2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

Enhancing Agent Autonomy and Adaptive Behavior by Knowledge Abstraction

Kumari Wickramasinghe and Damminda Alahakoon
School of Business Systems, Monash University, Australia
{kumari.wickramasinghe, damminda.alahakoon }@infotech.monash.
edu.au

Abstract

Agents are expected to exhibit autonomous behaviour by mapping sensory inputs to valid output actions by adapting to the changes in dynamic environments. To accomplish this, several agent architectures have been implemented, but to date it remains as a task of the programmer to explicitly specify the states that may be encountered in an environment with the corresponding action. Instead, in order to perform autonomously, the agent should possess a dynamically modifiable path referred also as a world model between sensory inputs and actuator outputs. Inspired by biological systems and psychology, this paper presents a methodology for building such an evolving world model through the knowledge of abstract concepts to enhance agent autonomy. In the proposed method the world model demonstrates the relationships between environment state to output actions and each relationship is denoted as a trace. The world model originates with a few initial traces and evolves as the agent receives experience in the environment. This provides a mechanism to retrieve a suitable action from a pool of traces by simultaneously activating them with the encountered environment state. Thus the action selected by the agent reflects the summed up content of all the activated traces providing enhanced autonomy. The paper also presents a conceptual framework which provides the foundation for the proposed solution.

Keywords: autonomous agents, dynamic environment, human behaviour, psychology, knowledge abstraction

1 Introduction

The '*behavior oriented*' [1] approach of AI, provides much attention to lifelike qualities of both agents and environments in order to obtain true autonomy and intelligence from agents. It is anticipated that the advances in agent technology will address this need of adaptive behavior in artificial systems.

An agent is a computer system that is placed in an environment, which is capable of autonomous actions in that environment in order to meet its design objectives. Therefore, autonomy refers to figuring out what the agent needs to do by itself in order to satisfy its design objectives. Real time environments are dynamic and continuous; therefore, it is impossible to define all possible scenarios in a discrete way. The internal organization is a priori unknown and "unexpectedness" is inherent to such environments. The definition of autonomous behavior in this paper refers to the ability of the agent to perform without failures when it encounters unexpected changes in the environment.

This paper provides a novel methodology for enhancing agent autonomy through an analogy with biological systems and psychology. The behavioral cycle of a human is captured into a framework and it is proposed as a conceptual framework for building agents. The conceptual framework is facilitated with the workings of episodic memory which is believed to be the main contributor for human autonomy. As it uses the knowledge of abstract concepts to derive a response it can be used as a mechanism to take decisions in an unknown situation based on known facts and past experiences. The new model is demonstrated in a "real estate" scenario to determine the price of a house and the results are compared with the solutions obtained using other intelligent techniques.

The paper is organized as follows: Section 2 contains the proposed methodology with a justification from human behavior and psychology. The demonstration of the method is done in Section 3. Section 4 of the paper contains the discussion and suggestions for future work.

2 Proposed Methodology

The task of an autonomous agent is to figure out the actions according to the encountered state while adapting to environment dynamisms. The theoretical basis for such an environment to action mapping is described in Section 2.1. Humans too perform an environment to action mapping by accumulating experiences. Inspired from human behavior and psychology, a conceptual framework for designing agents is provided in Section 2.2. The concept of episodic memory which provides this ability to humans is elaborated in Section 2.3 with a corresponding mapping between environment and actions.

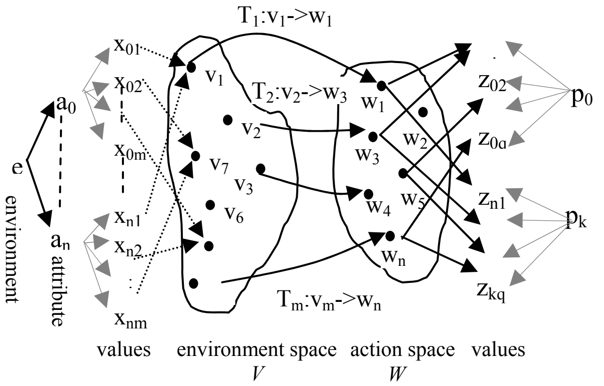

Figure 1: Environment space (V), action space (W) and mapping between V and W

2. 1 Environment and Action Selection

The environment has been defined [2] as a finite set E of discrete, instantaneous states,

$$E = \{e_0, e_1, ..., e_n\} \text{ ----------------------- (eq1)}$$

Each environment state $e \in E$, is a collection of attributes, $A = \{a_0, a_1, ..., a_m\}$ ------------------ (eq2)

