dependent variable – the average lifespan in 10 experimental runs of each agent control architecture, we also observe and measure the capability of each architecture in keeping the agent's internal variables within the acceptable range. Therefore, in each experimental run we recorded the change of all internal variables over time. We expected that a desirable control architecture for agents surviving in a highly dynamic environment should be able to maintain all internal variables in the acceptable range. In this study, this means that most of the time internal variables *glucose*, *moisture* and *energy* should be kept at a level higher than a threshold (half of the maximum value).

An agent will expire after 66666 timesteps of life span if it stays in a *non-mountain* (flat area) area and does not perform any action in an experimental run. If an agent is moving on the mountain area all the time, it will have approximately 26666 timesteps of life span. The period of time for each season is 40000 timesteps. These figures also apply to experiments with multiple communicative agents in the next section (Section 6.2).

6.1.2 Results

Figure 6.1 shows lifespans with confidence values[1] of four types of agents. Since we are also interested in observing each agent's comprehensive behaviour generated from its unique control architecture, Figures 6.2 and 6.3 illustrate, as examples, how well all four types of agents maintain their internal variables and Figure 6.4 shows a quantitative measurement for each variable staying outside the acceptable range over lifespan for each type of agent – for each variable, the lower the value is, the better maintenance of variables the agent has.

Experiment result for *Purely Reactive* agent in Figure 6.1 is different from results shown in previous studies (see Figures 4.1 , 4.2 and 4.3). The reason is that this PR agent has different internal variables and this set of experiments runs on a new environment, as shown in Figure 5.1 in Chapter 5.

6.1.3 Evaluation

Figure 6.1 shows significant results: namely that the average lifespan of the LTM agent and the STM+LTM agent outperform the PR agent, which implies that having LTM helps agents to be more adaptive in the sense of surviving in the highly dynamic environment. However, the performance of the *Trace-back* process from the STM agent is sometimes affected by the environmental dynamics, such as the seasonal resource distributions. Therefore the average lifespan of the STM agent, with a high confidence value, cannot be considered as outperforming the PR agent. Although from time to time the STM agent with *Trace-back* process is able to precisely undo all actions of an event and come back to the resource which was encountered previously.

[1] Confidence value is the mean of standard errors, it shows that 95% of the average of experiment results can be obtained from the confidence value interval.

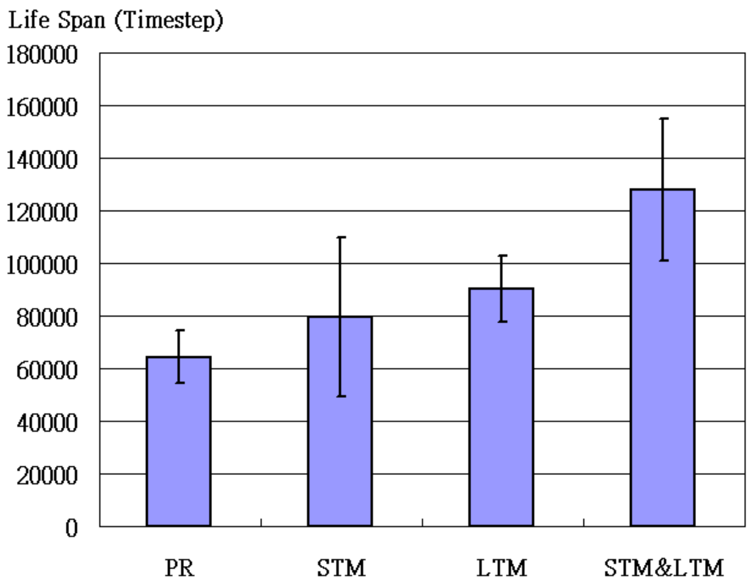

Figure 6.1: Experimental results with confidence values (error bars) showing the average lifespan of the 4 different agent control architectures: 1) Purely Reactive (PR), 2) Short-term Memory (STM), 3) Long-term Autobiographic Memory (LTM) and 4) Short-term Memory and Long-term Autobiographic Memory (STM+LTM) running 10 times in each condition in the environment. Note that the result for PR agent only applies to this experimental study because of the specific settings of both internal variables and environmental resources in this study. Therefore the result is different in comparing with Figures 4.1, 4.2 and 4.3.

The agents with STM+LTM have the highest average lifespan; this result is reflected in agents' memory control architecture as it combines the precision offered by the *Trace-back* process from STM and the flexibility of LTM to cope with the environmental dynamics. Furthermore, agents with LTM appear to be capable of maintaining their variables, except *Body Temperature*, in the acceptable range most of the time, compared to PR and STM agents, as examples show in Figures 6.2 and 6.3. In addition to these examples, a quantitative measurement is shown in Figure 6.4. The reason is that STM agents need to spend a certain amount of

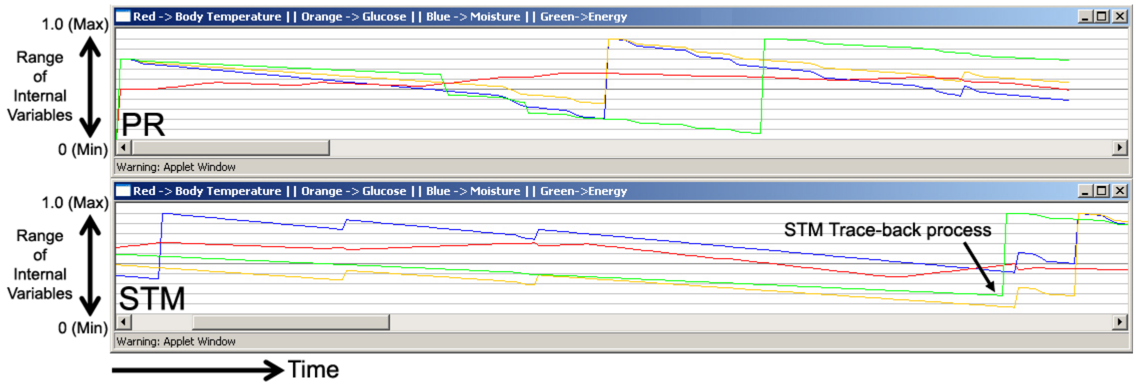

Figure 6.2: Examples of internal variables' changes of a Purely Reactive (PR) agent (upper graph) and a Short-term Memory (STM) agent (lower graph), in time window of length 25000 steps.

Figure 6.3: Examples of internal variables' changes of a Long-term Autobiographic Memory (LTM) agent (upper graph) and a Short-term Memory plus Long-term Autobiographic Memory (STM+LTM) agent (lower graph), in time window of length 25000 steps.

time and energy to execute the *Trace-back* process in order to reach the target resource or landform, as indicated in Figure 6.2. On the other hand, results from Figure 6.4 shows that agents with LTM (for both LTM and STM+LTM agents) perform better in maintaining the *Energy* and overall internal variables (any one of

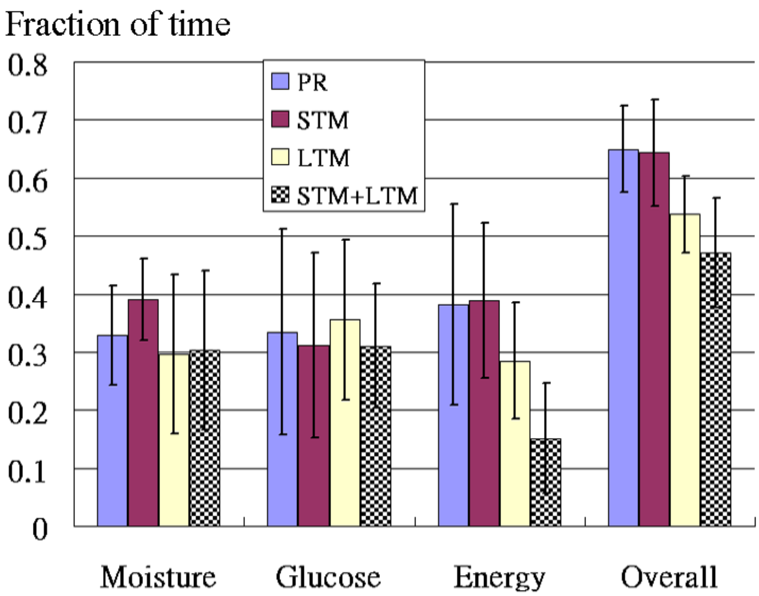

Figure 6.4: Fraction of time of each variable (*Moisture*, *Glucose* and *Energy*) and overall (any one of them) spent outside the acceptable range over lifespan for each type of agent control architecture, error bars are applied as confidence values of the results.

Moisture, *Glucose* and *Energy* is outside the acceptable range) during their lifetime. This result can be seen as reflecting the resource distributions in the environment, since resources providing *moisture* and *glucose* are often easy to find in various areas of the environment, so agents with reactive control and randomly moving around are able to encounter these resources. However, when agents need to look for the *Cave* to recover the *Energy*, LTM could provide a certain level of 'clue' for them to search around the relevant areas as *Cave* is relatively difficult to find in the environment.

Compare this simulation to the previous work (Ho et al. 2003, Ho et al. 2004) in Chapter 3 and 4, in which we studied a single PR or STM agent surviving in a flat and static virtual environment with constant resource distributions. Here results show that LTM agents with a sophisticated autobiographic memory architecture

– inspired by human memory research in psychology – can survive and cope with events in a dynamic and temporally rich environment with the characteristics of *irreversibility* and *non-commutativity.* Experimental results and observations showed that the mechanisms for guiding behaviour executions from PR and STM agents tend to be too simple for the dynamically changing environment.

On the other hand, after LTM agents learnt some environmental rules by experiencing them, such as that climbing up the mountain or getting stuck in the lake area will be more detrimental to the levels of its various internal variables than wandering in other areas, they tend to stay wandering in the area where they can find the necessary resource to maintain their internal variables in the acceptable range. The process of ranking event-significance also helps the agent to avoid going to areas highly costly for internal variables.

Finally, comparing with STM *Trace-back*, the process of LTM *Trace* keeps the agent's choice open towards all other types of resources. When accidentally sensing other resources rather than the target one decided by the ER and EFR processes, the agent will firstly pick up that resource and then continue the LTM trace, if still necessary, by again executing ER and EF processes to check out its current needs. Moreover, in each fixed period of time, the status of *LTM trace* will be updated in order to 1) cope with some target objects which are difficult to find in the area and 2) switch to other targets for fulfilling the same or another internal need.

6.1.4 Lessons Learnt

Through experimental results on agents surviving in the dynamic virtual environment with different memory control architectures (PR, STM, LTM and STM+LTM), we confirmed that a more sophisticated Long-term Autobiographic Memory control architecture effectively extends a PR agent's lifespan and increases the stability reflected in the changes of internal variables over time. We have also shown the design of improved STM and LTM control architectures in detail. The combination of them produces the best average lifespan in single-agent experiments coping with dynamic environmental conditions in a large-scale virtual environment.

From the observations on behavioural patterns produced by the agent with both STM and LTM, the current architecture could be improved to get a better coordination of functionalities of STM and LTM; for example, a motivation-based decision making mode, which has been studied preliminarily in (Ho, Avila-García and Nehaniv 2005a) by comparing the performance between Voting-Based, Winner-Take-All and Static-Threshold architectures in a similar virtual environment, can be used to solve the conflicts between the executions of output from STM and LTM. Motivation-based decision making has not been investigated in conjunction with the STM and LTM architectures.

6.2 Narrative Multi-agent Experiments

In this set of experiments with multiple autobiographic agents surviving in the dynamic environment, we investigate the following:

- How a communicative autobiographic agent performs story-telling and story-understanding.
- How telling and receiving stories improve the overall performance of agents in a group.
- What kind of story contents can emerge when a communicative autobiographic

	2 Agents	3 Agents	4 Agents	5 Agents
Non-Communicative	5 Runs	5 Runs	5 Runs	5 Runs
Communicative	5 Runs	5 Runs	5 Runs	10 Runs

Table 6.1: Experimental settings for measuring the narrative storying features from multiple LTM agents.

agent re-organises experiences from its own and other agents (see detailed observations studied in Chapter 7).

6.2.1 Experimental Settings

To investigate the influences on agents' performance from having a narrative story-telling structure based on a LTM control architecture, we studied the settings shown in Table 6.1 for multi-agent experiments in the same dynamic virtual environment. The starting position for all agents in the experiments is in the center of the oasis area. The agents' orientations and positions within that area are randomly chosen but the same for each run. Due to random avoidance behaviour the agents quickly disperse in the area. We measured the story-telling frequencies, including stories told to other agents and stories received from other agents for each agent individually together with the lifespan in each experiment.

Due to the computational constraints, the maximum number of multiple agents surviving in the virtual environment test-bed is five and each experimental run takes approximately two hours in the setting with five agents. As we are particularly interested in studying the emerging narrative story-telling between LTM agents with relatively high communication frequencies, we carried out ten experimental runs for the setting of five communicative LTM agents while other settings are based on five runs.

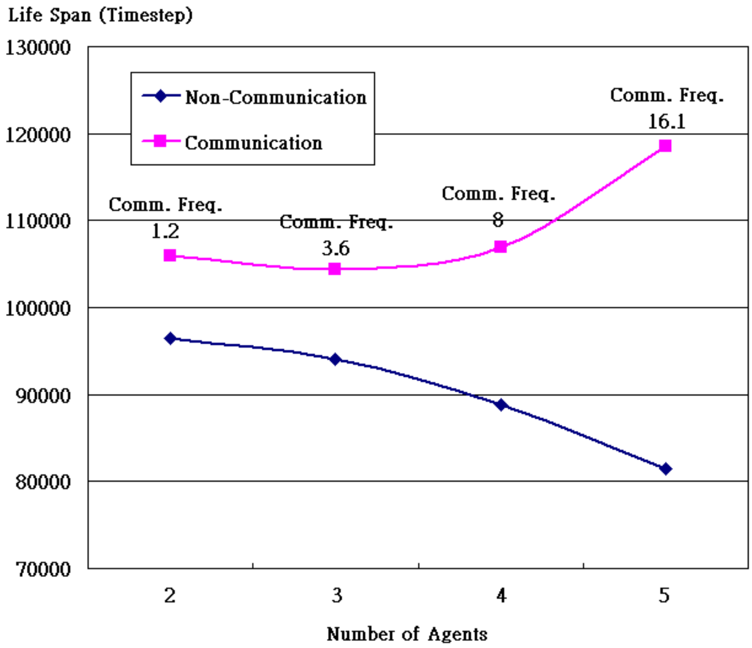

Figure 6.5: Experimental results of the overall average lifespan for all LTM agents in each experimental setting together with the value of average communication frequency on the top of the communicative agents' result.

6.2.2 Results

Figure 6.5 shows the overall average lifespan for all agents in each experimental setting. Each of the communicative agents' lifespan is shown with the average communication frequency above the result. Both Figure 6.6 and Figure 6.7 refer to communicative agents with five agents surviving in the environment. Figure 6.6 shows results of the agents' average lifespan in each experimental run against the total communication frequency including telling stories to and receiving stories from other agents. Figure 6.7 shows results of each agent's lifespan in all experimental runs against the story receiving frequency.

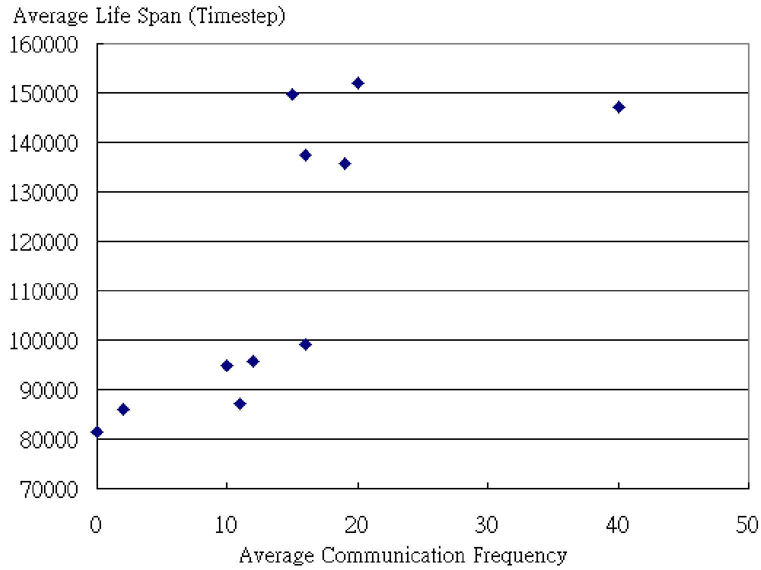

Figure 6.6: The relationship between average life span and average communication frequency. The correlation between variables was assessed by means of the Pearson Correlation test ($r = 0.678$, $p = 0.031$).

6.2.3 Evaluation

Generally Figure 6.5 indicates various interesting results. For non-communicative LTM agents, when there are more agents surviving in the environment, the average lifespan of these agents is more affected by the extra environmental dynamics created by the agents – agents are running around to disturb one's *Trace-back* process. However, LTM agents with narrative structure are able to cope with this situation by sharing meaningful stories among agents. An exponential-like trend of the increasing communication frequency can be observed when more communicative agents are surviving in the environment. Particularly when there are five agents in the environment, the average lifespan in this experimental setting reaches the highest score above all settings. This explains that narrative story-telling embedded in

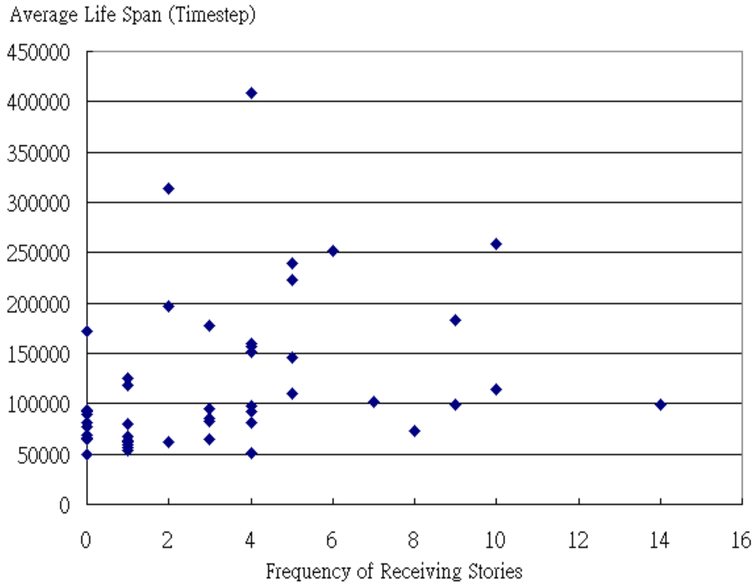

Figure 6.7: The relationship between each agent's life span and story receiving frequency. The correlation between variables was assessed by means of the Pearson Correlation test ($r = 0.304$, $p = 0.032$).

