Autobiographic Agents in Dynamic Virtual Environments - Performance Comparison for Different Memory Control Architectures

Wan Ching HO, Kerstin DAUTENHAHN, Chrystopher L. NEHANIV
Adaptive Systems Research Group
School of Computer Science, University of Hertfordshire
College Lane, Hatfield, Hertfordshire, AL10 9AB, UK
W.C.Ho, K.Dautenhahn, C.L.Nehaniv@herts.ac.uk

Abstract- In this paper, we extend our previous work 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, with a kind of communication of experiences in multi-agent experiments. In the current work 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 word. A large-scale and dynamic virtual environment is created to compare the performance of various types of agents with various memory control architectures, and each agent's behaviour is observed and analyzed together with lifespan measurements. Results confirm our previous research hypothesis that autobiographic memory can prove beneficial – indicating increases in the lifespan of an autonomous, autobiographic, minimal agent. Furthermore, the utility of combining Long-term Memory with Short-term Memory is established. We finally discuss the environmental factors influencing the performance of each architecture and the areas for future work.

1 Introduction

Remembering past events certainly helps an animal or a human to learn from experience; therefore, for many years, researchers in the fields of biology, artificial life and psychology have been investigating how memory influences the behaviour of both human and other animals. Our previous work has studied the length of autobiographic memory embedded in an artificial animal to improve its foraging behaviour [Ho et al., 2003] and the effect of sharing memories with other agents surviving in the same environment [Ho et al., 2004]. In this paper, we study the design space of agent control architectures with autobiographic memory and aim to develop a more sophisticated memory control architecture which is generic and adaptive for a minimal Artificial Life agent. Autobiographic memory is a specific kind of episodic memory, which in humans develops in childhood [Nelson, 1993]. Autobiographic agents are agents which are embodied and situated in a particular environment (including other agents), and which dynamically reconstruct their individual history (autobiography) during their lifespan, as defined in [Dautenhahn, 1996]. Autobiographic memory is an important ingredient for socially intelligent agents [Dautenhahn, 1999]. Moreover, it is useful for synthesizing agents that can behave adaptively [Nehaniv and Dautenhahn, 1998a], and for designing agents that apparently 'have a life' and thus appear believable and acceptable to humans [Dautenhahn, 1998].

For many decades researchers in psychology have widely studied schema theories which indicate the representations and encoding processes of human memory [Alba and Hasher, 1983]. As autobiographic memory is part of the human long-term episodic memory, we apply schema theories in designing autobiographic memory for autonomous Artificial Life agents. Here, an important feature is *event reconstruction*, which is reflected in the memory processes of selection and abstraction. A selection process specifies that all incoming stimuli are selectively remembered for our memory representation, and the abstraction process denotes that the meaning of an event or a message is stored without entirely refering to its original contents.

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; this broader *temporal horizon* can allow for planning for future actions and learning from past or imagined sequences of events. Our work focuses on realizing such mechanisms in artificial autonomous agents [Nehaniv and Dautenhahn, 1998a, Nehaniv, 1999, Nehaniv et al., 2002].

1.1 Related Work

The earlier study in [Ho et al., 2003] 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, compared in two distinct static environments. Experimental results produced evidence which confirmed 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].

We have also investigated multiple autobiographic agents able to share an experienced sequences 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 [Ho et al., 2004]. 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.

In the following sections we first describe the 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 2). Next we focus on three main agent control architectures (Section 3): Purely Reactive (PR), Short-term Memory (STM) and Long-term Memory (LTM). For each of them we illustrate the design concepts of the architecture in detail; particularly for the LTM architecture, we specify its main features, including Event Specific Knowledge (ESK), Event Reconstruction (ER), and Event Filtering and Ranking (EFR) processes. This is followed by the section on experiments and result analysis (Section 4). In the final section we summarize the conclusions and discuss the varied potential for future developments.