Each attribute $a \in A$, consists of values x_{ij}, where $x_{ij} \in X$, which represents a qualitative or quantitative value domain. Every attribute-value (*a-x*) pair is a vector in the *environment space V* of dimension *m* and the environment space is a Cartesian product of *a-x* pairs:

$\{x_{0i}\} \times \{x_{1j}\} \times ...... \times \{x_{nm}\}$

Agents have a finite set of possible actions $\{Ac\}$ available to them, $Ac = \{\alpha_0, \alpha_1, ..., \alpha_q\}$------------(eq3) limited by the number of actuators. Actions consist of parameters $p_i \in P$, where P is a finite set $P = \{p_0, p_1, ..., p_k\}$ -------------------------------- (eq4)

Each parameter $p \in P$, consists of values z_{ij}, where $z_{ij} \in Z$, which represents a qualitative or quantitative value domain. Every parameter-value (*p-z*) pair is a vector in the *action space W* of dimension *k*. Action space is a Cartesian product of *p-z* pairs: $\{z_{0l}\} \times \{z_{1m}\} \times ...... \times \{z_{nk}\}$

Applying the above definitions, the task of the agent when executing in dynamic environment is to identify the correct mapping from *environment space (V)* to the *action space (W)*. We have shown that it is a non linear and non topology preserving transformation between two vector spaces *V*, *W* of the form $T : V -> W$ [3]. *V*, *W* and $T : V -> W$ capture and build the world model for the agent to function autonomously. The environment to action mapping is graphically shown in Figure 1.

2.2 A Conceptual Framework for an Autonomous, Adaptive Agent

The proposed conceptual architecture is composed of four layers: *inheritance* (Ψ_i), *training* (Ψ_t), *experience* (Ψe) and *unexpected* (Ψu) as illustrated in Figure 2. Each layer depicts the functionality of a human being or an animal. Generally the primitive actions taken for survival come as pre wired neurons. This phenomenon is captured by Ψ_i in the conceptual framework. Ψ_i supports the concepts in *bottom-up AI* [4, 5] which emphasis on starting with basic competencies such as 'survival' skills of the sensory-motor coordination type (such as obstacle avoidance) on top of which more complex behavior can be developed through interactions with the environment. Ψ_t is built on top of Ψ_i to narrow the domain with domain specific abstract details. Ψ_i and Ψ_t model the evolutionary and self organizing process of the brain as it is said that evolution of the brain is achieved through genetically defined information and learning [6]. With maturity in a certain domain, humans gain experience on top of training which assists them with adaptive decision making. This is denoted by Ψ_e and it makes the agent more adaptive to the environment. Once the agent has Ψ_i, Ψ_t and Ψ_e, ultimately it will become capable of reacting to the "unexpected" nature of a dynamic environment using the pool of knowledge to obtain the summed up effect. It is mapped to Ψ_u in the conceptual view. With time, this layer can be absorbed into Ψ_e. The four layers in the conceptual framework are an attempt to discover and model the relationships organized in human episodic memory and using it to enhance agent autonomy.