LTM control architecture as an additional communication feature helps agents to be more adaptive in coping with different environmental dynamics.

Focusing on the experimental setting of five communicative LTM agents surviving in the environment, Figure 6.6 shows statistically significant results that the average lifespan of communicative LTM agents is correlated to the average communication frequency in each experiment runs. The more frequently these agents communicate, the higher average lifespan they can have, as indicated by the value of Pearson Correlation (r) and the significance of the results (p). We also studied the relationship between event receiving frequency and the lifespan of each communicative LTM agent individually in all experiments, as the results show in Figure 6.7. The correlation is weaker compared to Figure 6.6, but it is positive and statistically significant. The reason is that each individual agent behaved very differently from each other because of its random movements in the dynamic environment, we still need more sophisticated measurements to investigate other aspects of this result.

6.2.4 Lessons Learnt

Experimental results on multiple agents surviving in the dynamic virtual environment with LTM memory control architectures and the embedded narrative storytelling and understanding features confirmed that having narratively structured communications helps agents to share significant information and re-organize this information from their own perspective, which results in canceling extra environmental dynamics generated by multiple agents surviving in the environment and having an enhanced lifespan when more agent encounters promote a higher frequency of communications.

From the exponential-like increase of communication frequencies in the results of experiments, we speculate that narratively structured communications can be far more beneficial to LTM agents if we can solve the problem of computational constraints and allow more agents to be placed into an experiment run.

The *System Observer Interface* introduced in the next chapter (Chapter 7) illustrates the phenomenon of communicative LTM agents completely reusing other agents' stories for finding the necessary resources or partially utilizing some significant situations from those stories – an emerging *Mixed Reconstructions* feature.

Chapter 7

User Interface for Visualizing Memory

In this chapter we describe the system interfaces that we have developed for observing agents' behaviours and particularly memory contents during the simulations. The main purpose of developing system interfaces is to give ideas to human observers about what agents are going to do next through showing the dynamically changing memory contents, as research has pointed out that the lack of explicitly showing intelligent agents' goals causes agents' behaviours to be perceived as incoherent (Sengers 2003). The second purpose of having an advanced system interface – the Observer Interface which we have developed in the recent work (Ho et al. submitted) and Chapters 5 and 6 – is to enable us to investigate the effectiveness of agent communications: how other's agents' experiences are re-organized in an agents' own autobiographic memory schemata.

Figure 7.1: System interface in the early study (Chapter 3 and 4) for a *Purely Reactive* agent.

7.1 Simple Interface

In the early study (Chapter 3 and 4) we have developed very basic and text-based system interfaces for agents with different control architectures: Purely Reactive, Trace-back and Locality; as illustrated in Figures 7.1 to 7.3.

The most basic interface for Purely Reactive agents, as shown in Figure 7.1, provides a basic experimental control – pausing the experiment, so that an observer can look at the simulation from a different perspective by changing the current viewpoint to others, such as looking down from above or taking the agent-centered viewpoint (seeing the virtual environment through the agent's 'eye'). Furthermore,

Figure 7.2: System interface in the early study (Chapter 3 and 4) for *Trace-back* agent. Detailed explanations for Trace-back memory contents can be found in Figure 3.4 and Sub-section 3.2.1 in Chapter 3.

observers are able to see the current status of all three internal states, the moving speed, current behaviour executed by the agent and the accumulated lifetime.

Based on the basic interface for Purely-Reactive agents, interfaces for Trace-back and Locality agents contain in addition the memory contents which are being dynamically updated on the left-hand side. Therefore, in addition to knowing all the status of an agent and what it is doing, observers can also infer what the agent is going to do next – tracking the agent 'undoing' each past action in the trace-back process through seeing the memory contents.

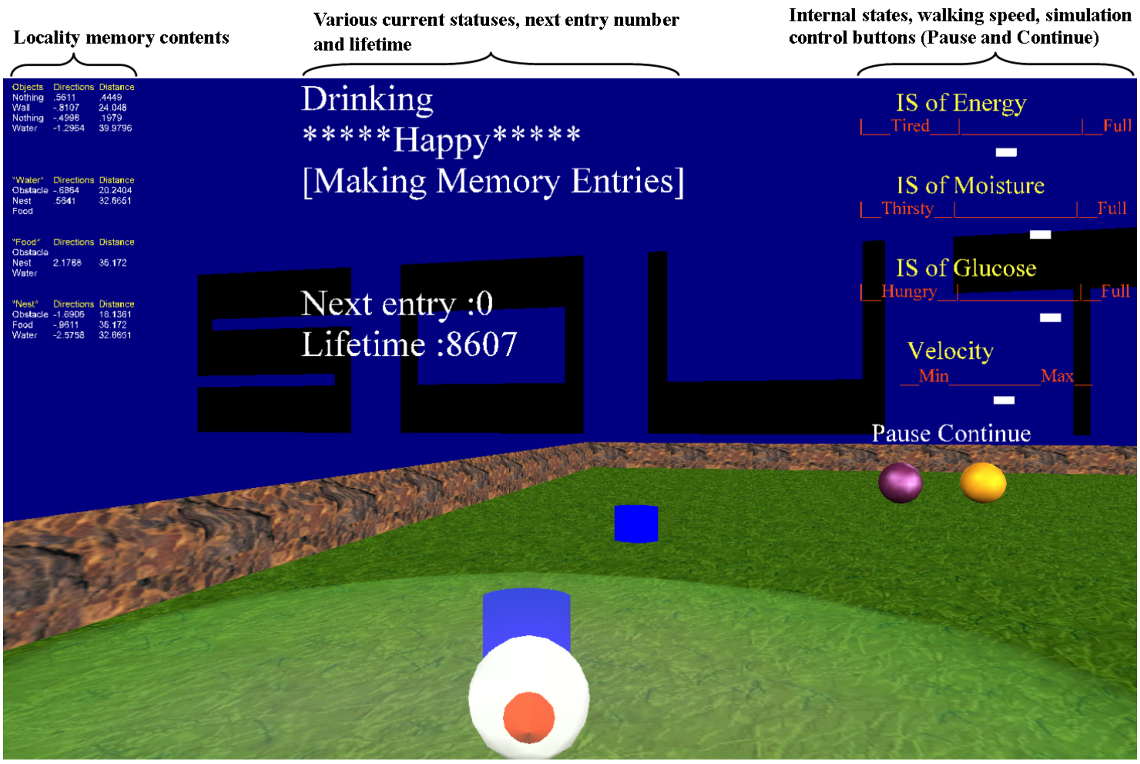

Figure 7.3: System interface in the early study (Chapter 3 and 4) for *Locality* agent. Detailed explanations for for Trace-back memory contents can be found in Figure 3.6 and Sub-section 3.2.1 in Chapter 3.

7.2 Observer Interface (OI)

In our recent research work aiming to develop a more sophisticated Long-term Autobiographic Memory architecture for autonomous agents, as described in Chapter 5 and 6, the Observer Interface (OI) we introduced for this work is essentially focusing on how contents of Short-term Memory (STM) and Long-term Autobiographic Memory (LTM) can be explicitly illustrated to human observers. Memory contents, particularly for LTM, are relatively complicated so a representation was chosen that is appealing in depicting agents' memories to observers.

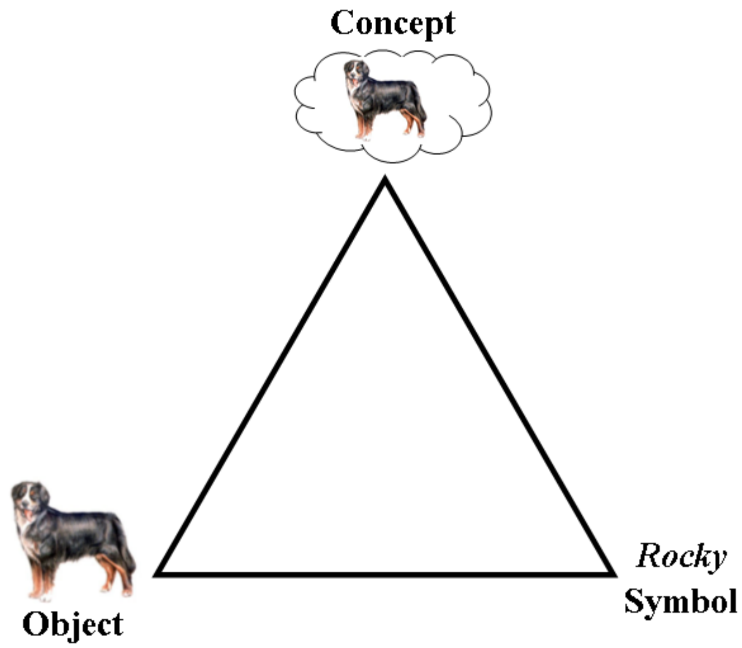

Figure 7.4: The meaning triangle representing the *Object*, *Concept* and *Symbol* of a dog ('Rocky' is the given name of the dog).

Considering the difficulties in representing the rich amount of memory contents from the schemata of autobiographic memory (see Sub-section 5.2.2 in Chapter 5), we adopted the concept of *meaning triangle* (Sowa 1999) to represent agents' memory contents in a higher level and easily understandable way – using icons. The term *meaning triangle* was popularized by Ogden and Richards, but Aristotle was the first to make the distinction, the history of meaning triangle can be found in Sowa (1999). Figure 7.4 shows an example of the meaning triangle. The meaning triangle shows how humans deal with the relationships between an real object, the concept of that object and the symbol of that object. Therefore, we introduced triples of icons to represent memory objects in the OI as it can be easier for human observers to understand 1) the current memory contents in an agent's LTM and 2) how these memory contents get re-organized dynamically. Although each triple of icons is not able to fully represent the details of a situation, it contains most representative information for an agent to find out a target situation in one step.

The system design of the Observer Interface can be found in Section B.3.7 in Appendix B.

Figure 7.5 shows the complete OI we developed for STM+LTM agents. Comparatively, the OI for PR agents has no memory contents to display, for STM agents it has only STM contents, and for LTM agents it has only reconstructed events from LTM. The OI offers the basic control for experiments (Start and Pause buttons) and indicates the current season of the environment and the agent's behaviour and lifetime. In addition to showing the contents of STM and LTM, it also updates agent's current internal variables and STM energy counters in a representation of lines – showing the trends of these variables over time.

Through the OI, we are also able to observe the details of how narrative storytelling helps agents' survival – the story properties reconstructed in ER process by communicative LTM agents. Reconstructed events from LTM of an individual agent are represented as groups of triple icons in the OI, as shown in Figure 7.5. Each triple contains *Landform*, *Object* and *Information Source* (identifies the agent). The interface represents the detailed situations of single events, as illustrated in Figures 7.6 to 7.9 which are extracted from the screenshots of the complete OI. As an example, a description of an illustrated event is given in Figure 7.6. We discovered that communicative LTM agents are not only able to completely reuse stories told by other agents (Figure 7.6 and 7.7), but they also reconstruct new stories by partly using situations extracted from other agents' stories (we call it *Mixed Reconstructions*, as shown in Figure 7.8 and 7.9). *Mixed Reconstructions* can

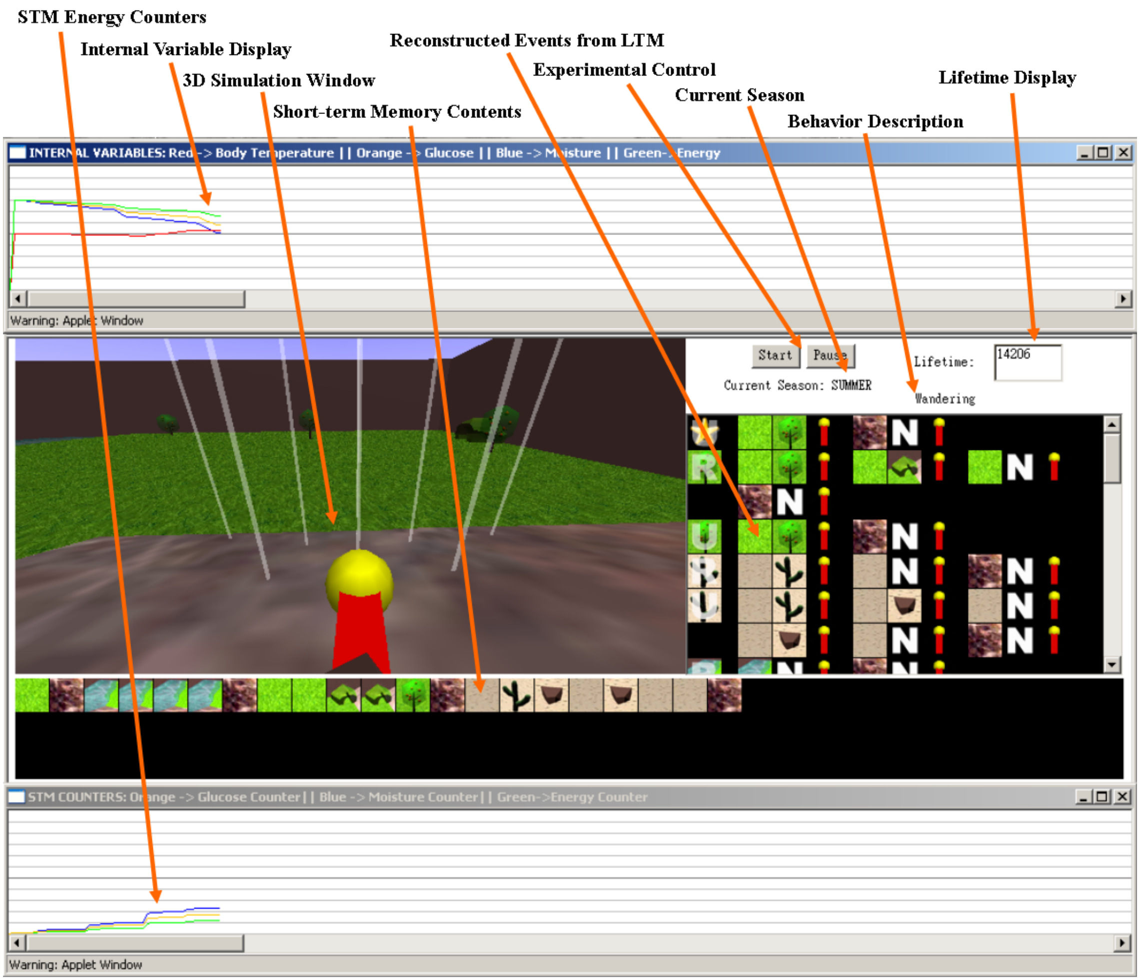

Figure 7.5: The Observer Interface for STM+LTM agents with indications of main components.

be seen as an *emerging effect* offered by the narrative story-telling structure and the characteristics of ER process in reconstructing events. This results in re-organizing significant situations to increase the efficiency of the reconstructed events for finding necessary resources in the dynamic environment. Furthermore, after agents survive for a certain period of time and communication frequency among them gets increased

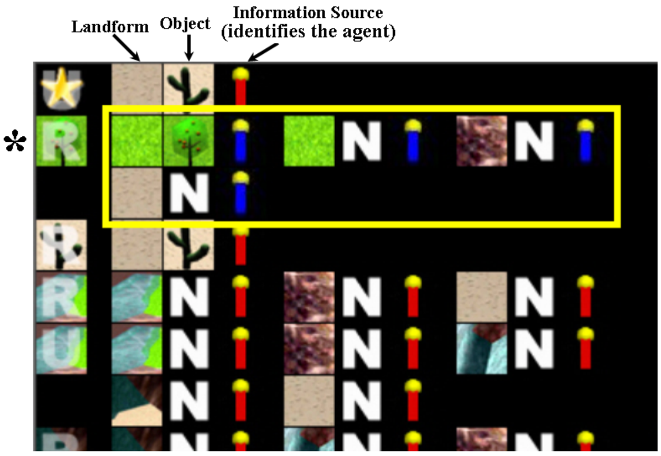

Figure 7.6: The highlighted area in the figure shows the agent has reconstructed an event completely from other agent's experience. This is a 'Redo' event with an apple tree as the target, as indicated by the letter 'R' on the left most apple tree icon (*). Four situations are shown in inverse order and each situation is represented by a triple of icons. Starting with the last triple (*DesertArea-NothingSensed-BlueAgentStory*), it shows the event was started when the blue agent sensed nothing (symbol 'N' in the middle icon) in the desert area. The blue agent then reached the mountain area, depicted by the second last triple (*MountainArea-NothingSensed-BlueAgentStory*). Finally the blue agent arrived in the oasis area (*OasisArea-NothingSensed-BlueAgentStory*) and sensed an apple tree (*OasisArea-AppleTreeSensed-BlueAgentStory*).

further, agents' experiences can be further shaped by exchanging the best stories offered from one agent to others.

7.3 Discussion

In this chapter we showed our progress in designing system interfaces for autobiographic agents. Regardless the interface is text-based or graphic-based, we believe that exhibiting contents of autobiographic agents' memory can be beneficial to hu-

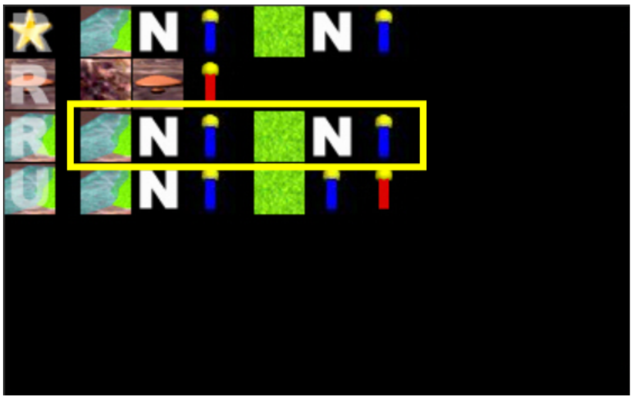

Figure 7.7: The highlighted area in the figure shows the agent has reconstructed an event completely from other agent's experience.