2 The Complex Dynamic Virtual Environment and Agent Embodiment

In order to create rich possibilities of temporal sequences of events for the agents and to examine the performance of our agent control architectures, a large, dynamic and complex "nature-like" virtual environment has been created by using VRML and Java programming languages. This environment is fairly different from other simple and flat agent testbeds since it has various types of resources, most of them dynamically distributed on different kinds of landform. Figure 1 illustrates the virtual environment model from two different perspectives.

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 a previously encountered situation by following actions in reverse order).

2.1 Environment Structure

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

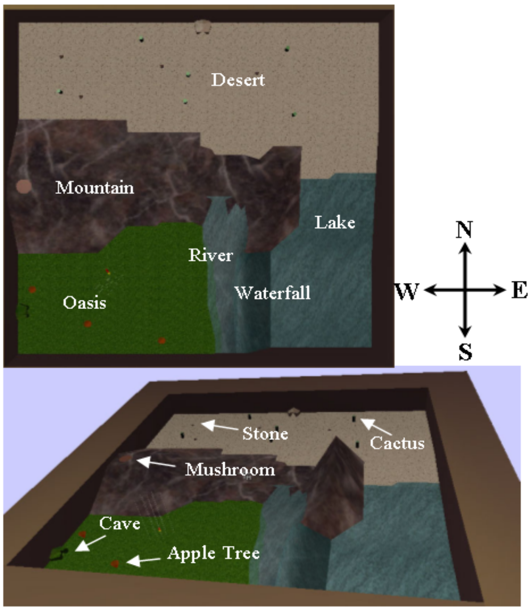


Figure 1: The simulated dynamic virtual environment viewed from two different perspectives.

- *Oasis* - this is generally a warm and flat area, which has three *Apple Trees* in the summer.
- *Desert* - a hot and flat area where 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 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 the 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 locates next to the oasis. Agents are able to swim in the river, but they cannot swim 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 downstream 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 water from the lake.
- 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.

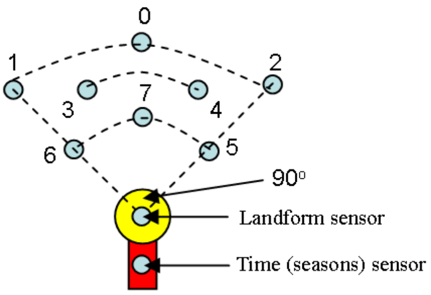


Figure 2: Hit-Ray sensors 0 - 7 for sensing both objects and Landforms, agent body has a landform sensor and a time sensor.

Alternatively, two seasons *Summer* and *Winter*, have been simulated in the environment to have a higher level of environmental dynamics (Table 1). Each season has the same duration but different effects on a) the level of heat and cold in different areas of the environment, b) dynamic resources allocation and c) the accessibility of the river.

2.2 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; agent body has a landform sensor and also a time sensor for sensing the current season of the environment. Figure 2 shows the distribution of these sensors.

All agents have a finite lifespan and are required to wander in the environment as their basic behavior. The survival of an agent depends on maintaining homeostasis for its four internal physiological 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 ideal - 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 ideal 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 ideal range again. If the value of one of the internal variables (*glucose*, *moisture* and *energy*) 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.

Internal variables	Relevant resource (effects)
Glucose	Apple Tree (+ 100%) Mushroom (+ 100%) Cactus (+ 10%)
Moisture	Apple Tree (+ 100%) Cactus (+ 10%)
Energy	Cave (+ 100%) Cactus - tough 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 2: Relationships between agents' internal variables and different resources and contexts in the environment.

The relationship between internal physiological variables and various types of resources in the environment are shown in Table 2.

3 Agent Memory Control Architectures

We aim to develop appropriate autobiographic memory architectures on top of a basic subsumption control architecture in order to enhance the agents' performance in surviving in a dynamic environment. To achieve this goal, we designed and implemented three different control architectures: Purely Reactive (PR), Short-term memory (STM) and Long-term memory (LTM). In addition to these three architectures, in the experiments section we also investigate the fourth type, which is built by combining STM and LTM into one architecture in order to broaden the agents' temporal horizon by taking advantage of more sophisticated memory control algorithms. For both STM and LTM architectures, their distinctive memory layers control algorithms are built on top of the PR architecture with the aim of inhibiting the execution of behavior from the PR architecture optionally.