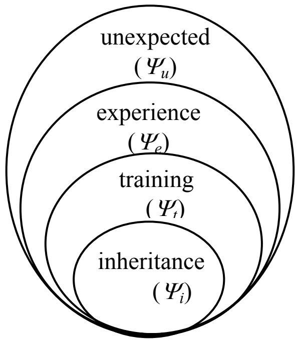

Figure 2: Conceptual framework

2.3 Justification from Human Behavior and Psychology

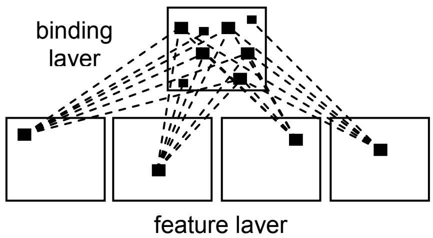

Figure 3: Representation of episodic memory in two layers

According to the proposed conceptual framework in Section 2.2, an evolving architecture is required to implement it as the decision making unit of the agent. With humans, the type of memory that builds a world model with exposure to past events is known as episodic memory [7]. Each rule, knowledge, training and experience is recorded as a separate trace in episodic memory providing a cumulative effect when taking decisions. We believe the autonomy and adaptive nature of humans is mainly due to the organization and retrieval structure inherent in the episodic memory. Therefore, the same concept is used to enhance agent autonomy enabling each event to produce a new memory trace. Consequently it is assumed that the knowledge of abstract concept is used to take a decision for a newly encountered situation [8, 9]. We have adopted such previous knowledge to enhance agent autonomy.

M. Moll and R. Miikkulainen have presented the episodic memory in two layers as illustrated in Figure 3 [10]. The feature layer (*FL*) consists of the experiences a human obtains throughout life. Each feature is demonstrated by a separate feature unit *fu* such that $fu_i \in FL$ for $i = 1 \ldots n$ where *n* is the number of features. The actual knowledge/rule is a combination of feature values, and that is represented in the binding layer (*BL*). For each experience there is a bidirectional connection between the binding layer and feature units $FL \leftrightarrow BL$ that contributes to the particular experience.

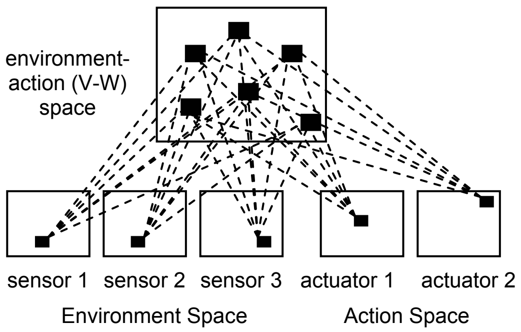

Figure 4: Mapping between Figure 1 and Figure 3

If the action space *W* in Figure 1 is superimposed on environment space *V* the resultant can be shown in Figure 4, using the concept of $FL \leftrightarrow BL$. The formation of Figure 4 comparing Figure 1 and 3 can be defined as follows:

- Each feature unit fu_i is a collection of values y_j where $y_j \in Y$ and *Y* is a finite set. For example if fu_i = *time*, then Y is the finite set of all the values time can take for the applications such as $Y = \{1ns, 0.1s, 10\min, 1hour, ...\}$
- *fu* can be either an input feature unit (fu_{input}) or an output feature unit (fu_{output}) such that $fu_{input} \in fu_i,\ i = 1....n1$, $fu_{output} \in fu_j,\ j = 1....n2$ where $n1$ = total no. of input feature units, $n2$ = total no. of output feature units.
- If $Y1$ is the set with all the input feature values and $Y2$ is the set with all the output values then $Y1 = \{fu_i\}$ *and* $Y2 = \{fuj\}$
  $Y1 \subset Y \text{ and } Y2 \subset Y \text{ and } Y1 \cap Y2 = \phi \text{ and } Y1 \cup Y2 = Y$
- Comparing with eq2 and eq4 of Section 2.3 (environment as represented by sensors and actions as represented by actuators)
  $Y1 \equiv A \text{ and } Y2 \equiv P$
- Comparing with eq1 and eq3 of Section 2.3
  $E = \{e_0, e_1, ..., e_n\} \equiv fu_{input}$
  $Ac = \{\alpha_0, \alpha_1, ..., \alpha_q\} \equiv fu_{output}$
- *BL* consists of mappings among fu_{input} and fu_{output}:
  $fu_{input} \leftrightarrow fu_{output} \in BL$
- Transformation *T* from environment space *V* to action space *W* is
  $T : V -> W \equiv fu_{input} \leftrightarrow fu_{output} \in BL$
- The input-output relationship in *T* is captured by the memory traces, by recording each rule, knowledge, training and experience as distinct traces.