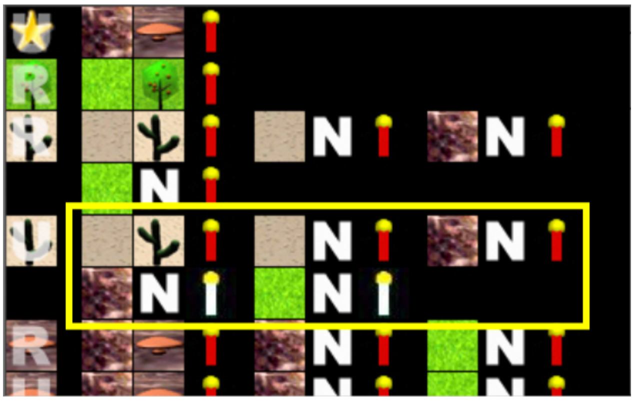

Figure 7.8: The highlighted area in the figure shows the agent has reconstructed an event by applying *Mixed Reconstructions* – this event contains situations from own and other's experiences. Experience from another agent can be seen from the *Information Source* with the white color agent body in the last two situation triples.

man observers in understanding agents' behaviours since the interface acts as a medium for them to comprehend the complete behavioural patterns generated from each agent control architecture. In addition to enabling human observers to infer agents' goals and making agents' behaviours to be perceived as coherent, these be-

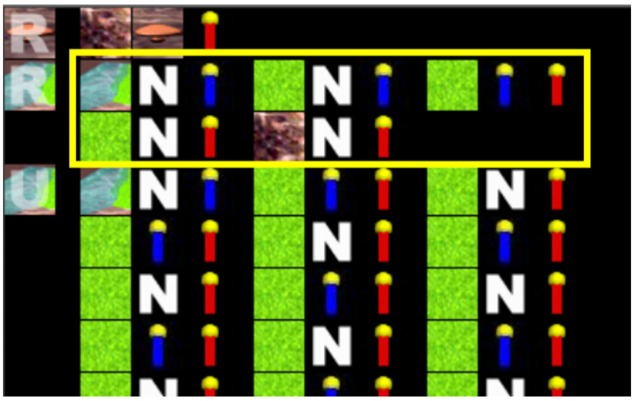

Figure 7.9: The highlighted area in the figure shows the agent has reconstructed an event by applying *Mixed Reconstructions* – this event contains situations from own and other's experiences.

havioural patterns can be seen as further explanations for experimental results we obtained in Chapter 6.

The current implementation of the Observer Interface (OI) is rather basic. We would like to improve it in future to be more interactive and graphically representative. With regard to user interactions, we would like to allow users to change the contents of agents' memory or to decide which event is more significant to the agent through the interface.

Chapter 8

Conclusion and Future Work

8.1 Overall Discussion

The earlier study in Chapter 3 highlights the effectiveness of autobiographic memory applied to an autonomous agent from an Artificial Life perspective. The virtual experimental-based approach deals with different implementation designs of control architectures for autobiographic agents, including detailed measurements of the agents' lifetimes compared with purely reactive agents, and studied in two distinct static environments. Experimental results produced evidence which supported the research hypothesis that autobiographic memory can prove beneficial, indicating increases in the lifetime of an autonomous autobiographic minimal agent. In particular, both Trace-back and Locality autobiographic memory architectures, with or without noise interference, showed superiority over purely reactive control (Ho et al. 2003). Trace-back is crucial for using autobiographic memory, see Section 5.2.2.

In Chapter 3 we have also investigated multiple autobiographic agents able to share an experienced sequence of events (perceptions and actions) with others who have the same goals of wandering and searching for resources so as to survive in the environment. The results of this study provided experimental evidence to reconfirm that within our framework autobiographic agents effectively extend their lifespan by embedding an Event-based Memory which keeps track of agents' previous action sequences as compared to a Purely Reactive subsumption control architecture. Multi-agent environmental interference dynamics resulted in decreasing average lifespan of agents. Some appropriate combinations of factors, e.g. communication motivation and cost factors, resulted in improved performance.

Through both earlier studies we learnt how behaviour patterns produced by a simple autobiographic memory (a sequence of actions) and its Trace-back process guide agents to survive better in static virtual environments; and we hypothesised that agents with simple autobiographic memory architectures will not be able to cope with the dynamics, such as *irreversible* and *non-commutative* events as well as unstable resources distributions, generated by a large and complex environment. This hypothesis led us, inspired by psychology research of general human memory and autobiographic memory, to investigate how *Event Specific Knowledge* which contains unstructured past experiences is reconstructed and organized by memory schemata in human memory during the remembering process. Therefore in Chapter 5 we developed Long-term Autobiographic Memory architectures (LTM) inspired by psychological models of human memory schema and autobiographic memory.

Having specified the *event significance* by measuring the positive or negative change of internal variables in the design of new LTM architectures, autobiographic agents in both single-agent and multi-agent experiments are able to manage and utilize past experiences by figuring out which one is most useful among all reconstructed events in a specific circumstance. Moreover, reconstructed events which contain sequences of experienced situations can guide an LTM agent to handle events with characteristics of 1) irreversibility: e.g. when an agent fails to swim against the water current of river and waterfall, it remembers situations in an event which covers areas of river or waterfall that are irreversible; and 2) non-commutativity: having experienced the penalty (-10% of *Energy*) by trying to touch the cactus without first picking up a stone and later successfully consuming the cactus with an stone, the agent remembers the correct sequence of actions for eating the cactus. Regarding non-commutative events, the sensitivity to ordering of events (e.g. pick up stone *before* attempting to open cactus vs. attempt to open cactus *before* picking up the stone) is rather different from the problems which are typical in spatial navigation where often resequencing of actions (e.g. achieving a sequence of displacements encoded by vectors) leads to the same result independent of order.

Observations on experiments in Section 6.1 of Chapter 6 indicate that events which were experienced by an LTM agent and brought a considerable number of negative changes to internal variables of this agent, would help the agent to be more adaptive in facing the environmental dynamics, such as coping with various types of object distributions. For example, an agent remembers that it was able to find apples from an apple tree in the oasis area during the summer season and it assumes apples also can be found in the same area all the time; therefore, later in the the winter, this agent tries to get apples by searching around in the oasis area, but it fails due to the dynamic resource distributions in the environment. In this case an environmental rule indicating 'oasis area has no apples in winter' is learnt by the agent through having negative changes of various internal variables when searching for apples which don't exist in the oasis area in winter time.

In Chapters 6 and 7 through experimental results on agents surviving in the new dynamic virtual environment with different memory control architectures and the embedded narrative story-telling and understanding features, we confirmed that:

1. The more sophisticated Long-term Autobiographic Memory control architecture effectively extends a PR agent's lifespan and increases the stability reflected in the changes of internal variables over time.

2. Having narratively structured communications helps agents to share significant information and re-organize this information from their own perspective, which results in canceling extra environmental dynamics generated by multiple agents surviving in the environment and having an enhanced lifespan when more agent encounters promote higher frequency of communications.

Furthermore, the combination of the improved STM and LTM control architectures produces the best average lifespan in single-agent experiments coping with dynamic environmental conditions in a large-scale virtual environment. Also, the *System Observer Interface* has been developed in Chapter 7 to illustrate the phenomena of communicative LTM agents completely reusing other agents' stories for finding the necessary resources or partially utilizing some significant situations from those stories – so that *Mixed Reconstructions* features emerge.

8.2 Main Contributions to Knowledge

Imagine a virtual environment inhabited by a group of autobiographic agents which 1) autonomously explore their dynamic environment in search of resources and 2) share their meaningful experiences as stories to help each other in adapting to their surroundings. Although each agent tries to remember situations and reconstruct events about the environmental conditions from an individually unique perspective, sharing stories between them ultimately produces better utilisation of environmental landmarks for finding the closest resource needed by a specific agent.

I have successfully applied the basic methodology outlined in the Chapter 1 to develop appropriate computational memory architectures for narrative autobiographic agents. Long-term Autobiographic Memory (LTM) architectures that were developed in this thesis have shown the ability of agents reconstructing their own autobiography through experiencing different events and telling as well as accepting stories during the interaction with other agents in the environment.

8.2.1 Primary Contributions

This thesis contributes both to the field of Artificial Life and Artificial Intelligence. It combines ideas from these two fields for the cognitive system modelling, simulation, behaviour architecture modeling and reasoning system. Our contributions have been published in the intelligent agent literature (Ho et al. 2003, Ho et al. 2004) and artificial life literature (Ho et al. 2005b), and submitted to a cognitive systems journal (Ho et al. submitted).

In Artificial Life, this work is the first to implement autobiographic agents with sophisticated memory architectures which are able to:

1. Reconstruct events by situations remembered in the memory and organise them into memory schemata.

2. Specify *event significance* by monitoring changes of internal variables for deciding which reconstructed event to be executed.

3. Allow basic bottom-up narrative structure to emerge via story-telling and story-understanding features offered by the memory architectures.

It is important that our goal is *neither* to attempt to develop a memory control architecture for agents to build mental maps of the environment for the purpose of navigation *nor* to use memory and reasoning processes to generalise dynamic environmental conditions for creating cost-effective schemes for agents' adaptations. We focus on building autobiographic agents which utilize *episodic information* to realise 'meaningful' (from the perspective of the agent) events and reconstruct stories from their own as well as other agents' experiences only during their lifetime, therefore off-line training data provided for agents to pre-learn certain tasks for surviving in the environment does not exist.

8.2.2 Auxiliary Contributions

Our study is also the first to emphasize non-commutative and irreversible sequences of events in studying agents' behavioural patterns in artificial life environments. *Irreversibility* highlights the difficulties for the basic *trace-back* mechanism that simply 'undoes' each action from the remembered sequence. Both *non-commutativity* and *irreversibility* require a more flexible memory architecture to remember or try out the correct sequence of a particular event.

The Observer Interface developed in this research helps human observers to easily infer goals of autobiographic agents through using iconic symbols to visualise meaningful memory contents dynamically – by depicting an agent's current internal needs, STM memory contents and LTM reconstructed events. Moreover, human observers can experience agents' behavioural changes from different perspectives in the three-dimensional virtual environment using the Observer Interface: a) looking down from above; or b) using first-person perspective by following an agent's movements and seeing through its 'eyes'.

8.3 Conclusion

In this dissertation I presented the results of interdisciplinary research related to the fields of Artificial Intelligence and Artificial Life. I proposed, implemented and experimentally verified a framework of developing computational memory control architectures for narrative autobiographic agents. In this approach, generic adaptive control architectures were built by placing autobiographic memory layers on top of autonomous reactive architectures. Virtual environments with various levels of dynamics and complexity were created for experiments to measure the performance of each architecture and to investigate the effect of communicating episodic memories between autobiographic agents.

In conclusion, the design considerations of narrative autobiographic memory architectures has identified the following key points:

- General knowledge of environmental conditions helps a memory agent to be adaptive in the environment; particularly significant events remembered by autobiographic agents are crucially useful for outperforming the reactive agents under circumstances created by the environmental dynamics.

- The significance level of an event stored in an agent's autobiographic memory is determined by the dynamic change of the agent's internal variables. The chosen event which will be re-executed by the autobiographic agent has the highest event significance if more than one event has been reconstructed from the memory (see sub-section 5.2.2).

- The autobiographic memory architecture which comprises Long-term and Short-term memory has superior performance for autonomous agents to survive in the dynamic and complex environment, since it has both 1) precision in Short-term memory *Trace-back* process and 2) flexibility offered by Long-term Autobiographic Memory – events reconstructed by fragmented situations from past experiences.

- By increasing the complexity of the dynamic environment, a rich variety of events can be generated for agents to experience and remember; therefore their varied life stories could be beneficial for the survival of the whole group through sharing significant events among them.

- The abstraction of high-level narrative structure for agent life stories is enabled to emerge via reconstructing events from own experiences as well as stories told by other agents.

- Memory contents of autobiographic agents are comprehensible by human observers through the Observer Interface. Hence agents' behavioural patterns are exhibited as goal related and narratively structured.

In contrast to research in developing virtual agents for interactive story-telling systems (Mateas 1999, Mateas and Stern 2002, Stern 2003, Cavazza et al. 2003), in which agents' new experiences are constructed essentially from the interaction history with the users; this research concerns the dynamic experience reconstruction for the agents in a bottom-up perspective. Consequently, the outcome of the simulated ‘remembering’ processes such as Event Reconstruction and Event Filtering and Ranking, is profitable to the agent; as the fundamental purpose for this research is to study the outcome of agents’ survival influenced by having significant events in autobiographic memory and sharing stories to other agents.

Regarding the algebra of time (Nehaniv and Dautenhahn 1998a), through presenting the implementation of autobiographic memory architectures and the experimental results, we showed that the extended *temporal horizon* of autobiographic agents, which was created by remembering significant events, can allow for planning for future actions and storytelling about past events in both static and dynamic environments.

8.3.1 Future Work

In future, this work can be extended in many ways including improvements to the agent control architectures and introducing emotional states.

Agent Control Architectures

A certain level of randomness could be added into Event Specific Knowledge (ESK) and the result of reconstructed events for Long-term Autobiographic Memory (LTM) agents in creating the effect of ‘Forgetting’. In psychology research forgetting is an important characteristic of human memory that helps humans to learn new tasks and adapt to new environments quickly (Markovitch and Scott 1988, Smith 1998). Thus agents’ ESK can be redesigned to remember just significant events which have been successfully re-executed, other events could be randomly deleted after a certain period of time passes by. However, the side effect of forgetting is that agents will sometimes face incomplete events for the process of *LTM Trace-back* because situations in different events are linked together in sequence. In this circumstance, agents can attempt to fix incomplete events by the enhanced design of 1) ER process – comparing all events internally in order to locate incomplete events and make them traceable; or 2) 'communication protocol' – finding out incomplete events through accepting and matching events from other agents.

Realizations in artificial agents of story-telling and narrative features can benefit from the increased temporal horizon of autobiographic agents using temporally extended meaningful information (Ho et al. 2004, Nehaniv 1999, Nehaniv et al. 2002). By receiving and re-using (and verifying) stories from other agents, an agent with Long-term Autobiographic Memory may be able to recognize other agents individually. If a narrative autobiographic agent is able to maintain interaction histories which keep track of the usefulness of stories told by other individuals, this agent could selectively choose an event from one of its 'favourite' agents to execute. This implies that a certain level of trust, as well as distrust, could be built up between agents as time passes by, cf. work on trust in multi-agent societies (Castelfranchi and Falcone 1998, Witkowski, Artikis and Pitt 2000).

Therefore, by extending this research, it might be possible to design and implement an autobiographic agent society where story-telling will serve as the basis of social networks emerging among agents in this society. As an agent's autobiographic experiences shared as story-telling with other agents might not be beneficial to all of them in some circumstances, the formation of one agent's social network will depend on the levels of trust on other individual agents and this trust is built up from successfully reusing and evaluating stories told by other individuals. The formation of cultural differences among autobiographic agents in the same society might also be observed when the environmental or living conditions vary from area to area inhabited by the society.

Remembering and Emotion

The functions of making memories, remembering and emotion locate in the same temporal lobes area of the human brain. The ways emotion can influence remembering episodic events has attracted a substantial amount of research for many decades in different research fields such as psychology and neuropsychology, see Dolan (2002) and McGaugh and Cahill (2002) for overviews. For example, McGaugh and Cahill (2002) suggested that emotional events are specially encoded so that they are difficult to forget. We expect that linking basic emotional states to the existing autobiographic memory architectures in a bottom-up fashion would be beneficial in further modulating the behaviours of an autonomous agent; since when a human is experiencing a significant event, in addition to the change of the episodic memory his/her emotional states are also affected.

Potential Applications

Computer Games

> *"The next big challenge for game AI may be getting a game's cast of characters better at learning and social interaction."* (Cass 2002, page 44)

In recent years, some concepts from nouvelle AI have been experimentally applied to the design of various types of new computer games, as introduced in Subsection 2.3 in Chapter 2. These attempts successfully enhanced the abilities of those

Non-Player Characters (NPCs), so that these games have become more challenging to the player and the entertaining features of these games have also been enriched.

In various types of computer game, such as role-playing and simulation, the design of bottom-up and emergent narrative is very influential on players' gaming experience. Researchers in the fields of Narrative Intelligence and Game AI have suggested that emergent narrative in computer games with open-ended story and AI can be highly engaging since its features give many more chances to players to interact with NPCs (Cass 2002, Aylett 2005). We expect that using autobiographic memory for a NPC to remember significant events which occurred during the interaction with a player will enhance their learning abilities. Thus the NPC can have better social interaction with other NPCs and the player.

Interactive Story-telling Systems To support the story-telling environment, synthetic characters with autobiographic memory will be able to construct their own autobiography through experiencing different events and telling as well as accepting (and re-interpreting) stories during the interaction with other agents and users in the environment; thus the knowledge representation for autobiographic memory will be able to take advantage of memories of significant and 'meaningful' events from the agent's perspective. Moreover, the changes of emotional states of a synthetic character can be used with other internal variables for measuring the significance of each event that happens to the character.

In interactive story-telling systems, a particular knowledge representation for autobiographic memory in representing different types of knowledge from both the environment and the user input could be investigated and developed. Since autobiographic agents have individually significant events to remember, they will be able to provide enough richness in story-telling environment for keeping the user's interest. Having the ability to re-interpret stories by autobiographic agents, particularly in the environment creating open-ended stories for emergent narrative, helps them in making the user's experience more 'life-like', believable, and ultimately enjoyable which might greatly facilitate learning. A story-telling environment populated by autobiographic agents will be considered to enhance the users' satisfaction, enjoyability and engagement when users are interacting with the system.