3.1 Purely Reactive Architecture (PR)

In this paper, we define a PR agent as an agent who makes its decisions for executing behavior totally based on its internal physiological variables and sensory inputs. Therefore we designed and implemented the PR agent by using a basic subsumption control architecture [Brooks, 1985], as illustrated in Figure 3. 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. 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, but the internal vari-

	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 1: Environmental heats, resource allocations and river conditions in the dynamic environment

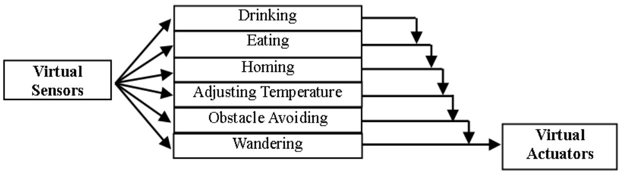


Figure 3: Behaviour hierarchy which is based on the subsumption architecture for a Purely Reactive (PR) agent.

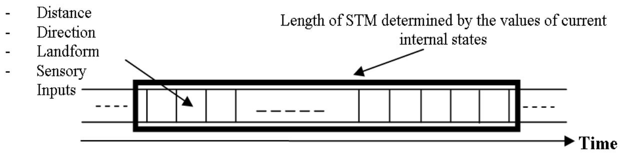

- Distance
- Direction
- Landform
- Sensory Inputs
Length of STM determined by the values of current internal states
Time

Figure 4: Short-term Memory (STM) with information indicating contents of each entry and the change of its length.

able which needs that particular resource is higher than the corresponding threshold.

3.2 Short-term Memory Architecture (STM)

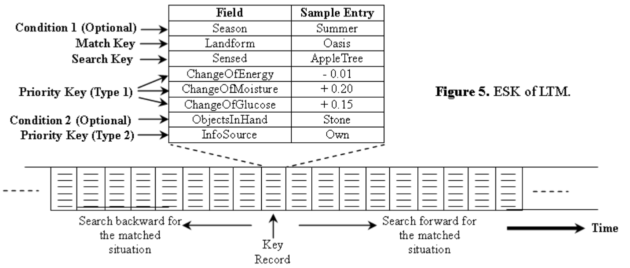
On the basis of the design of a PR architecture, STM with memory Trace-back possesses a memory module on top of the subsumption architecture. An STM agent has a dedicated mechanism for making memory entries as the remembering process, and using the memory as a tracing process. In the case of the Trace-back mechanism, the agent has a finite number of memory entries. 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 behavior. This is called *Event-based* memory entry making mode. 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 travelled (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 infinite index number. The abstract model of STM is shown in Figure 4.

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. Once Trace-back has started, the agents will simply 'undo' all previous behaviors. This mechanism has a close connection to the algebraic 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 actions undoing all memories entries and has reached the target resource. At this moment, the agent will start sensing around for the resource. During the Trace-back process, we have also introduced noise (Gaussian, standard deviation 5°) to slightly alter the Direction value when the agent is retrieving an entry from its STM. 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 Track-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 facing the environmental dynamics, such as unstable resource distributions and the flowing direction of the river and waterfall in summer, these sometime cause the agent to fail in executing the Trace-back process, where upon the agent will erase all the memory entries in STM. As an STM agent can not remember an unlimited number of entries; the number of entries in STM is determined by estimating the costs of executing 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 4. 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].

3.3 Long-Term Autobiographic Memory (LTM)

Inspired by human long-term memory [Alba and Hasher, 1983] and autobiographic memory



Field	Sample Entry
Season	Summer
Landform	Oasis
Sensed	AppleTree
ChangeOfEnergy	- 0.01
ChangeOfMoisture	+ 0.20
ChangeOfGlucose	+ 0.15
ObjectsInHand	Stone
InfoSource	Own

Figure 5: Event Specific Knowledge (ESK) of Long-term Memory (LTM).