According to the above analysis, the theory behind environment to action mapping is depicted in Figure 4. Therefore, we believe that the organization and action retrieval mechanism of episodic memory can be applied to enhance the autonomy of agents in a dynamic environment.

2.4 Determination of Output Action

The aim of this section is to demonstrate how the organization of episodic memory can be used by the agents to determine the output action autonomously. In doing so, for clarity of representation, each trace in *V-W* space of Figure 4 is illustrated as an array of features as shown in left side of Figure 5. Arrays $M_i, i = 1.....k+2$ depict

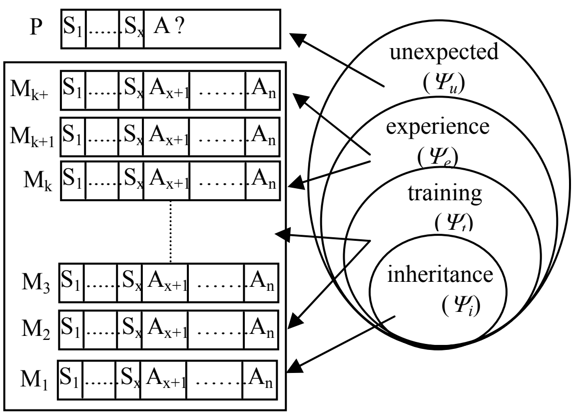

Figure 5: Mapping between conceptual framework and memory traces in episodic memory

$T : V -> W \equiv fu_{input} \leftrightarrow fu_{output} \in BL$ and
$S_j \in fu_{input},\ j = 1.....x \text{ and}$
$A_j \in fu_{output},\ j = x+1.....n$

Also Figure 5 depicts the mapping between conceptual framework and memory traces in episodic memory. According to Figure 5, a new situation *(P)* consists of fu_{input} and the task of the agent is to determine fu_{output} using accumulated knowledge in M_i's. Once a new situation is handled it will become part of experience layer. As the agent accumulates new experiences, the number of memory traces will be increased providing an evolutionary effect resulting from knowledge accumulation.

The work carried out by Hintzman with episodic memory deals with feature identification, if the memory traces are organized as shown in Figure 5 and if features of a memory trace is partially known [9]. We have extended the concept to enhance agent autonomy. The steps involved in the process are as follows:

Step 1: From already organized memory traces $M_1 \ldots M_{K+2}$, calculate the similarity factor of each trace to the new situation *P*. Similarity of trace M_i to *P* is denoted by e_i such that:

$$e_i = \frac{1}{n} \sum_{j=1}^{n} p_j \bullet M_{i,j}$$

where *n* – number of features(environment + action)

$M_{i,j}$ - value of j^{th} feature of trace *i*

p_j – value of the jth feature of new situation *P*

Step 2: Not all the traces with a value for e_i will contribute to the output of *P*. Use M_i with $e_i > th$ (threshold) to derive the action part of *P*: $A_{x+1} \ldots A_n$ such that

$$A_j = \sum_{i=1}^{m} e_i \bullet M_{i,j}$$

Step 3: Identify possible output values
  for trace $M_i = 1 \ldots n$
    if $e_i > th$
      mark A_j's of M_i as possible output values
      $A_j \in \Omega$, where Ω is the set with possible output values

Step 4: Filtering the values in Ω to obtain a single output value for actuators such that: $A_j, j = x+1 \ldots n = filter(\Omega)$

The filtering technique is application specific. The above steps 1-4 with a filtering technique are used in Section 3 to demonstrate the actual use of these in a dynamic environment.