In recent years, AI story-telling applications which aim to entertain the user, e.g. Interactive Drama Facade (Mateas and Stern 2002), or have an educational purpose, e.g. the VICTEC project (VICTEC 2005), make good use of emotional synthetic characters to create emergent narrative. Both types of application can be one of the future directions for this research. By accompanying the state-of-the-art features, e.g. emotional states and supporting emotive facial expressions, autobiographic memory could also be a crucial component for developing synthetic characters which can learn from their own significant experiences.

8.3.2 Concluding Remark

As modeling characteristics of autobiographic memory is a very difficult task in the research fields of both Artificial Intelligence and Artificial Life, in this PhD research we have been facing many challenges to develop memory architectures for bottom-up narrative autobiographic agents. Although the novel architectures we developed in this research have been evaluated only in Artificial Life environments, we expect that they would be able to deliver new features in the design of virtual synthetic agents situated in narrative story-telling systems and computer games.

"The study of autobiographical memory, then, represents a challenge to the cognitive psychologist and the challenge is how to understand personal meanings." (Conway 1990, page 186)

Appendix A

Process Description Language (PDL)

Process Description Language (PDL) was originally developed to produce cooperative dynamics architectures, in which many active processes operate in parallel and represent behaviours taking information from sensors to generate a control action if needed (Steels 1992). To construct our autobiographic memory architectures in our research, we have made a limited use of PDL by implementing a typical PDL process which gives a certain influence to the motor speed variable of agents' actuator.

In each time step, rather than assigning a specific value to the motor speed, a fixed proportion of the motor speed is added or subtracted to the previous output value until the maximum or minimum speed has been reached. In this way agents' movements are perceived as more smooth and 'life-like' by human observers. Another advantage of using PDL is the simple implementation. An example is given in Figure A.1 illustrating a conceptual graph and the pseudo-code of the PDL implementation in this research.

```
//PDL process to control the speed of agent
function speedControl() {
    // when an obstacle is sensed and the speed has not reached minimum
    if (distance <= sensoryRange and velocity >= minSpeed) {
        // decrease a fixed tiny amount of the speed value in each time step
        velocity = velocity - 0.0005;

    // when no obstacle is sensed and the speed has not reached maximum
    }else if(distance > sensoryRange and velocity <= maxSpeed) {
        // increase a fixed tiny amount of the speed value in each time step
        velocity = velocity + 0.0005;
    }
}
```

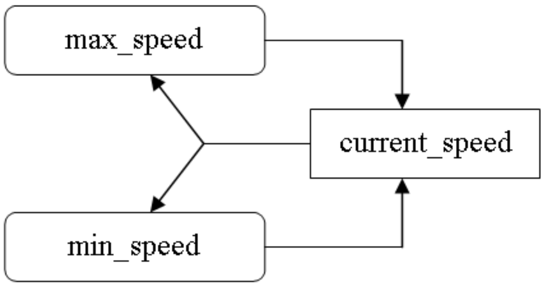

Figure A.1: The implementation of PDL process in controlling the motor speed of the agent.

Appendix B

System Design for Architectures and Environments

In this appendix, we illustrate system design diagrams for different virtual environments and agent control architectures. In order to show the main characteristics of each architecture, descriptions are provided and some important features are highlighted in each diagram.

To help understanding the symbols used in each data flow diagram which represents the functional structure of an agent control architecture, Figure B.1 shows standard data flow diagram symbols with descriptions.

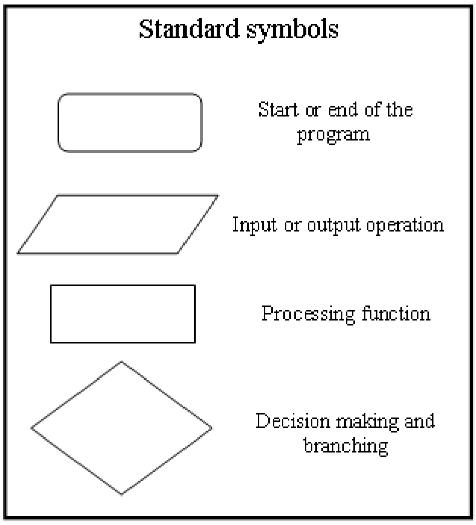

Figure B.1: Standard symbols used for creating data flow diagram in this appendix.

B.1 Single-Agent Experiments in Early Studies

B.1.1 Static Virtual Environment

The design of the static virtual environment for single autobiographic agent experiments is shown in Figure B.2. This environment has three different types of resources, four obstacles and boundaries. Screenshots of this environment and agent's movements can be found in Sub-section 3.2.1 in Chapter 3.

Figure B.3 illustrates the interactions between the Script program which controls an agent's behaviours and the virtual environment. In order to update an agent movement in each time step, the Script program has to gather information about the agent's current position and the locations of objects in the environment.

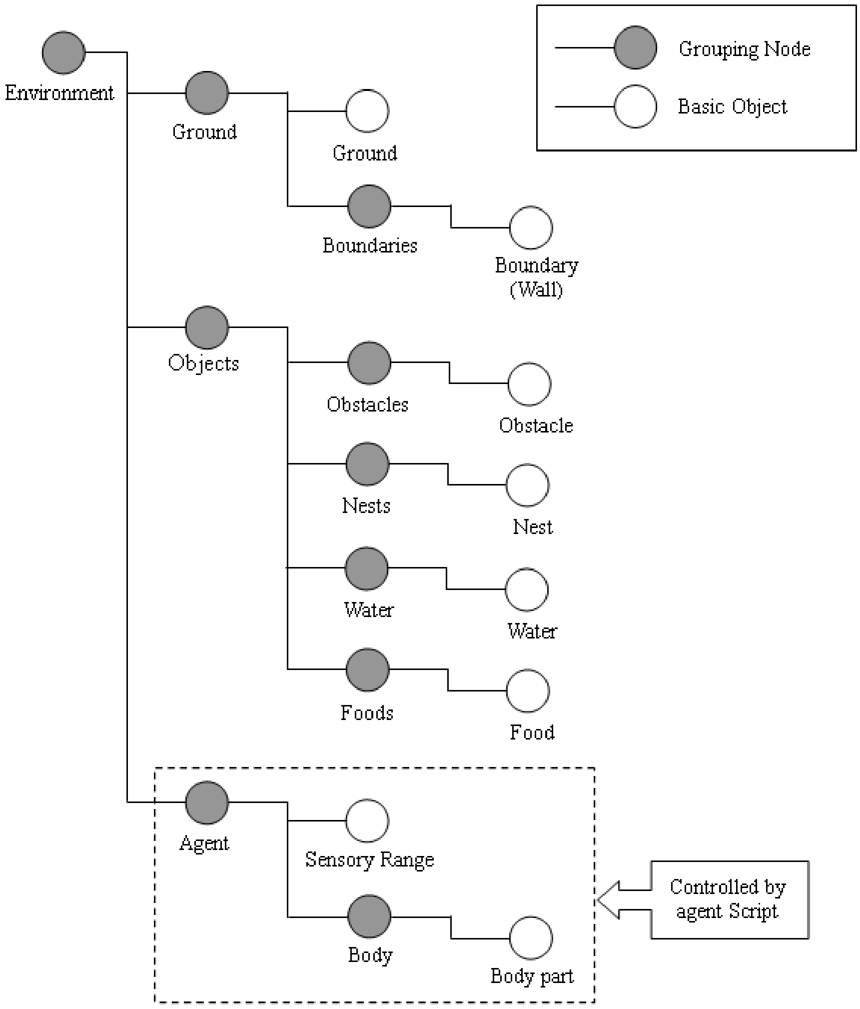

Figure B.2: The layout of the static virtual environment for single autobiographic agent experiments in our early study (see Chapter 4). Nodes relating to environmental lighting and user viewpoints are omitted in the figure.

B.1.2 Purely Reactive Agent Architecture

To model the algorithm implemented for the *Purely Reactive* (PR) agent, Figure B.4 illustrates the data flow diagram which represents the structured design of the PR architecture. A PR agent reacts to the objects in the environment and at the same

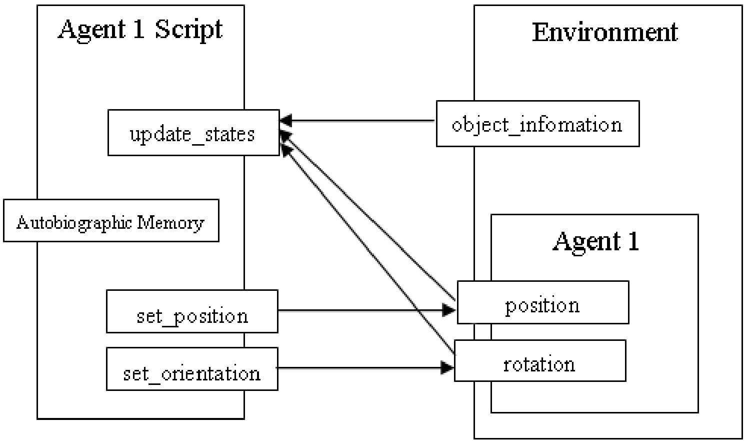

Figure B.3: Interactions between the Script program which controls an agent's behaviours and the virtual environment.

time, it has three different internal variables to maintain.

B.1.3 Trace-back Autobiographic Memory Architecture

To model the algorithm implemented for the *Trace-back* autobiographic agent, Figure B.5 illustrates the data flow diagram which represents the structured design of the Trace-back autobiographic memory architecture. Based on the design of the PR architecture, a Trace-back autobiographic agent has the same embodiment as a PR agent and also three different internal variables to maintain. For the data structure of Trace-back memory, see Figure 3.4 in Sub-section 3.2.1 in Chapter 3.

B.1.4 Locality Autobiographic Memory Architecture

To model the algorithm implemented for the *Locality* autobiographic agent, Figure B.5 illustrates the data flow diagram which represents the structured design of the Locality autobiographic memory architecture. Based on the design of the PR

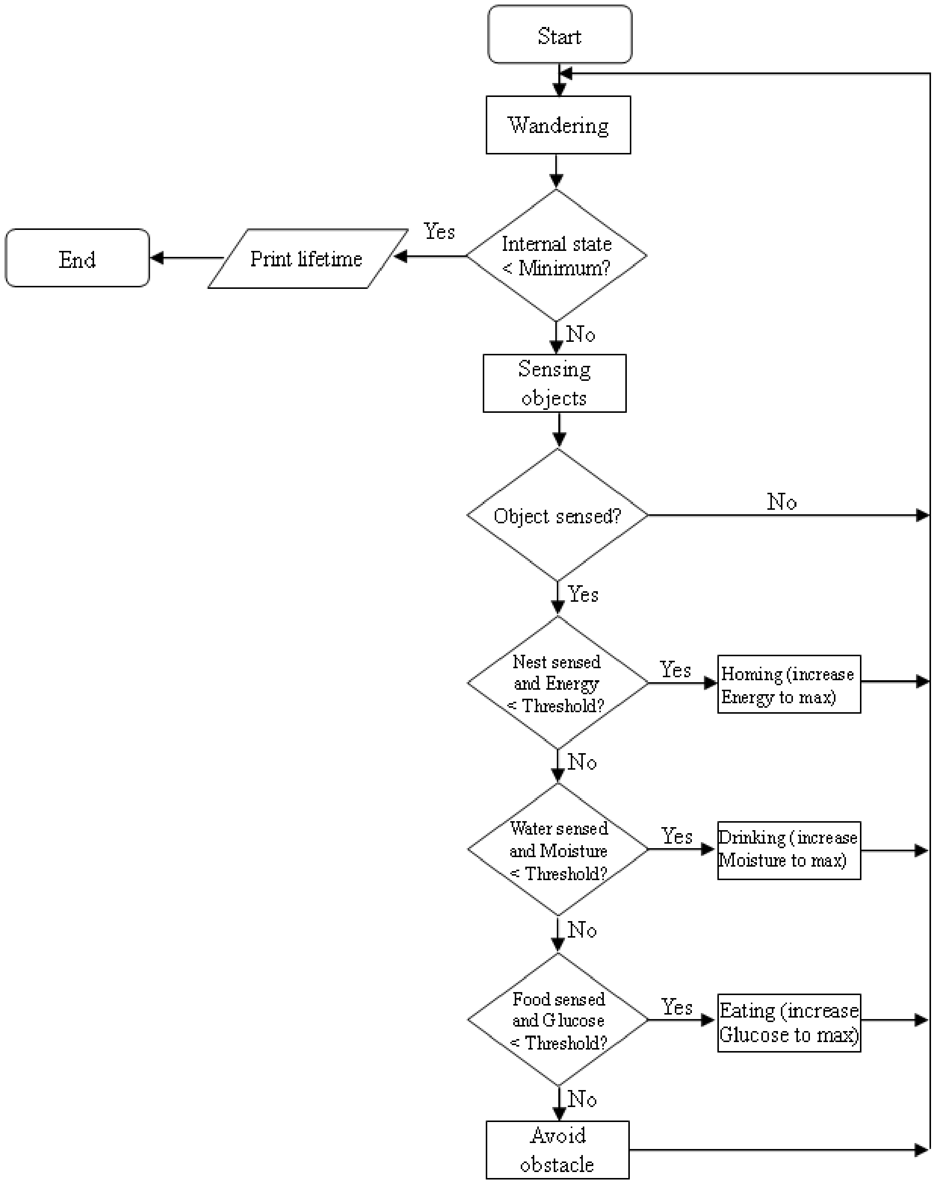

Figure B.4: Data flow diagram of Purely Reactive agent architecture for single-agent experiments in our early studies.

architecture, a Locality autobiographic agent has the same embodiment as a PR agent and also three different internal variables to maintain. For the data structure

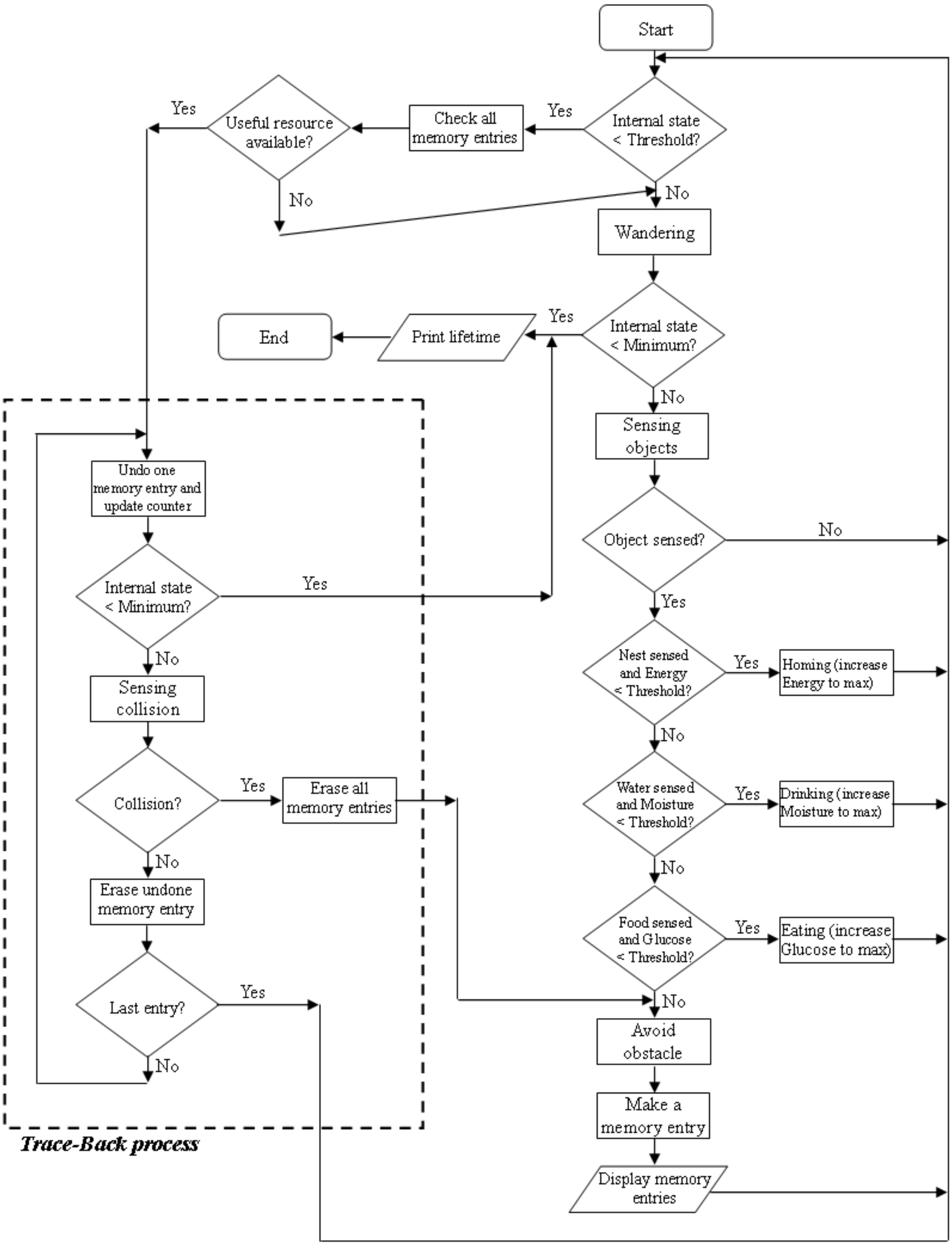

Figure B.5: Data flow diagram of Trace-back autobiographic memory architecture for single-agent experiments in our early studies.