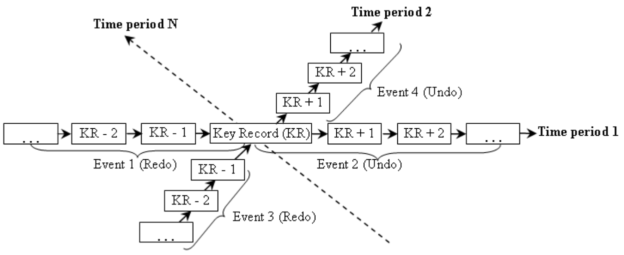


Figure 6: Result from Event Reconstruction (ER) process - autobiographic memory schema.

[Conway, 1992] models from related research in psychology, 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.

3.3.1 Event Specific Knowledge (ESK)

An LTM agent surviving in the dynamic environment has a long list of 'history', 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 physiological variables; the name of each field in a LTM record and sample entries are shown in Figure 5.

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 precision when filtering out events in the Event Filtering and Ranking (EFR) processes, more details are provided in Sub-section 3.3.3.

3.3.2 Event Reconstruction (ER) Process

The Event Reconstruction process proceeds as follows: if one of the internal variables of the LTM agent is lower than the threshold, the agent will search all records in its LTM and to retrive 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 amount of records from ESK and reconstructs each event by using the 'meaningful' *Search Key* (Figure 5). 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 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 ER process. The *Key Record* (Figure 5) 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 6 shows, after all possible events have been reconstructed by records from ESK, an autobiographic memory schema dedicated for satisfying a specific internal physiological variable.

With regard to the dynamic virtual environment introduced in Section 2, all possible events which are generated by the environment and can be remembered by the agent in its LTM, are classified in Table 3.

3.3.3 Event Filtering and Ranking (EFR) Processes

After a 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 of them 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 doesn't match any target situation, in the second step of EFR processes some events which are inappropriate to the

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), Temperature(D)	River	Summer only
Looking for Lake	Moisture(I), Temperature(D)	Lake	Summer only
Looking for Caves	Energy(I)	Caves at Mountain's foot	
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	Temperature(D), Glucose(I) (from Mushroom)	Mountain Area	
Location of Oasis area	Temperature(D), Glucose(I), Moisture(I) (from Apple Tree))	Oasis Area	
Location of Desert area	Temperature(I), 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 3: Possible events for Long-term Memory (LTM) agent to remember (I: Increase, D: Decrease).

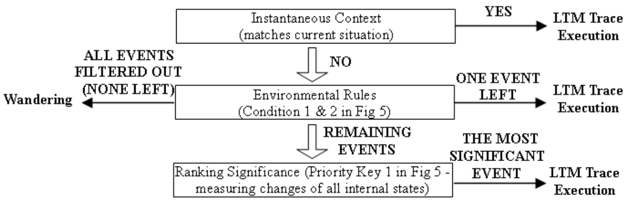

Instantaneous Context
(matches current situation)
YES
LTM Trace Execution
NO
ALL EVENTS FILTERED OUT (NONE LEFT)
Wandering
Environmental Rules
(Condition 1 & 2 in Fig 5)
ONE EVENT LEFT
LTM Trace Execution
REMAINING EVENTS
THE MOST SIGNIFICANT EVENT
Ranking Significance (Priority Key 1 in Fig 5 - measuring changes of all internal states)
LTM Trace Execution

Figure 7: Event Filtering and Ranking (EFR) processes.

current situations will be filtered out by using environmental rules (shown as *Condition 1 & 2* in Figure 5). 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) to do the *LTM Trace*. The most significant event is calculated by measuring the total change of internal physiological variables *glucose*, *moisture* and *energy*. EFR processes are illustrated in Figure 7.

To execute a *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 to the mountain area, and then 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 only searches for a *Stone* after it reaches the desert area.

4 Experiments

To measure the performance of four types of agent architecture 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 2.0GHz PC with 512MB 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 starting position for all agents in the experiments is in the center of the oasis area.