2.5 Comparison of the Proposed Methodology with Existing Intelligent Agents.

Agent autonomy is a major factor considered in all the agent architectures. For example deliberative agents consist of a central reasoning system [11] to facilitate agent decision making or autonomy. They accomplish goals using a set of rules defined as plans and maintain an internal world model that is useful for plan generation. Belief, Desires and Intentions (BDI) model [12, 13] is one of the matured agent architecture for deliberative agents. As described in [14] the autonomous action selection of BDI agents is a process as follows:

$B := B_o$ /* initial beliefs about the world */
while true do
  get next percept p /*take the sensory input*/
  $B := brf(B, p)$
  $I := deliberate(B)$ /*goal to achieve*/
  $\pi := plan(B, I)$
  execute (π)

Deliberation refers to the understanding of available options (*D*) and selection of one option (*I*) as the final goal.

$$D := options(B)$$
$$I := filter(B, D)$$

The beliefs and plans in BDI architecture are static and the task of the agent is to identify the specific rule that matches with the current environment state. In the proposed methodology the beliefs and plans are generated from the environment. If unknown environment state is encountered, proposed method is capable of generating an action by performing Steps 1-4. Each rule, knowledge, experience the agent identifies makes new memory traces providing an evolutionary nature.

3 Demonstration of the Methodology

In this section, the proposed methodology is demonstrated using a real estate scenario. The task is to determine the price of houses considering factors such as number of bed rooms, other rooms, bath rooms, square feet and acres. Currently neural network techniques are used successfully in such applications. But our aim was to verify the feasibility of the proposed model as applied to the decision making unit of an agent for such applications and to highlight the advantages of the new methodology.

The environment is captured by the attributes *bedroom, other rooms, baths, square feet, acres* and the action space consist of a single attribute *price*. The data set is a sample used in NeuroShell 2 by Ward Systems Group Inc. Compared to the conceptual framework, the data set is considered as $\Psi_i + \Psi_t$ layers of the conceptual framework.

3.1 Organization of the Data Set as Episodic Memory

The environment and action attributes were encoded using a set of bit patterns to generate memory traces as previously shown in Figure 5. The bit pattern was defined by considering the minimum and maximum value of each attribute. For example *min(bed rooms)* = 2 and *max(bed rooms)* = 5 in the given data set. Therefore 4 bits were used to record the number of beds: 1000 – 2 beds, 0100 – 3 beds and so on. This organization of data set as memory traces is shown in Figure 6(a).

Table 1: Displacement of P with traces in Ω

trace Ω	Price '000	displacement for					total displ
		bed rooms	other rooms	baths	squ. feet	ac	
Ω_1	200	1	-1	0	-1	-6	-7
Ω_2	200	1	0	0	0	-4	-3
Ω_3	205	0	0	0	-4	-1	-5
Ω_4	125	1	0	2	1	-1	3
Ω_5	**134.9**	**0**	**0**	**1**	**0**	**-1**	**0**
Ω_6	133	1	1	2	1	-1	4
Ω_7	130	1	0	2	1	1	5

3.2 Obtaining the output for new/unknown situations

For the given scenario, a new/unknown situation (*P*) consists with known values for environment attributes and price has to be determined as shown in Figure 6(a). A memory trace that purely represents *P* may not be present in the memory traces that consist of $\Psi_i + \Psi_t$. In this situation price of *P* has to be determined using the traces that partially match with *P*. Steps 1-4 described in Section 2.4 were carried out as below to determine a price for *P*.

Step 1: Calculation of e_i

Each environment attribute of P is matched against the memory traces as shown in Figure 6(a). If at least a single attribute is matched, that memory trace was considered as a contributing trace.

Step 2: Identification of a threshold - *th*

From the selected memory traces, for each attribute a histogram was generated to determine a threshold *th* as shown in Figure 6(b).

Step 3: Identification of M_i's with $e_i > th$

Once *th* is determined, a re-projection to memory traces was carried out as shown in Figure 6(c) to determine the best possible set of options Ω (in BDI terminology the desires D).

The first data item in training set was partially changed as (4, 4, 3, 2000, 0.4, ?) and considered as *P*. Intuitively, we feel that the answer would be a value lower than 150,000. But it has been mapped to several other traces by partially matching the attribute values.