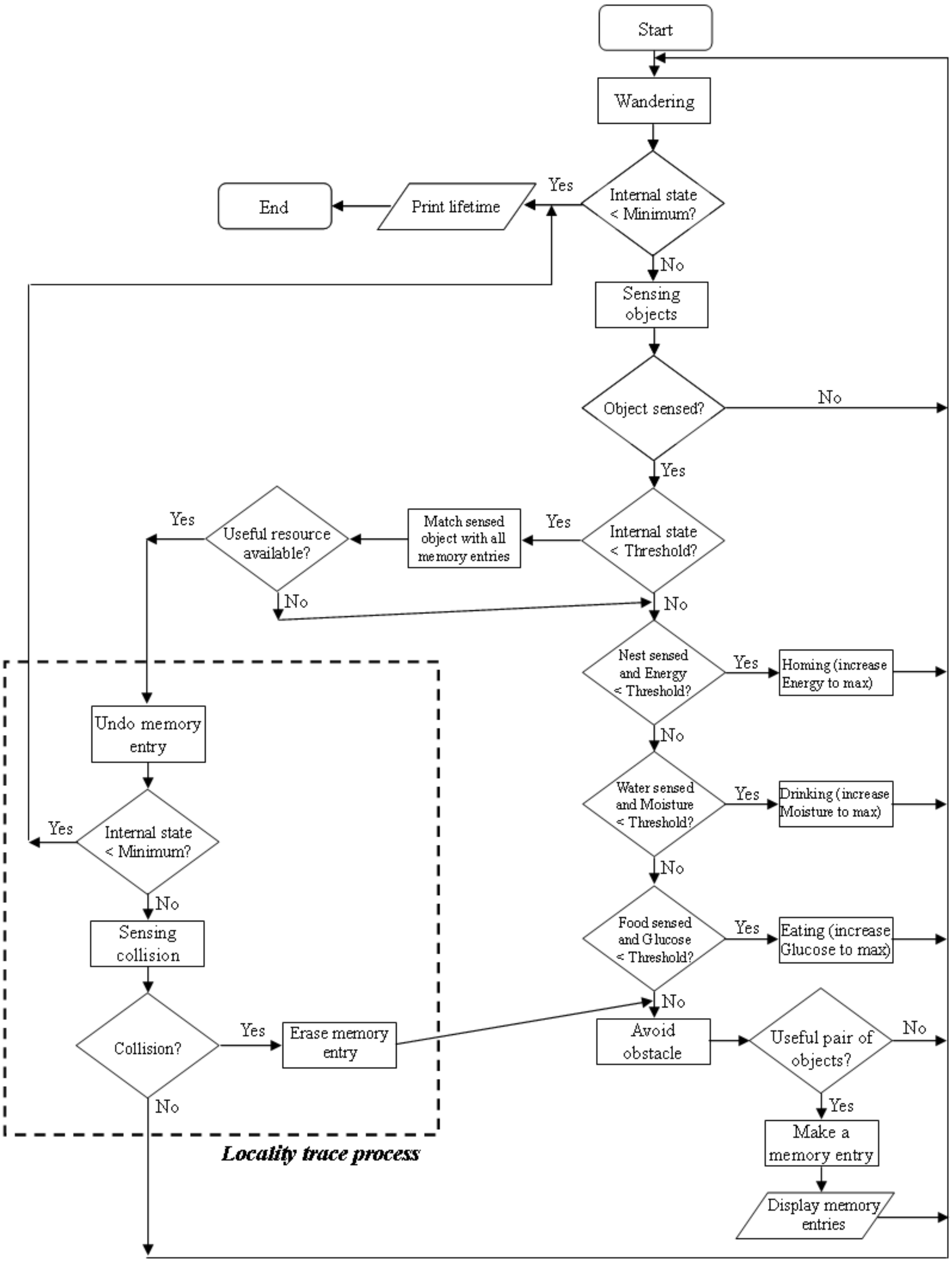

Figure B.6: Data flow diagram of Locality autobiographic memory architecture for single-agent experiments in our early studies.

of Trace-back memory, see Figure 3.6 in Sub-section 3.2.1 in Chapter 3.

Programming source codes with comments and descriptions of different files which describe the implementations of the environment and architectures can be found in Section C.1 in Appendix C.

B.2 Multi-Agent Experiments in Early Studies

B.2.1 Static Virtual Environment

The design of the static virtual environment for multiple autobiographic agent experiments is shown in Figure B.7. Compared with the environment used in the previous work which studied single autobiographic agent, this environment has only one type of resource. All agents have the same design for the body and sensor; therefore they are copies of the same agent object. Screenshots of this environment and agent can be found in Sub-section 3.2.2 in Chapter 3.

Figure B.8 illustrates the interactions between the Script program which controls an agent's behaviours and the virtual environment. An agent's movements and communications are controlled by Script programs: in each time step, an agent's position will be updated with the calculations of distances and angels facing other objects or agents in the environment, communication (copying another agent's memory) may occur when the internal variable of the agent is lower than the threshold and another agent is sensed.

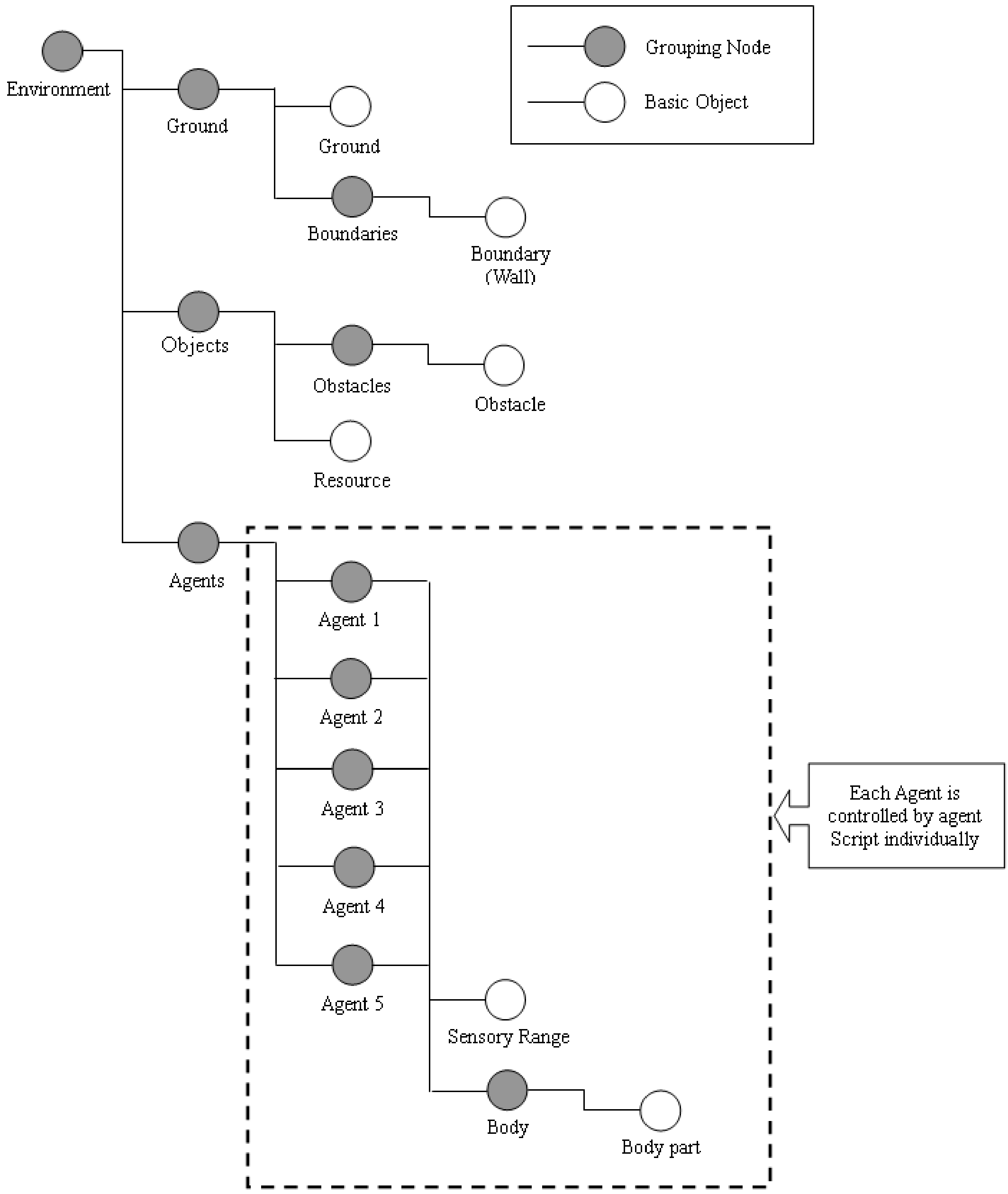

Figure B.7: The layout of the static virtual environment for multiple autobiographic agent experiments in our early studies. Nodes relating to environmental lighting and user viewpoints are omitted in the figure.

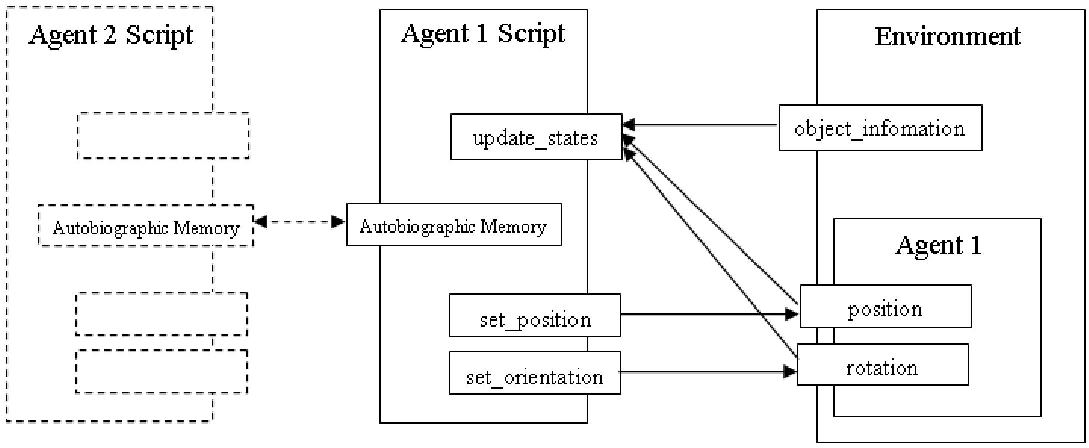

Figure B.8: Interactions between the Script program which controls an agent's behaviours and the virtual environment.

B.2.2 Purely Reactive Agent Architecture

To model the algorithm implemented for the *Purely Reactive* (PR) agent, Figure B.9 illustrates the data flow diagram which represents the structured design of the PR architecture. The PR agent reacts to the objects in the environment and at the same time, it has an internal variable to maintain for its survival.

B.2.3 Trace-back Autobiographic Memory Architecture

To model the algorithm implemented for the second *Trace-back* autobiographic agent for studying multiple autobiographic agents sharing their memories and surviving in a static environment, Figure B.10 illustrates the data flow diagram which represents the structured design of the Trace-back autobiographic memory architecture. Based on the design of the PR architecture, a Trace-back autobiographic agent has the same embodiment as a PR agent and also one internal variable to maintain. Furthermore, there is an *Energy Counter* to prevent the agent from 'dying' during the Trace-back

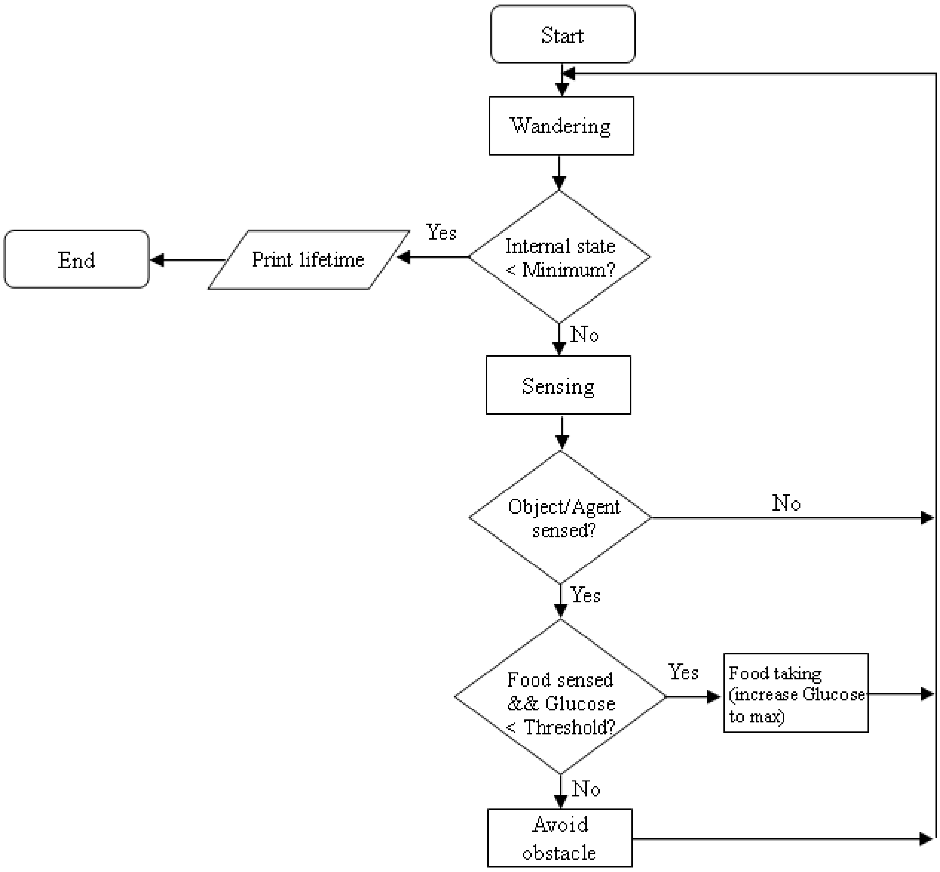

Figure B.9: Data flow diagram of Purely Reactive agent architecture for multi-agent experiments in our early studies.

process. For the data structure of Trace-back memory, see Figure 3.13 in Subsection 3.2.2 in Chapter 3.

Programming source codes with comments and descriptions of different files which describe the implementations of the environment and architectures can be found in Section C.2 in Appendix C.

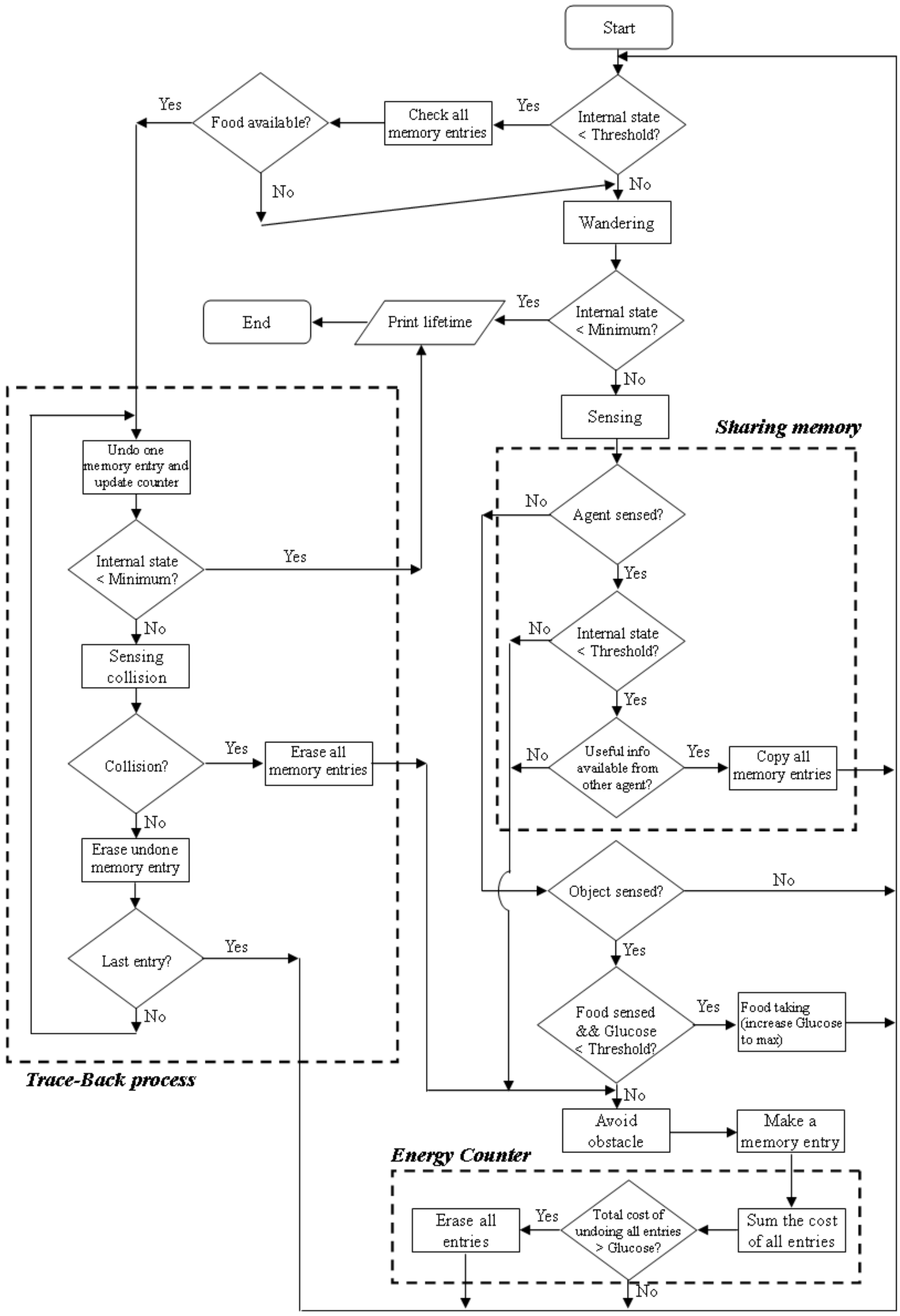

Figure B.10: Data flow diagram of Trace-back autobiographic memory architecture for multi-agent experiments in our early studies.

B.3 Studies of Long-term Autobiographic Memory Architectures

B.3.1 Large and Dynamic Virtual Environment

The design of the large and dynamic virtual environment for our studies of single and multiple autobiographic agent experiments is shown in Figure B.11. In addition to having various types of object and landform, this environment is relatively large in comparison to the environments used in the early studies. Moreover, environmental heat provided by different types of landforms is constantly changed in each season. River, Waterfall and Lake are special landforms, since they provide also moisture resource (water) to the agent. However, in the winter season water in these three landforms will be frozen and thus the moisture resource will not be available.