Apart from the main measured dependent variable - the average lifespan in 10 experimental runs of each agent control architecture, we also observe the capability of each architecture in keeping up its internal variables in the ideal value range. Therefore, in each experimental run we recorded the change of all internal variables by monitoring over time. We expect that a desirable control architecture for agents surviving in a highly dynamic environment should be able to maintain all internal variables in the ideal value range - in this study, this means 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).

4.1 Results

Figure 8 shows lifespans of four types of agent. Since we are also interested in observing each agent's comprehensive behaviour generated from its unique control architecture, Figure 9 and Figure 10 illustrate how well all four types of agents maintain their internal physiological variables.

4.2 Discussion and Analysis

Figure 8 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

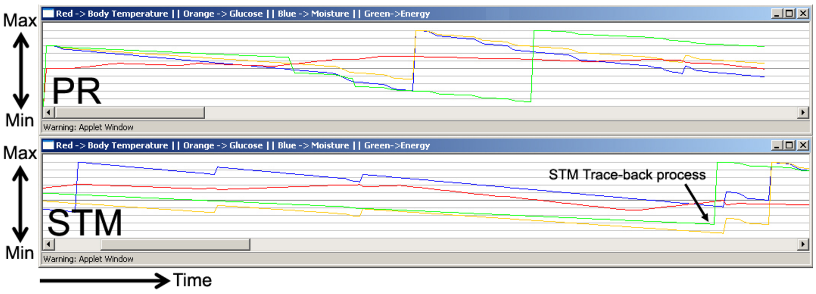

Max
Red -> Body Temperature || Orange -> Glucose || Blue -> Moisture || Green->Energy
PR
Min
Warning: Applet Window
Max
Red -> Body Temperature || Orange -> Glucose || Blue -> Moisture || Green->Energy
STM Trace-back process
STM
Min
Warning: Applet Window
Time

Figure 9: 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 lengh 25000 steps.

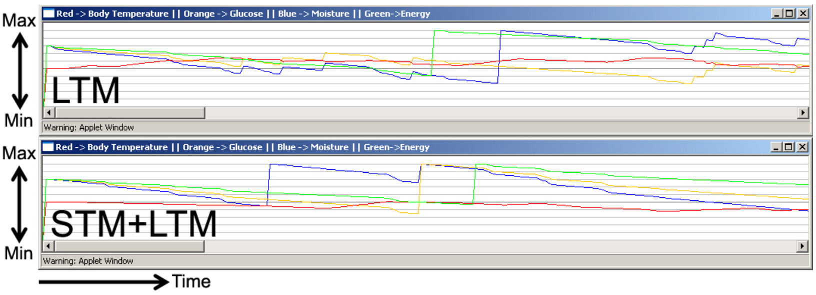

Max
Red -> Body Temperature || Orange -> Glucose || Blue -> Moisture || Green->Energy
LTM
Min
Warning: Applet Window
Max
Red -> Body Temperature || Orange -> Glucose || Blue -> Moisture || Green->Energy
STM+LTM
Min
Warning: Applet Window
Time

Figure 10: Examples of internal variables' changes of a Long-term Memory (LTM) agent (upper graph) and a Short-term Memory plus Long-term Memory (STM+LTM) agent (lower graph), in time window of lengh 25000 steps.

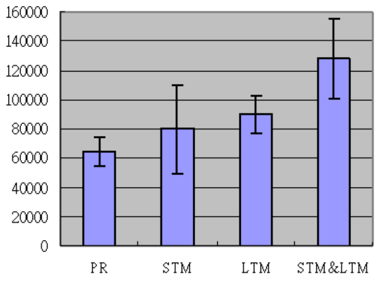


Figure 8: Experimental results with confidence values (error bars) showing the average lifespan of the 4 different agent control architectures running 10 times in each condition in the environment.

of the STM agent, with a high range of 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.