Step 4: Filtering technique to identify the final output

As discussed in Section 2.4 filtering technique is application specific. For this application, the displacement of traces in Ω to *P* is considered as the filter. Figure 6(d) demonstrates how the displacement filter was applied to traces in Ω. Table 1 lists the actual displacement of *P* with traces of Ω. The best applicable trace, which has the least displacement, is highlighted and it corresponds to a price of 134,900. Therefore according to this methodology, the price for *P* is 134,900.

Similarly, the methodology was applied to new set of data items (*P*'s) to determine the value of price which is considered as the unknown attribute. As these data items are not present in memory traces that comprise of $\Psi_i + \Psi_t$ and are considered as Ψ_u. The corresponding price value obtained using the proposed methodology for each data item is shown in Table 2 with a comparison value obtained using back propagation neural network technique (BPNN) (refer Section 3.3).

Table 2: Value of price for new data items obtained using proposed methodology and BPNN

data item (*P*)	expected price '000	price '000	
		proposed methodology	BPNN
(4,4,3,2000,0.4,?)	<150	134.9	168.407
(2,3,2,800,0.25,?)	>85	95	95.211
(3,5,4,3100,2.6,?)	around 269	254.9	260.609
(4,4,2,1500,0.25,?)	around 134.9	134.9	141.697
(3,3,2,1500,0.5,?)	<144.5	134.9	106.076

3.3 Comparison of Result with Neural Network techniques

A back propagation neural network (BPNN) approach was used to compare the results. A BPNN with a single hidden layer was trained with the same data set which was used to generate the memory traces of $\Psi_i + \Psi_t$ and used the test scenarios (*P*'s) discussed in Section 3.2. The configuration of the BPNN which produced the outputs shown in Table 2 is as follows: learning rate = 0.1, momentum = 0.1, input layer neurons = 5, hidden layer neurons = 5, output layer neurons = 1.

3.4 Analysis of the Result

The price of *P* obtained using the proposed methodology is quite satisfactory compared to the price obtained using BPNN. With further training and with change of parameters BPNN can produce much accurate result. But how much training is required is a priori unknown and it is highly sensitive to the application. When the agent is operating in dynamic environment, it has to produce an action within certain time period. Therefore, we believe the proposed methodology has a higher potential in obtaining autonomous decisions in a dynamic environment.