There are two types of objects: static and dynamic. The distribution of dynamic objects is different in each season. Stone is a specific kind of dynamic object since it can be collected by the agent and go with the agent until the agent drops it down. A drawing model and screenshots of this environment and the agent design can be found in Sub-section 5.2.1 in Chapter 5.

Figure B.12 illustrates the interactions between the Script program which controls an agent's behaviours and the dynamic virtual environment. Similar to the design in early studies of basic autobiographic architecture, an agent's movements and communications are controlled by Script programs. However, there is an extra *Environment Script* for controlling the environmental dynamics – objects' distribution and environmental effects, such as fog, lighting and dynamic landforms.

B.3.2 Purely Reactive Agent Architecture

Similar to the design in our early studies, Figure B.13 illustrates the data flow diagram which represents the structured design of the Purely Reactive (PR) architecture. Since the agent has one more internal state Body Temperature to maintain, it has to reach a proper area to adjust Body Temperature when its value exceeds the upper or lower threshold.

B.3.3 Short-term Memory Architecture

Based on the design of Purely Reactive agent, Short-term Memory (STM) architecture has one improved feature compared with *Trace-back* architectures from the early studies – *Energy Counter* which dynamically shrinks the size of the memory according to the energy counter values. Figure B.14 illustrates the data flow diagram which represents the structured design of the STM architecture. Environmental rules can be learnt if an STM agent has experienced an irreversible event. These rules can help the agent to validate memory entries which don't violate any rules learnt. For the data structure of Short-term Memory, see Figure 5.6 in Sub-section 5.2.2 in Chapter 5.

B.3.4 Long-term Autobiographic Memory Architecture

Figure B.15 illustrates the data flow diagram which represents the structured design of the Long-term Autobiographic Memory (LTM) architecture. Instead of displaying all memory entries in each time step, a LTM agent remembers new situations in Event Specific Knowledge (ESK) and only displays events reconstructed from ESK by using the Observer Interface when a Trace-back process is triggered. Environmental rules can be learnt if an LTM agent has experienced an irreversible event. These rules can help the agent to filter events violating any rule learnt. For the data structure of LTM in *Event Specific Knowledge* (ESK), see Figure 5.7 in Sub-section 5.2.2 in Chapter 5.

B.3.5 Short-term and Long-term Memory Architecture

Figure B.16 illustrates the data flow diagram which represents the structured design of the Short-term and Long-term Autobiographic Memory (STM+LTM) architecture. STM+LTM architecture has both STM and LTM; therefore an agent can retrieve useful experience from both memories. The current setting is that STM has higher priority to be searched and reused if there is information regarding a useful resource. When the agent is not able to find anything useful in STM, LTM memory will be checked. Due to the space constraints in the diagram, details of STM and LTM are left out from the diagram.

B.3.6 Long-term Communicative Memory Architecture

Figure B.17 illustrates the data flow diagram which represents the structured design of the Long-term Communicative Autobiographic Memory architecture. This architecture is based on the LTM architecture and with the extra process to pass the event created from an agent's own LTM to the receiving agent.

B.3.7 Observer Interface

Figure B.18 shows the system diagram for the overall interactions between observer, virtual environment implemented by VRML scene and Java programs displaying agents' status, memory contents as well as providing the simulation control. Since VRML is a object description language for creating three-dimensional objects and scenes, it has limited functions and sensors for user interaction. To achieve displaying concise memory contents by using icons and agents' internal variables in a line chart which is continuously updated in each time step, at the same time providing simple control for the observer to pause the simulation in order to carry out detailed investigations in a particular moment, External Authoring Interface (EAI) has been used to bridge the communications between VRML scene and Java programs.

In each time step, the Java programs can receive events from the VRML scene through EAI, such as values of agents' current internal states, lifespan and various types of memory contents. Then each Java program class will display these events through different Applets in the Observer Interface (OI) on a HTML page. Figure B.19 shows the abstract class diagram with relationships between Java classes. All classes can be divided into four groups:

- *agentState*, *GraphPanel*, *DataSource* and *DataPoint* classes are to show the agents' current status which includes lifespan and internal variables.

- *Pause* class is to pause the simulation and display the current season of the environment.

- *STMContent* class is to show agents' Short-term Memory contents through icons.

- *LtmER* and *canvasNew* classes are to illustrate the results of Long-term Autobiographic Memory Event Reconstruction process – all possible reconstructed events to be chosen to re-execute.

Programming source codes with comments and descriptions of different files which describe the implementations of the environment, architectures and the OI can be found in Section C.3 in Appendix C.

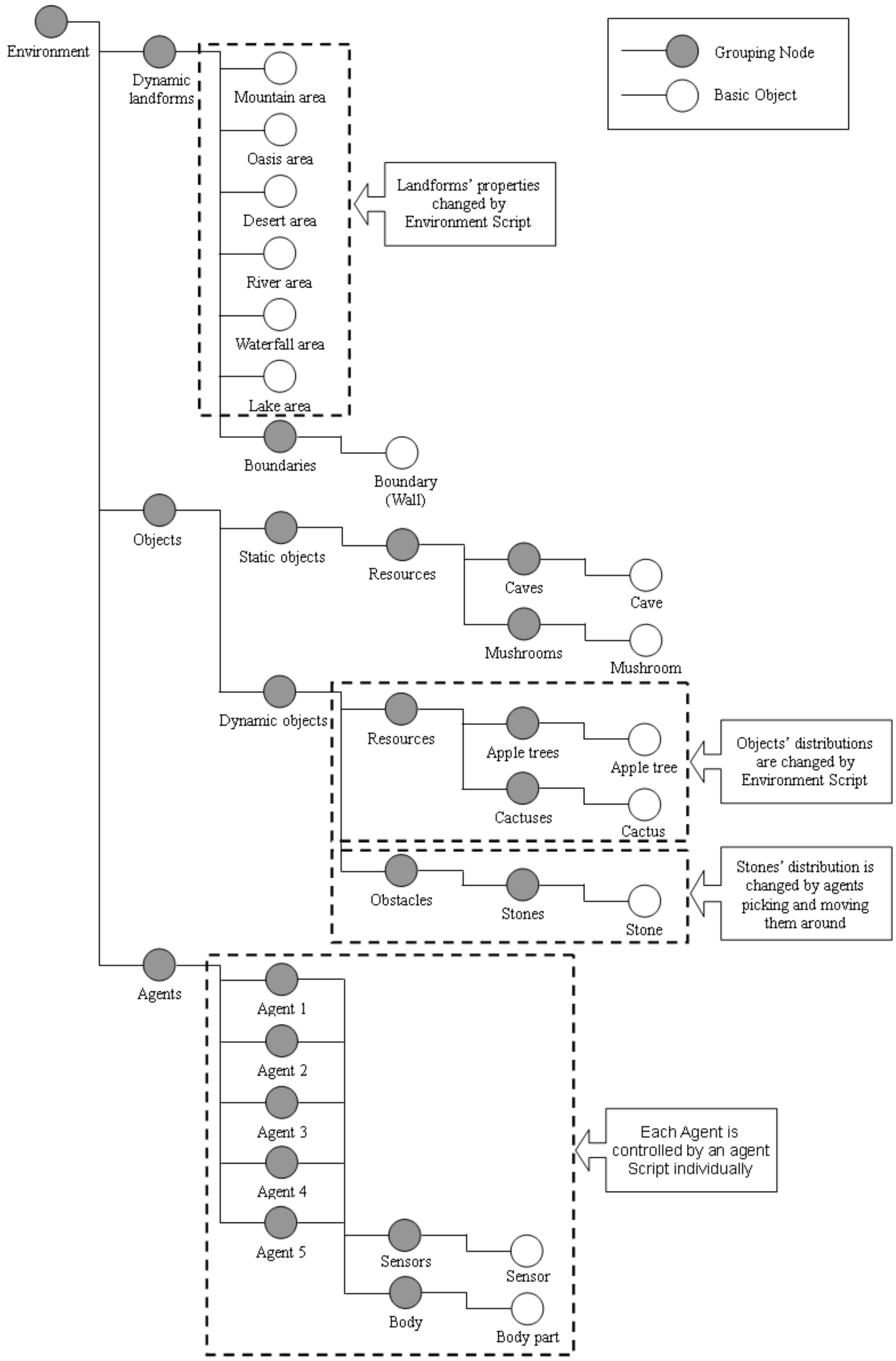

Figure B.11: System design of the large and dynamic virtual environment for single and multiple autobiographic agent experiments. Nodes relating to environmental lighting and user viewpoints are omitted in the figure.

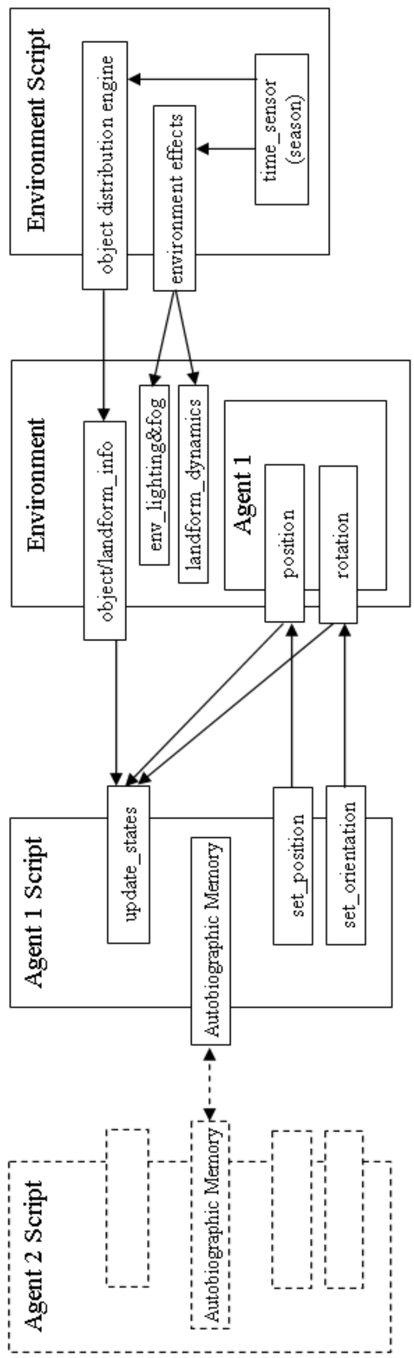

Figure B.12: Interactions between the Script program which controls an agent's behaviours and the dynamic virtual environment.

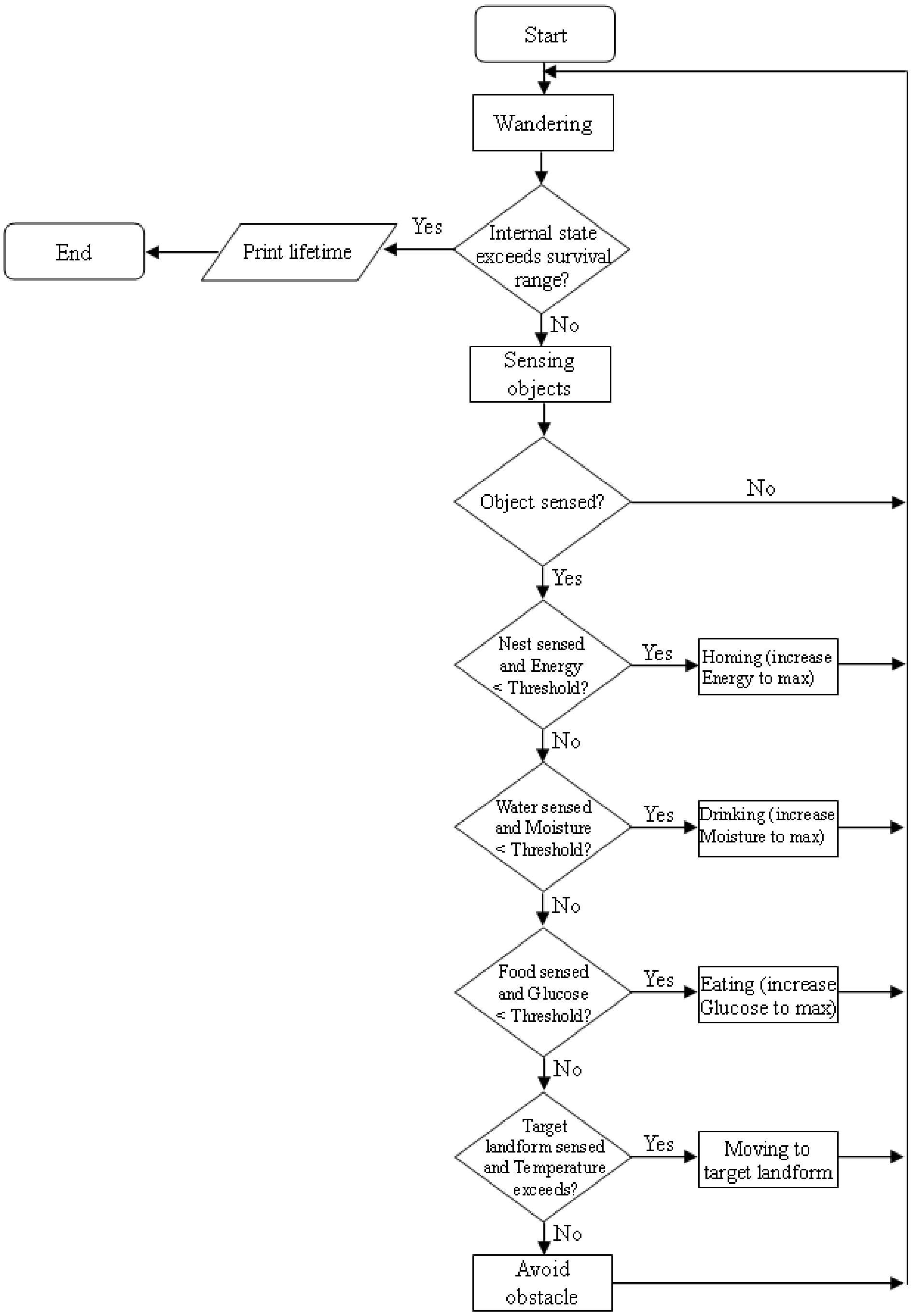

Figure B.13: Data flow diagram of Purely Reactive agent architecture.

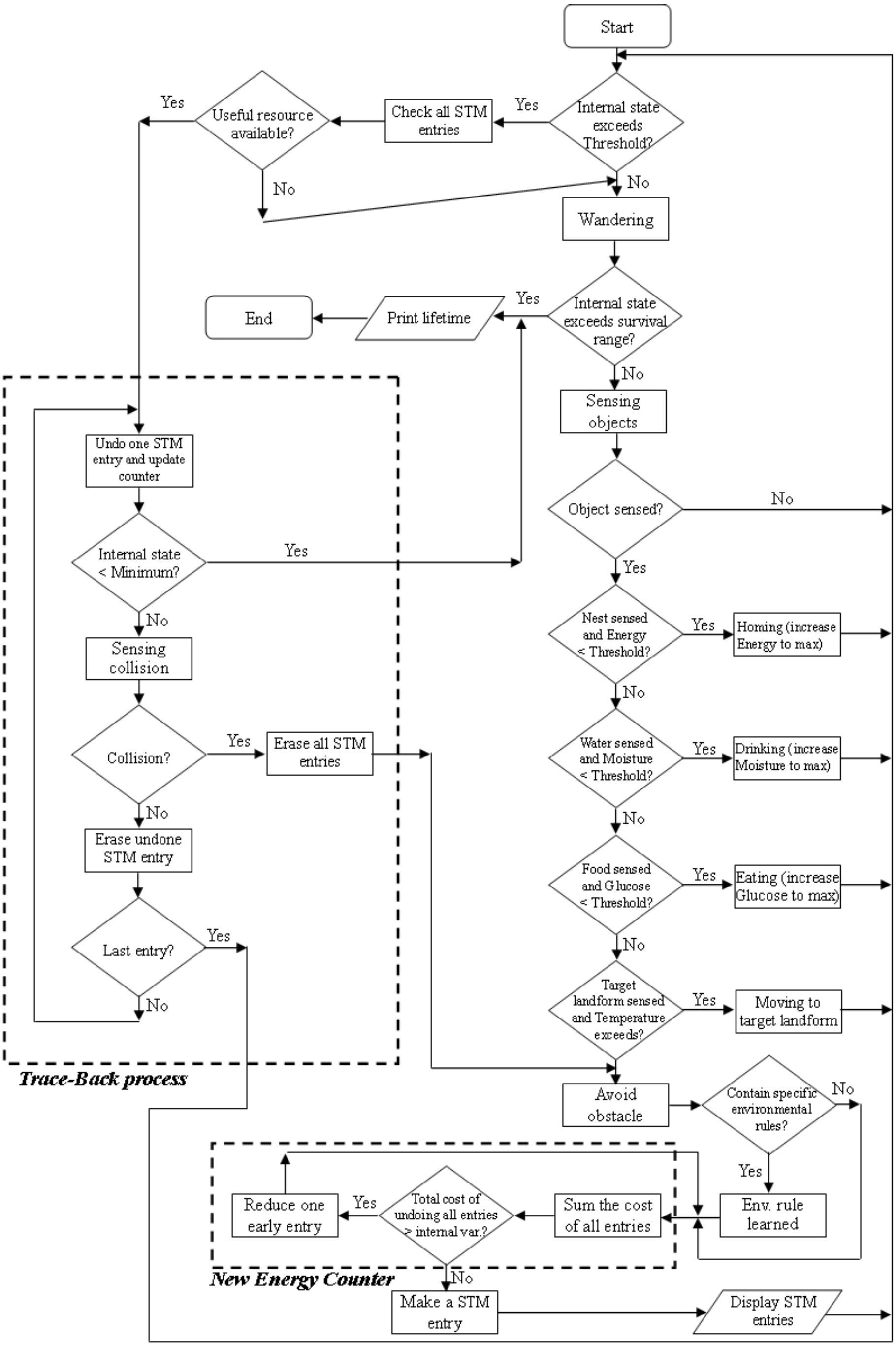

Figure B.14: Data flow diagram of Short-term Memory (STM) architecture.

