A Self-Organizing Multi-Memory System for Autonomous Agents

Wenwen Wang, Budhitama Subagdja, Ah-Hwee Tan
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
Email: wa0003en, budhitama, asahtan@ntu.edu.sg

Yuan-Sin Tan
DSO National Laboratories
20 Science Park Drive, Singapore 118230
Email: tyuansin@dso.org.sg

***Abstract*—This paper presents a self-organizing approach to the learning of procedural and declarative knowledge in parallel using independent but interconnected memory models. The proposed system, employing fusion Adaptive Resonance Theory (fusion ART) network as a building block, consists of a declarative memory module, that learns both episodic traces and semantic knowledge in real time, as well as a procedural memory module that learns reactive responses to its environment through reinforcement learning. More importantly, the proposed multi-memory system demonstrates how the various memory modules transfer knowledge and cooperate with each other for a higher overall performance. We present experimental studies, wherein the proposed system is tasked to learn the procedural and declarative knowledge for an autonomous agent playing in a first person game environment called Unreal Tournament. Our experimental results show that the multi-memory system is able to enhance the performance of the agent in a real time environment by utilizing both its procedural and declarative knowledge.**

***Index Terms*—self-organizing, ART, agent, episodic memory, semantic memory, procedural memory, Unreal Tournament.**

I. Introduction

Memory forms the basis of human's knowledge and behaviors. Two types of long term memories facilitate our brain to react upon current situations based on past experiences: explicit (declarative) memory and implicit (non-declarative or procedural) memory. As their names suggest, explicit memory can be easily expressed using words, whereas implicit memory cannot be easily expressed. Within declarative memory, concepts and relations are remembered explicitly, while in non-declarative memory, information is processed without using explicit knowledge. Declarative memory typically can be further categorized into two types: episodic memory enables one to remember personal experiences that can be explicitly stated while semantic memory stores meanings, concepts, rules, and general facts unrelated to specific experiences [19].

Evidences from cognitive neuroscience imply that both declarative and procedural memory are crucial parts for supporting many different types of cognitive functions. Procedural memory executes complex procedures in the absence of consciousness. Through intensively rehearsal on complex action sequences, procedural memory guides the association of all relevant cognitive modules to accomplish various tasks, which serves a critical role for the further development of cognitive skills. On the other hand, while semantic memory represents high level concepts and knowledge which forms the basis of our understanding, recent research has found episodic memory to be crucial in supporting many cognitive capabilities, including concept formation, spatio-temporal learning and goal processing [3].

Although prior research showed that the different components of our long term memory systems interact with one another to serve their roles and functionalities, most existing computational studies on memory learning still model them as isolated memory systems. Even those models, which focus on understanding the relations among memories, are usually limited to studying the interactions between just two memory systems (e.g. between semantic and episodic memory). In this paper, we propose a biologically inspired multi-memory system for modeling the structures and connections between the procedural and declarative memories, based on a unified set of computational principles and algorithms under fusion Adaptive Resonance Theory (fusion ART) [16]. Using fusion ART as a building block, our proposed multi-memory system includes a procedural memory model that learns decision rules through reinforcement learning, an episodic memory model that encodes an individual's experience in the form of events and spatio-temporal relations among events, and a semantic memory that captures factual knowledge from past experiences. The system further incorporates a generic process of memory consolidation, wherein the specific information stored in the episodic memory can be transferred to produce more general and abstract semantic knowledge. In this way, the proposed system supports not only the distributed and redundant knowledge representations across multiple memory models, but also independent and parallel processing in different time scales, achieving robust memory learning and preventing catastrophic forgetting.

We have conducted experimental studies, wherein the proposed multi-memory model is used to learn procedural and declarative memory of an agent playing in a first person shooting game called Unreal Tournament. Our experiment results show how the proposed memory system improves the learning capability of the agent in real time while managing and utilizing various types of procedural and declarative knowledge.

The rest of this paper is organized as follows. Section II provides a brief discussion of related works on procedural and declarative memory models. Section III presents the architecture of our proposed multi-memory system, including the modeling of episodic, semantic and procedural memory, as well as their interactions. Section IV shows the experimental evaluation of the proposed system on a shooting game domain. The final section concludes and highlights future work.