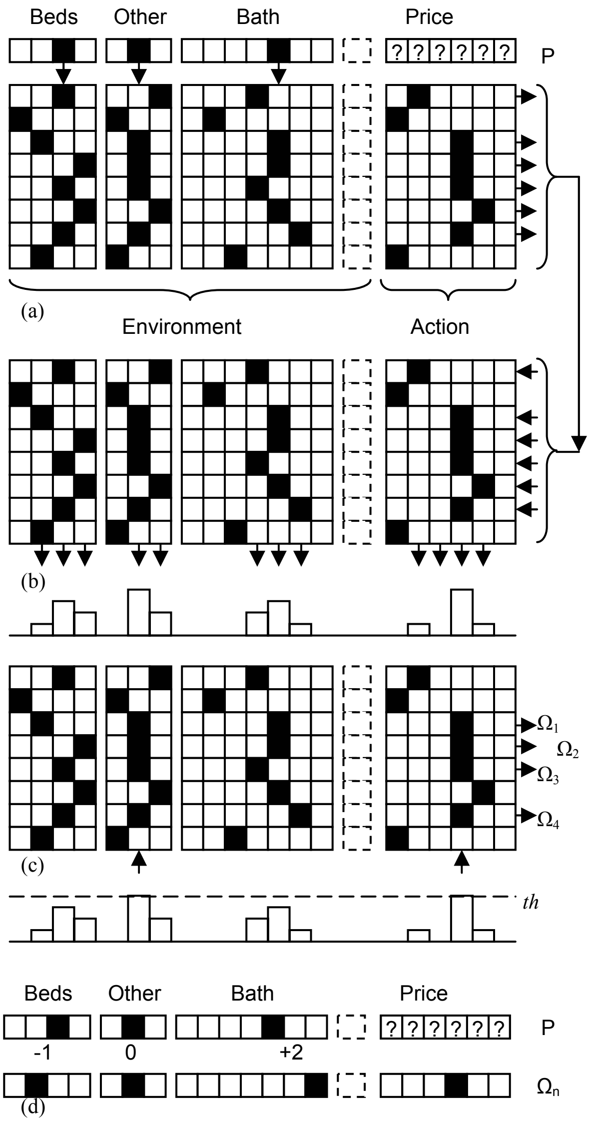

Figure 6: Process of deriving an output using Steps 1-4 in Section 2.4 using memory traces (Refer to text for details)

4 Discussion and Future Work

Through out the paper it is argued that for a dynamic environment, identifying all the environment states in advance is not feasible and the success of static action/plans is not guaranteed. Instead, as the organization of episodic memory provides humans with unique features, similar concept is used in this paper to enhance agent autonomy. The conceptual framework proposed in Section 2.2 provides the foundation for applying this technique.

The main advantage of the proposed methodology compared to other intelligent techniques such as neural network is that it does not require training. Acquiring a new experience means incorporating a new memory trace to the existing traces, exhibiting an evolving nature. Contribution to the enhancement of autonomy can occur from this stage. Also with neural networks, large number of parameters such as learning rate, momentum, number of hidden layers and number of neurons in each hidden layer have to be computed and adjusted. The values of all the parameters are highly data and application sensitive. But with the proposed method threshold *th* and a filtering technique have to be determined. In addition, if memory traces can be obtained from different application domains, the output action for an unknown state can be determined using cross domain knowledge.

Application of the proposed methodology to diverse agent applications will be our main future work. Also the cross domain effect needs further investigation.

References

1. Steels, L., "The Artificial Life Roots of Artificial Intelligence", *Artificial Life*, 1, pp. 75-100 (1994).
2. Wooldridge, M., *An Introduction to MultiAgent Systems*, John Wiley & Sons Ltd (2002).
3. Wickramasinghe, L.K., Alahakoon, L.D., and Smith, K.A., "Improving the Self Adaptive Behavior of Agents by Reverse Engineering Biological Systems", *Submitted to Journal of Artificial Intelligence Research* (2004).
4. Brooks, R.A., "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, 2(1), pp. 14-23 (1986).
5. Maes, P., "Behavior-Based Artificial Intelligence", *Proceedings of Fifteenth Annual Meeting of the Cognitive Science Society*, pp. 74-83 (1993).
6. Kasabov, N., "A framework for evolving connectionist systems and the "eco" training method", *Proceedings of ICONIP'98 - The Fifth International Conference on Neural Information Processing*, Japan, IOS Press, pp. 1232-1235 (1998).
7. Tulving, E., *Elements of Episodic Memory*,Oxford University Press (1983).
8. Hintzman, D.L., "MINERVA 2: A simulation model of human memory", *Behavior Research Methods, Instruments & Computers*, 16(2), pp. 96-101 (1984).
9. Hintzman, D.L., "Schema Abstraction in a Multiple-Trace Memory Model", *Psychological Review*, 93(4), pp. 411-428 (1986).
10. Moll, M. and Miikkulainen, R., "Convergence-Zone Episodic Memory: Analysis And Simulations", *Neural Networks*, 10, pp. 1017-1036 (1997).
11. Ginsberg, M., "Universal planning: An (almost) universally bad idea", *AI Magazine*, 10(4), pp. 40-44 (1989).
12. Rao, A.S. and Georgeff, M.P., "BDI agents: From theory to practice", *Proceedings of Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 312-319 (1995).
13. Haddadi, A. and Sundermeyer, K., *Belief desire intention agent architectures*, in *Foundations of Distributed Artificial Intelligence*, G.M.P. O'Hare and N.R. Jennings, Editors, Wiley, New York (1996).
14. Wooldridge, M., *Practical Reasoning Agents*, in *An Introduction to MultiAgent Systems*, John Wiley & Sons, Ltd, pp. 65-82 (2002).