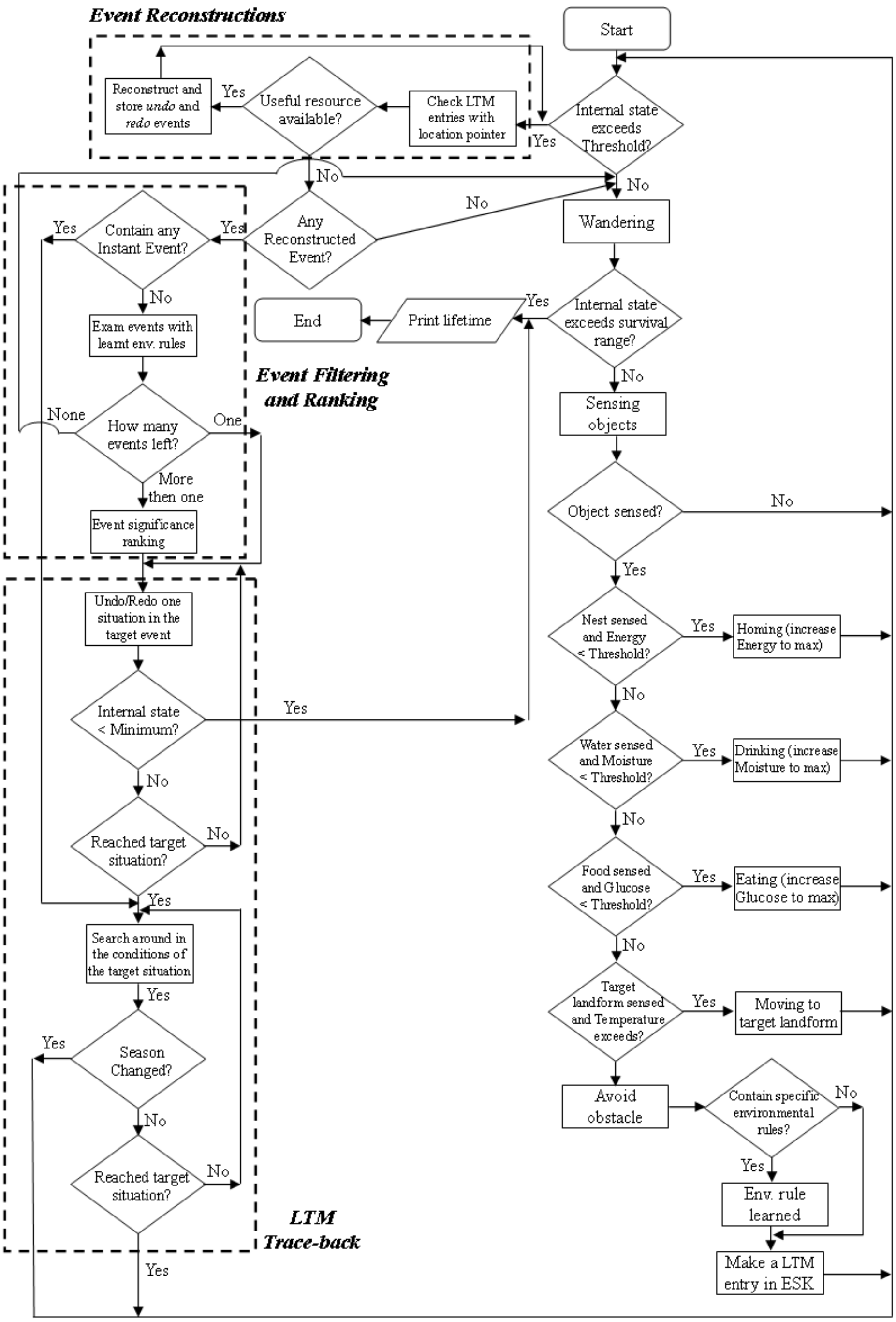

Figure B.15: Data flow diagram of Long-term Autobiographic Memory architecture.

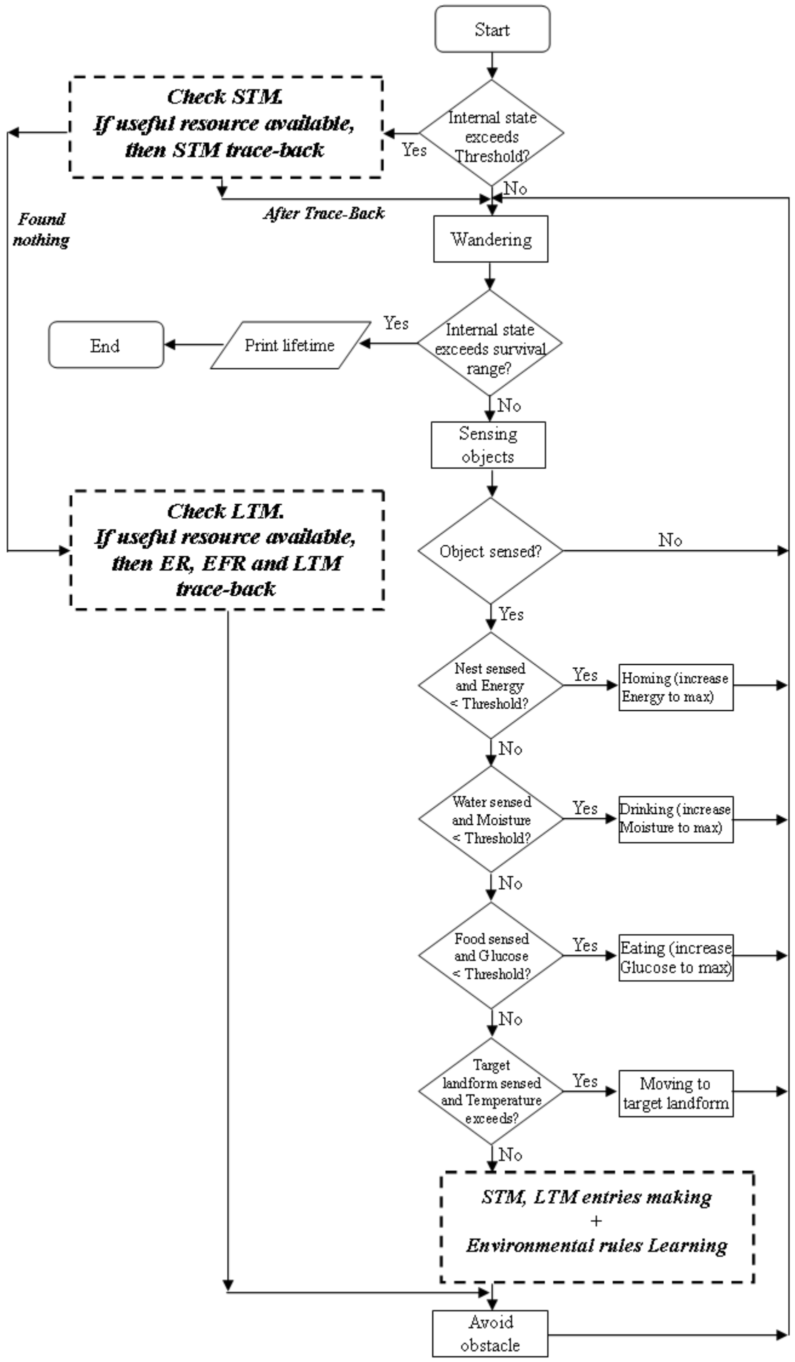

Figure B.16: Data flow diagram of Short-term and Long-term Autobiographic Memory architecture.

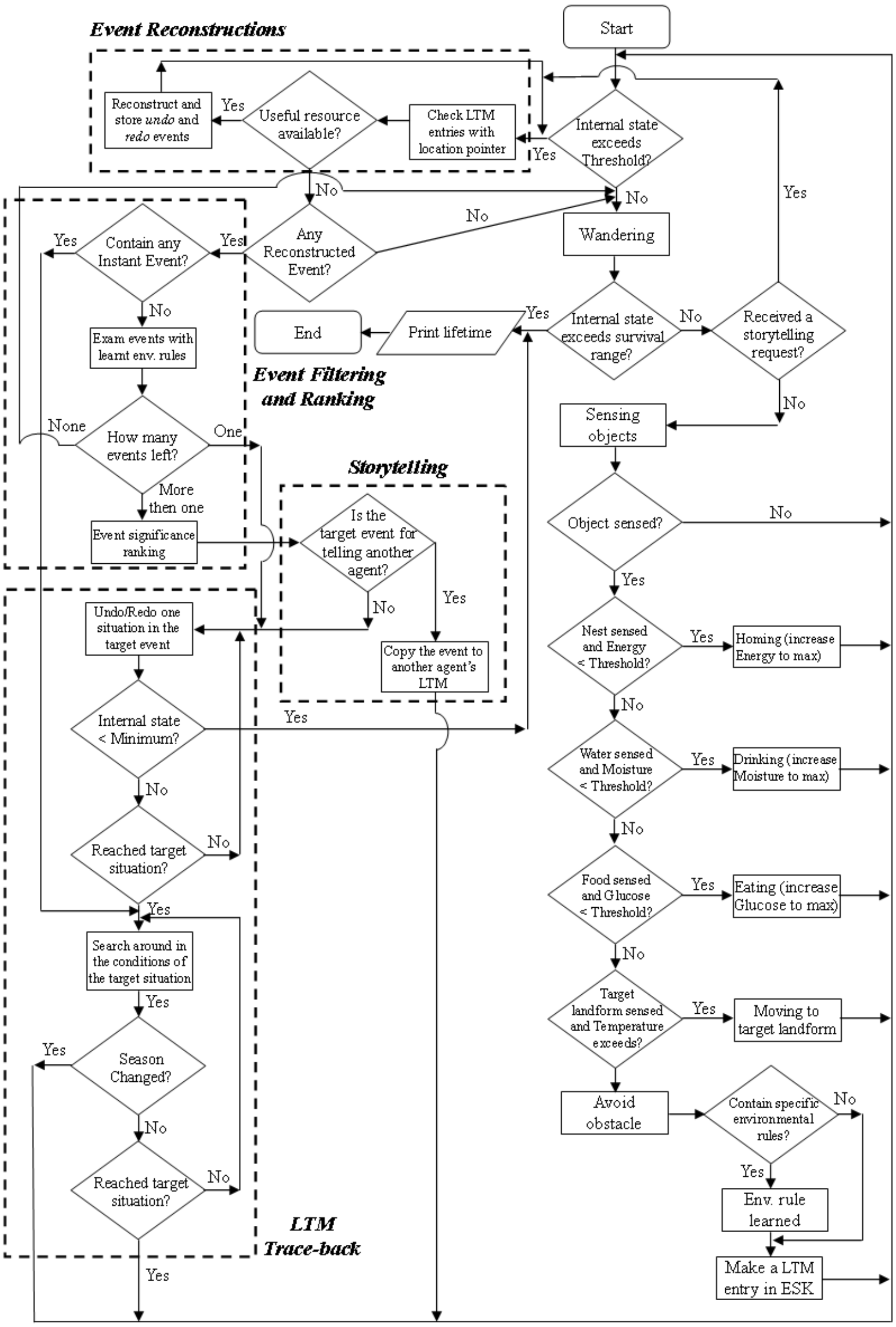

Figure B.17: Data flow diagram of Long-term Communicative Autobiographic Memory architecture.

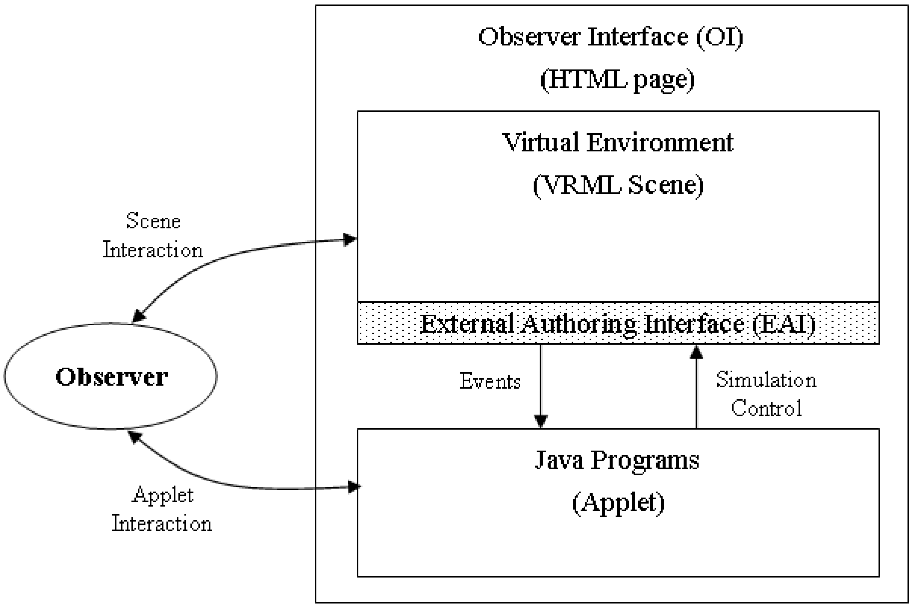

Figure B.18: System diagram for the overall interactions between observer, virtual environment and Java Applet.

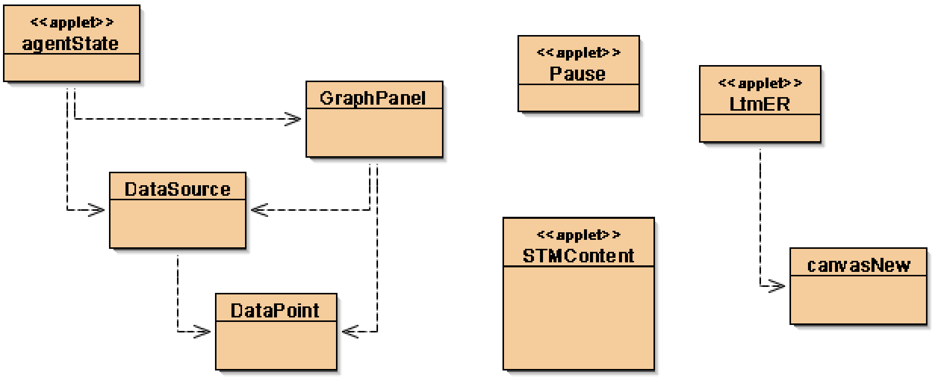

Figure B.19: Java class diagram for the Observer Interface (OI).

Appendix C

Program Source Code for Implementations

This appendix provides a CD-ROM which contains programming source code for the implementations of virtual environments and agent control architectures described in this thesis. The description of the content of each file is given in each sub-section which links to files under a specific folder in the CD-ROM.

The CD-ROM is attached to the inside of the back cover.

C.1 Single-Agent Experiments in Early Experimental Studies

- Referring to Sub-section 3.2.1 in Chapter 3.

- Folder name: **Pre-SA**.

- ***.gif** and ***.jpg** are graphical files for the virtual environment's texture mapping.

- **PR.wrl** is a VRML file for testing the *Purely Reactive* architecture. The static virtual environment has been created in this VRML file.

- **EB_mem50.wrl** is a VRML file for testing the *Trace-back* memory architecture with *Event-based* memory entry making mode and the memory length is 50 entries. The static virtual environment has been created in this VRML file.

- **TB100mem50.wrl** is a VRML file for testing the *Trace-back* memory architecture with *Event-based* memory entry making mode. The memory length is 50 entries and the agent makes an entry in every 100 time steps. The static virtual environment has been created in this VRML file.

- **Locality.wrl** is a VRML file for testing the *Locality* memory architecture. The static virtual environment has been created in this VRML file.

- **cosmo_win95nt_eng.exe** is the VRML plug-in to run this set of VRML files and to be used in all Windows platforms. By installing this plug-in, browsers like Internet Explorer or Navigator can support running VRML simulations.

C.2 Multi-Agent Experiments in Early Experimental Studies

- Referring to Sub-section 3.2.2 in Chapter 3.

- Folder name: **Pre-MA**.

- ***.gif** and ***.jpg** are graphical files for the virtual environment's texture mapping.

- **PR_2A.wrl** is a VRML file for testing 2 *Purely Reactive* agents running in the environment. The static virtual environment has been created in this VRML file.

- **TB_3A.wrl** is a VRML file for testing 3 *Trace-back* memory agents with communicating their memories in the simulation. The static virtual environment has been created in this VRML file.

- **cosmo_win95nt_eng.exe** is the VRML plug-in to run this set of VRML files and to be used in all Windows platforms. By installing this plug-in, browsers like Internet Explorer or Navigator can support running VRML simulations.

C.3 Studies of Long-term Autobiographic Memory Architectures

- Referring to Sub-section 5.2.2 in Chapter 5.

- Folder name: **Recent**.

- **stone4.gif**, **grass2.jpg**, **mine.png** and **sand.jpg** are graphical files for the virtual environment's texture mapping.

- **image_*.jpg** are icons used in Observer Interface to show agents' STM or LTM contents.

- ***.wav** are sound files used to model environmental sound effects in different areas.

- Folder named **objects** contains VRML object files for 3D models of *cactus*, *stone*, *mushroom* and *apple tree* in the virtual environment.

- ***.java** and ***.class** are source codes and class files respectively for the Observer Interfaces of different architectures.

- **PR.wrl** is a VRML file for testing the *Purely Reactive* architecture. The dynamic virtual environment has been created in this VRML file.

- **STM.wrl** is a VRML file for testing the *Short-term Memory* architecture. The dynamic virtual environment has been created in this VRML file.

- **LTM.wrl** is a VRML file for testing the *Long-term Autobiographic Memory* architecture. The dynamic virtual environment has been created in this VRML file.

- **STM_LTM.wrl** is a VRML file for testing the architecture with *Short-term Memory* and *Long-term Autobiographic Memory*. The dynamic virtual environment has been created in this VRML file.

- **LTM_Comm3.wrl** is a VRML file for testing 3 *Long-term Communicative Autobiographic Memory* agents with communicating their memories in the simulation.. The dynamic virtual environment has been created in this VRML file.

- **blaxxunContact51.exe** is the VRML plug-in to run this set of simulations and to be used in all Windows platforms. By installing this plug-in, browsers like Internet Explorer or Navigator can support running VRML simulations and getting extra Java class files for executing different Observer Interfaces.