The agent with STM+LTM has the highest average lifespan; this result is reflected in agent's 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 memory appear to be capable to maintain their physiological variables in the ideal value range most of the time, comparing with PR and STM agents, as shown in Figure 9 and Figure 10. The reason is that STM agents need to spend a certain amount of time and internal variables' energy to execute the *Trace-back* process in order to reach the target resource or landform, as indicated in Figure 9.

Compared to the previous work [Ho et al., 2003, Ho et al., 2004], in which we studied a single PR or STM agents surviving in a flat and static virtual environment with constant resource distributions; in this work 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*. Experiment 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 when climbing up the mountain or getting stuck in the lake area will cost more internal variables than wandering in other areas. They tend to stay wandering in the area where they can find every necessary resource to maintain their internal variables in the ideal value 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* process, 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 be found in the area and 2) switch to other targets for fulfilling the same need of internal need or other internal needs.

5 Conclusions and Future Work

Through experimental results on agents with different memory control architectures surviving in the dynamic virtual environment, 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 physiological variables over time. We have also shown the design of improved STM and LTM control architectures in detail, and the combination of them produces the best result in average lifespan coping with dynamic environmental conditions in a large-scale virtual environment.

In future, this work can be extended in many ways. Firstly, STM and LTM agents' further potential can be discovered by running them in a even larger scale dynamic environment with different object distribution rules. Secondly, we can improve the current architecture to get a better coordination of functionalities of STM and LTM; for example, motivation-based decision making mode can be used to solve the conflicts between the executions of output from STM and LTM. Last but not least, we are interested in designing and implementing multiple autobiographic agents in 'story-telling' virtual environments, in which agents can communicate their experience to each other or to humans.

Realizations in artificial agents of story-telling and narrative features can benefit from the temporal horizon of autobiographic agents using temporally extended meaningful information [Ho et al., 2004, Nehaniv, 1999, Nehaniv et al., 2002], and should thus benefit from LTM. In addition to this, by receiving and re-using (and verifying) events from other agents ('stories'), an agent with Long-term Autobiographic Memory may be able to recognize other agents individually. During the agent's lifetime, since it manages the history of interactions with other agents in the past and there will be an increasing influence on agents' behaviours from this history, some processes in EFR can later be developed into having higher priorities in choosing which event is to be executed; this implies that certain levels of trust, as well as distrust, could be built up between agents as time passes by.

Bibliography

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

[Blaxxun, 2004] Blaxxun (2004). Blaxxun technologies - products - blaxxun contact 5.1. http://www.blaxxun.com/en/products/contact/index.html.

[Brooks, 1985] Brooks, R. (1985). A robust layered control system for a mobile robot. MIT AI Lab Memo, 864.

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

[Dautenhahn, 1996] Dautenhahn, K. (1996). Embodiment in animals and artifacts. In *AAAI FS Embodied Cognition and Action*, pages 27–32. AAAI Press. Technical report FS-96-02.

[Dautenhahn, 1998] Dautenhahn, K. (1998). The art of designing socially intelligent agents - science, fiction, and the human in the loop. *Applied Artificial Intelligence*, 12(7-8):573–617.

[Dautenhahn, 1999] Dautenhahn, K. (1999). Embodiment and interaction in socially intelligent life-like agents. In Nehaniv, C. L., editor, *Computation for Metaphors, Analogy and Agent*, volume 1562 of *Springer Lecture Notes in Artificial Intelligence*, pages 102–142. Springer.

[Ho et al., 2003] Ho, W. C., Dautenhahn, K., and Nehaniv, C. L. (2003). Comparing different control architectures for autobiographic agents in static virtual environments. In *Intelligent Virtual Agents 2003 (IVA 2003)*, pages 182–191. Springer LNAI 2792.

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

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

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

[Nehaniv and Dautenhahn, 1998b] Nehaniv, C. L. and Dautenhahn, K. (1998b). Semigroup expansions for autobiographic agents. In *Proceedings of the First Symposium on Algebra, Languages and Computation (30 October-1 November 1997, University of Aizu, Japan)*, pages 77–84. Osaka University.

[Nehaniv et al., 2002] 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. In *Artificial Life VIII*, pages 345–349. MIT Press.

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