II. Related Work

In literature, many models have been proposed to represent and learn each individual type of memory. Several works develop computational models of episodic memory as a trace of events and activities stored in a linear order wherein some operations are designed specifically to retrieve and modify the memory to support particular tasks [9], [20]. These approaches are limited to learning complex relations between events and retrieving episodes with imperfect or noisy cues. Some works use neural networks to model episodic memory with inherent supports for partial matching and pattern generalization. Shastri focuses on complex relational representation in SMRITI [13]. The model can handle role-entity bindings in which retrieval cues can involve transient values for retrieval using partial information, while omitting the temporal or sequential relations between items altogether. On the other hand, TESMECOR [12] employs sparsely distributed neural network approach to handle spatio-temporal or multimodal patterns, which rapidly stores distributed patterns while providing a robust retrieval with complex sequential representation. However, it still retains the sequences of events, rather than chunks of episodes.

Early semantic memory modelings provide various abstract computational models ranging from statistical, associative to neural network models. While most associative models (e.g. [1]) and network models (e.g. [6]) focus on representation but not on learning of such semantic knowledge; typical statistical models (e.g. [7], [9]) allow learning but only work for a limited form of semantic memory. Some others are based on neural architectures corresponding to the memory systems in the brain. Hinton [5] emulates the semantic network model by setting up interconnected neural fields reflecting different elements of a proposition. Beyond representing relationships between concepts, the connectionist architecture supports recollection and generalization through pattern completion across the network. Farah and McClelland [4] suggest a bidirectional network model consisting of different interconnected neural fields corresponding to their sensory-functional features.

Besides studying individual memory as an isolated system, some existing works attempt to understand the interactions among declarative memories. REM-II [9] connects episodic memory and semantic memory together to learn statistical relationships between items within and across time. Another episodic memory model based on the SOAR [10] embeds episodic memory directly to the symbolic semantic memory model as additional properties providing contextual and historical information of each assertion and update in the memory. A more realistic interaction model called Complementary Learning Systems (CLS) [8] reflects connections between hippocampus and neocortex (brain areas commonly known as associated with episodic and semantic memory respectively) in the brain and comprises a particular memory consolidation process.

The typical connectionist models of procedural memory (e.g. [2]) learn cognitive and motor skills as associated pairs or temporal sequences of actions. On the other hand, fragment-based or chunking models (e.g. [11]) acquire procedural knowledge through case-based learning of memory chunks or fragments. More recently, the hybrid models of the two procedural learning paradigms further investigate the cognitive capabilities and functionalities emerging from the interaction between procedural and declarative learning. In these models, the sub-symbolic learning from procedural memory (e.g. [14]) takes the controls of various tasks, while the symbolic and explicit rules are inferred through the declarative memory.

III. The Multiple-Memory Model

The proposed procedural-declarative memory system is considered to be an integral part of the reasoning mechanism of an agent (Figure 1). At each point of time, the agent interacts with the environment and performs certain tasks based on its current level of procedural skills and declarative knowledge. The perception and information characterizing a single experience from the agent reasoning system can be captured as a snapshot and then encoded to be an individual event. The encoded event is held temporarily in a shared working memory in which the episodic memory automatically stores and organizes events in subsequent order into units of episode. The events and episodes stored in episodic memory trigger the learning of semantic memory through a consolidation process, which further leads to the formation of new knowledge and behavior patterns.

In general, the three proposed memory modules learn and run concurrently but in different paces. Their major memory operations to facilitate the reasoning process can be listed as follows:

- **Action selection by procedural memory**. At any point in time, the current action to take is selected through the procedural memory by executing the fired procedural rule.
- **Automatic encoding in episodic memory**. Events held in working memory are automatically captured and stored in episodic memory.
- **Memory consolidation by pattern reinstatements**. At some point in time, the contents of episodic memory are read out to the working memory which starts the learning process in semantic memory.
- **Declarative memory retrieval by pattern completion**. The information and knowledge in the episodic and semantic memory can be retrieved by providing memory cues as a subset or portions of the target information to be retrieved.