- To start a specific simulation together with the relevant Observer Interface, open one of the following HTML files:
  - **interface_PR.html** is the single-agent simulation for *Purely Reactive* agent.
  - **interface_STM.html** is the single-agent simulation for *Short-term Memory* agent.
  - **interface_LTM.html** is the single-agent simulation for *Long-term Autobiographic Memory* agent.
  - **interface_STM_LTM.html** is the single-agent simulation for *Short-term Memory* and *Long-term Autobiographic Memory* agent.
  - **interface_LTM_Comm3.html** is the multi-agent simulation with 3 *Long-term Communicative Autobiographic Memory* agents.

Bibliography

Alba, J. W. and Hasher, L.: 1983, Is memory schematic?, *Psychological Bulletin* **93**, 203–231.

Arkin, R. C.: 1998, *Behavior-Based Robotics*, MIT Press, Cambridge, MA.

Atkinson, R. C. and Shiffrin, R. M.: 1968, Human memory: A proposed system and its control processes, *in* K. W. Spence and J. T. Spence (eds), *The Psychology of Learning and Motivation: Advances in Research and Theory*, Vol. 2, New York: Academic Press, pp. 89–195.

Aylett, R.: 1999, Narrative in virtual environments: Towards emergent narrative, *Proc. Narrative Intelligence*, AAAI Fall Symposium 1999, AAAI Press, pp. 83–86.

Aylett, R.: 2005, Letting go: Emergent narrative and interactive narrative, Artificial Intelligence Seminar Series, University of Bath. Presentation slides.

Aylett, R. and Luck, M.: 2000, Applying artificial intelligence to virtual reality: Intelligent virtual environments, *Applied Artificial Intelligence* **14**(1), 3–32.

Barclay, C. R.: 1986, Schematization of autobiographical memory, *in* D. C. Rubin (ed.), *Autobiographical Memory*, Cambridge University Press, Cambridge, USA, pp. 82–99.

Barthes, R.: 1966, Introduction to the structural analysis of narrative, *Image, Music, Text*, New York: Hill and Wang, pp. 79–124.

Bartlett, F. C.: 1932, *Remembering: A Study in Experimental and Social Psychology*, Cambridge University Press, Cambridge, Great Britain.

Beer, R. D.: 1995, On the dynamics of small continuous-time recurrent neural networks, *Adaptive Behavior* **3**(4), 469–509.

Beer, R. D.: 1996, Towards the evolution of dynamical neural networks for minimally cognitive behavior, *in* P. Maes, M. Mataric, J. Meyer, J. Pollack and S. Wilson (eds), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, MIT Press, pp. 421–429.

Blaxxun: 2004, Blaxxun technologies - products - blaxxun contact 5.1, http://www.blaxxun.com/en/products/contact/index.html. Last accessed 16-09-2005.

Bonabeau, E., Dorigo, M. and Theraulaz, G.: 1999, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press.

Bower, G. H., Black, J. B. and Turner, T.: 1979, Scripts in memory for text, *Cognitive Psychology* **11**, 177–220.

Brooks, R.: 1986, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **2**(1), 14–23.

Bruner, J.: 1991, The narrative construction of reality, *Critical Inquiry* **18**(1), 1–21.

Burghardt, G. M.: 1977, Ontogeny of communication, *in* T. A. Sebeok (ed.), *How Animals Communicate*, Indiana University Press, Bloomington, IN, pp. 71–79.

Cañamero, L., Avila-Garcia, O. and Hafner, E.: 2002, First esperiments relating behavior selection architectures to environmental complexity, *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, pp. 3024–3029.

Cass, S.: 2002, Mind games, *IEEE Spectrum* pp. 40–45.

Castelfranchi, C. and Falcone, R.: 1998, Principles of trust for MAS: Cognitive anatomy, social importance, and quantification., *Proceedings of Third International Conference on Multi-Agent Systems*, Paris, France, pp. 72–79.

Cavazza, M., Martin, O., Charles, F., Mead, S. J. and Marichal, X.: 2003, Interacting with virtual agents in mixed reality interactive storytelling, *Intelligent Virtual Agents 2003 (IVA 2003)*, Springer LNAI, pp. 231–235.

Center for the Study of Complex Systems at the University of Michigan: 2005, Swarmwiki, http://www.swarm.org/. Last accessed 16-09-2005.

Champandard, A.: 2003, *AI Game Development*, New Riders.

Clayton, N. S., Griffiths, D. P. and Dickinson, A.: 2000, Declarative and episodic-like memory in animals: Personal musings of a scrub jay, *in* C. Heyes and L. Huber (eds), *The Evolution of Cognition*, MIT Press, chapter 15, pp. 273–288.

Cohen, N. and Squire, L.: 1980, Preserved learning and retention of pattern analyzing skill in amnesia: Dissociation of knowing how and knowing that, *Science* **210**, 207–209.

Conway, M. A.: 1990, *Autobiographical Memory: An Introduction*, Open Univ. Press, Buckingham.

Conway, M. A.: 1992, A structural model of autobiographical memory, *in* M. A. Conway, D. C. Rubin, H. Sinner and E. W. A. Wagner (eds), *Theoretical Perspectives on Autobiographical Memory*, Dordrecht, the Netherlands: Kluwer, pp. 167–194.

Conway, M. A.: 1996, Autobiographical memories and autobiographical knowledge, *in* D. C. Rubin (ed.), *Remembering our past: Studies in autobiographical memory*, Cambridge Univ. Press, Cambridge, UK, pp. 67–93.

Conway, M. A. and Pleydell-Pearce, C. W.: 2000, The construction of autobiographical memories in the self memory system, *Psychological Review* **107**, 261–288.

Conway, M. A., Pleydell-Pearce, C. W. and Whitecross, S. E.: 2001, The neuroanatomy of autobiographical memory: A slow cortical potential study (scp) of autobiographical memory retrieval, *Memory and Language* **45**, 493–524.

Cuperlier, N., Laroque, P., Andry, P. and Gaussier, P.: 2004, Learning to plan and build experience via imitation in a social environment, *Proceedings of Artificial intelligence and Soft Computing / International Association of Science and TEchnology for Development (ASC/IASTED).*

Cyberbotics: 2005, Webots, http://www.cyberbotics.com/. Last accessed 16-09-2005.

Dautenhahn, K.: 1996, Embodiment in animals and artifacts, *AAAI FS Embodied Cognition and Action*, AAAI Press, pp. 27–32. Technical report FS-96-02.

Dautenhahn, K.: 1997, I could be you - the phenomenological dimension of social understanding, *Cybernetics and Systems* **28**(5), 417–453.

Dautenhahn, K.: 1999a, Embodiment and interaction in socially intelligent life-like agents, *in* C. L. Nehaniv (ed.), *Computation for Metaphors, Analogy and Agent*, Vol. 1562 of *Springer Lecture Notes in Artificial Intelligence*, Springer, pp. 102–142.

Dautenhahn, K.: 1999b, The lemur's tale - story-telling in primates and other socially intelligent agents, *Narrative Intelligence*, AAAI Fall Symposium.

Dautenhahn, K.: 2002, The origins of narrative in search for the transactional format of narratives in humans and other social animals, *Cognition and Technology: Co-existence, Convergence, Co-evolution (IJCT)* pp. 97–123. John Benjamins Publishing Company.

Dautenhahn, K.: 2003, Stories of lemurs and robots - the social origin of story-telling, *in* M. Mateas and P. Sengers (eds), *Narrative Intelligence*, John Benjamins Publishing, pp. 63–90.

Dautenhahn, K. and Christaller, T.: 1996, Remembering, rehearsal and empathy - towards a social and embodied cognitive psychology for artifacts, *in* S. O'Nuallain and P. McKevitt (eds), *Two sciences of the mind. Readings* *in cognitive science and consciousness*, John Benjamins North America Inc, pp. 257–282.

Dautenhahn, K. and Coles, S.: 2001, Narrative intelligence from the bottom up: A computational framework for the study of story-telling in autonomous agents, *The Journal of Artificial Societies and Social Simulation (JASSS)* **4**(1), 1–15. http://jasss.soc.surrey.ac.uk/4/1/1.html (Last accessed 16-09-2005).

Dautenhahn, K. and Nehaniv, C. L.: 1998, Artificial life and natural stories, *International Symposium on Artificial Life and Robotics (AROB III)*, Vol. 2, Beppu, Oita, Japan, pp. 435–439.

Dolan, R. J.: 2002, Emotion, cognition, and behavior, *Science* **298**(5596), 1191–1194.

Elman, J. L.: 1990, Finding structure in time, *Cognitive Science* **14**, 179–211.

Franklin, S. and Graesser, A.: 1997, Is it an agent, or just a program? a taxonomy for autonomous agent., *Proc. of the Third International Workshop on Agent Theories, Architectures, and Languages, Intelligent Agents III*, New York:Springer-Verlag, pp. 21–35.

Freud, S.: 1963, Three essays on the theory of sexuality, *in* J. Strachey (ed.), *The Standard Edition of the Complete Works of Freud*, Vol. 7, Hogarth Press, London.

Goguen, J.: 2001, The structure of narrative, http://www-cse.ucsd.edu/classes/sp01/cse171/story.html. Last accessed 16-09-2005.

Heidegger, M.: 1972, *On Time and Being*, Harper Torchlight Books.

Ho, W. C., Avila-García, O. and Nehaniv, C. L.: 2005a, Assessing the performance of different behavior selection architectures in a large and complex virtual environment, *Proceedings of IEEE Congress on Evolutionary Computation - Special Session: Complex Adaptive Systems*, IEEE, pp. 359–365.

Ho, W. C., Dautenhahn, K. and Nehaniv, C. L.: 2003, Comparing different control architectures for autobiographic agents in static virtual environments, *Intelligent Virtual Agents 2003 (IVA 2003)*, Springer LNAI, pp. 182–191.

Ho, W. C., Dautenhahn, K. and Nehaniv, C. L.: 2005b, Autobiographic agents in dynamic virtual environments - performance comparison for different memory control architectures, *Proceedings of IEEE Congress on Evolutionary Computation - Special Session: Artificial Life*, pp. 573–580.

Ho, W. C., Dautenhahn, K. and Nehaniv, C. L.: submitted, A bottom-up study in autonomous agents with autonomous memory and narrative storytelling in a dynamic virtual environment, *Cognitive Systems Research*.

Ho, W. C., Dautenhahn, K., Nehaniv, C. L. and te Boekhorst, R.: 2004, Sharing memories: An experimental investigation with multiple autonomous autobiographic agents, *IAS-8, 8th Conference on Intelligent Autonomous Systems*, IOS Press, Amsterdam, NL, pp. 361–370.

Hornung, A., Lakemeyer, G. and Trogemann, G.: 2003, An autonomous real-time camera agent for interactive narratives and games, *Intelligent Virtual Agents 2003 (IVA 2003)*, Springer LNAI, pp. 236–243.

Isla, D. and Blumberg, B.: 2002, New challenges for character-based ai for games, *AAAI Spring Symposium on AI and Interactive Entertainment*, Palo Alto, CA.

Johnson-Laird, P. N.: 1980, Mental models in cognitive science, *Cognitive Science* **4**(1), 71–115.

Kelly, M. P. and Dickinson, H.: 1997, The narrative self in autobiographical accounts of illness, *The Sociological Review* **45**(2), 254–278.

Kolodner, J. L.: 1993, *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, CA.

Leake, D.: 1996, CBR in context: The present and future, *in* D. Leake (ed.), *Case-Based Reasoning: Experiences, Lessons and Future Directions*, MIT Press, pp. 3–30.

Linde, C.: 1993, *Life Stories: The Creation of Coherence*, Oxford University Press.

Lowe, R. J., Cañamero, L., Nehaniv, C. L. and Polani, D.: 2004, The evolution of affect-related displays, recognition and related strategies, *in* J. Pollack, M. Bedau, P. Husbands, T. Ikegami and R. Watson (eds), *ALIFE IX: Proc. 9th Intl. Conf. on the Simulation and Synthesis of Living Systems*, MIT Press, Cambridge, MA, pp. 176–181.

Markovitch, S. and Scott, P. D.: 1988, The role of forgetting in learning, *Proceedings of The Fifth International Conference on Machine Learning*, Morgan Kaufmann, pp. 459–465.

Mateas, M.: 1999, An Oz-centric review of interactive drama and believable agents, *Artificial Intelligence Today* pp. 297–328.

Mateas, M. and Stern, A.: 2002, Architecture, authorial idioms and early observations of the interactive drama facade, *Technical report*, School of Computer Science, Carnegie Mellon University. CMU-CS-02-198.

McGaugh, J. L. and Cahill, L.: 2002, Emotion and memory: Central and peripheral contributions, *in* R. J. Davidson, K. R. Scherer and H. H. Goldsmith (eds), *Handbook of Affective Science*, Oxford University Press, pp. 93–116.

Meyer, J.-A.: 1995, The animat approach : Simulation of adaptive behavior in animals and robots, *in* F. Alexandre and J.-D. Kant (eds), *Actes de la conference Neurosciences Pour l'Ingénieur*, Publications Loria, Nancy, pp. 1–21.

Miller, G. A.: 1956, The magical number seven plus or minus two: Some limits on our capacity for processing information, *Psychological Review* **63**, 81–97.

Minsky, M.: 1975, A framework for representing knowledge, *in* P. H. Winston (ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, pp. 211–277.

Minsky, M.: 1982, Why people think computers can't, *AI Magazine* **3**(4), 3–15. http://web.media.mit.edu/ minsky/papers/ComputersCantThink.txt (Last accessed 16-09-2005).

Nehaniv, C. L.: 1999, Narrative for artifacts: Transcending context and self, *Narrative Intelligence*, AAAI Fall Symposium 1999, AAAI Press, pp. 101–104. Technical Report FS-99-01.

Nehaniv, C. L. and Dautenhahn, K.: 1998a, Embodiment and memories - algebras of time and history for autobiographic agents, *14th European Meeting on Cybernetics and Systems Research*, Embodied Cognition and AI symposium, pp. 651–656.

Nehaniv, C. L. and Dautenhahn, K.: 1998b, Semigroup expansions for autobiographic agents, *in* T. Imaoka and C. L. Nehaniv (eds), *Proceedings of the*

First Symposium on Algebra, Languages and Computation), University of Aizu, pp. 77–84.

Nehaniv, C. L., Dautenhahn, K. and Loomes, M. J.: 1999, Constructive biology and approaches to temporal grounding in post-reactive robotics, *Sensor Fusion and Decentralized Control in Robotic Systems II*, Vol. 3898, pp. 156–167. SPIE.

Nehaniv, C. L., Polani, D., Dautenhahn, K., te Boekhorst, R. and Cañamero, L.: 2002, Meaningful information, sensor evolution, and the temporal horizon of embodied organisms, *Artificial Life VIII*, MIT Press, pp. 345–349.

Neisser, U.: 1986, Nested structure in autobiographical memory, *in* D. C. Rubin (ed.), *Autobiographical Memory*, Cambridge University Press, Cambridge, UK, pp. 71–88.

Nelson, K.: 1993, The psychological and social origins of autobiographical memory, *Psychological Science* **4**, 7–14.

Norman, D. A. and Bobrow, D. G.: 1975, On the role of active memory processes in perception and cognition, *in* C. N. Cofer (ed.), *The Structure of Human Memory*, W. H. Freeman and Company, pp. 114–132.

Peters, C. and O'Sullivan, C.: 2002, Synthetic vision and memory for autonomous virtual humans, *Computer Graphics Forum* **21**(4), 743–753.

Russell, S. and Norvig, P.: 1994, *Artificial Intelligence: A Modern Approach*, Prentice Hall, p. 23.

Saraswat, V. A., Jagadeesan, R. and Gupta, V.: 1994, Foundations of timed concurrent constraint programming, *1994 IEEE Symposium on Logic in Computer Science*, pp. 71–80.

Schank, R. C.: 1982, *Dynamic Memory: Theory of Reminding and Learning in Computers and People*, Cambridge University Press.

Schank, R. C. and Abelson, R. P.: 1977, *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum Assicuates, Hillsdale, New Jersey.

Sengers, P.: 2000, Narrative intelligence, *in* K. Dautenhahn (ed.), *Human Cognition and Social Agent Technology*, John Benjamins, chapter 1, pp. 1–26.

Sengers, P.: 2003, Narrative and schizophrenia in artificial agents, *in* M. Mateas and P. Sengers (eds), *Narrative Intelligence*, Amsterdam: John Benjamins, pp. 259–278.

Shafir, S. and Roughgarden, J.: 1996, The effect of memory length on the individual fitness of a lizard, *in* R. K. Belew and M. Mitchell (eds), *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Santa Fe Institute Studies in the Science of Complexity, Addison Wesley, Reading, MA, pp. 173–182.

Smith, F.: 1998, *The Book of Learning and Forgetting*, Teachers College Press.

Sowa, J. F.: 1999, *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Course Technology.

Steels, L.: 1992, The PDL reference manual, VUB AI Lab Memo, AI laboratory, Brussels. Pages 92-5.

Steels, L.: 2003, Intelligence with representation, *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* **361**(1811), 2381–2395.

Stern, A.: 2003, Virtual babyz: Believable agents with narrative intelligence, *in* M. Mateas and P. Sengers (eds), *Narrative Intelligence*, John Benjamins Publishing, pp. 215–227.

Sternberg, R.: 1994, *In search of the human mind*, New York: Harcourt Brace.

Sutton, R. S. and Barto, A. G.: 1998, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.

Templeton, J. L. and Giraldeau, L.: 1996, Vicarious sampling: the use of personal and public information by starlings foraging in a simple patchy environment, *Behavioural Ecology and Sociobiology* **38**, 105–114.

Tulving, E.: 1972, Episodic and semantic memory, *in* E. Tulving and W. Donaldson (eds), *Organization of Memory*, Academic Press, New York, pp. 381–403.

VICTEC: 2005, Virtual ICT with empathic characters, http://www.victec.org/. Last accessed 29-09-2005.

Watkins, C. J. C. H.: 1989, *Learning from Delayed Rewards*, PhD thesis, Cambridge University.

Witkowski, M., Artikis, A. and Pitt, J.: 2000, Trust and cooperation in a trading society of objective-trust based agents, *Proceeding of Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies*, pp. 127–136.