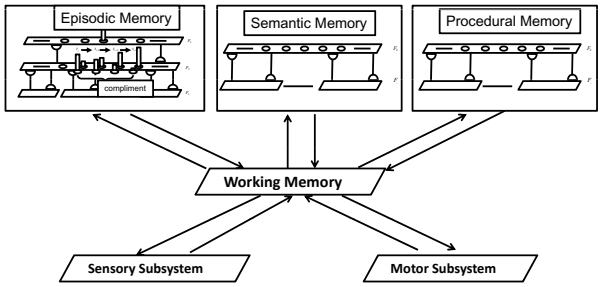

Fig. 1. The multi-memory system

- **Behavior learning in procedural memory**. The association between behavior and current state of the agent can be learned by procedural memory through reinforcement learning across the sensory, motor, and feedback channels.

The proposed memory model is based on fusion Adaptive Resonance Theory (ART) [16] which performs unsupervised learning input patterns in real time. Given a set of input patterns , one for each input channel, fusion ART employs a bi-directional process of recognition and prediction to find the best matching category. It also learns continuously by updating the weights of neural connections at the end of each search cycle. In addition, the fusion ART model may grow dynamically by allocating a new category node if no match can be found. This type of neural network is chosen as the building block of our memory system as it enables continuous formation of memory with adjustable vigilance to control the growth of the network and the level of generalization. By applying fuzzy operations and *complement coding* [16], fusion ART can generalize input patterns dynamically and capture a range of values every time it learns.

A. Fusion ART

Fusion ART network is used to learn the individual memory modules in a unified manner. In this case, each memory trace stored is represented as a multi-channel pattern. Figure 2 illustrates the fusion ART architecture, which may be viewed as an ART network with multiple input fields.

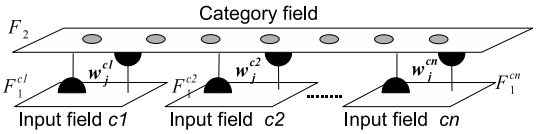

Fig. 2. Fusion ART

The detailed dynamics of a multi-channel fusion ART can be described as follows.

Input vectors: Let $\mathbf{I}^k = (I_1^k, I_2^k, \ldots, I_n^k)$ denote an input vector, where $I_i^k \in [0,1]$ indicates the input i to channel k, for $k = 1, \ldots, n$. With complement coding, the input vector $\mathbf{I}^k$ is augmented with a complement vector $\bar{\mathbf{I}}^k$ such that $\bar{I}_i^k = 1 - I_i^k$.

Input fields: Let F_1^k denote an input field that holds the input pattern for channel k. Let $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_n^k)$ be the activity vector of F_1^k receiving the input vector $\mathbf{I}^k$ (including the complement).
Category fields: Let F_i denote a category field and $i > 1$ indicate that it is the ith field. The standard multi-channel ART has only one category field which is F_2. Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ be the activity vector of F_2.
Weight vectors: Let $\mathbf{w}_j^k$ denote the weight vector associated with the jth node in F_2 for learning the input pattern in F_1^k.
Parameters: Each field's dynamics is determined by choice parameters $\alpha^k \geq 0$, learning rate parameters $\beta^k \in [0,1]$, contribution parameters $\gamma^k \in [0,1]$ and vigilance parameters $\rho^k \in [0,1]$.

The dynamics of a multi-channel ART can be considered as a system of continuous resonance search processes comprising the basic operations as follows.
Code activation: Given an activity vector, $\mathbf{x}^k$, a node j in F_2 is activated by the choice function

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.
Code competition: A code competition process follows to select a F_2 node with the highest choice function value. The winner is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \tag{2}$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$ indicating a *winner-take-all* strategy.
Template matching: A template matching process checks if resonance occurs. Specifically, for each channel k, it checks the *match function* m_J^k of the chosen node J meets its vigilance criterion such that

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \tag{3}$$

If any of the vigilance constraints is violated, mismatch reset occurs or T_J is set to 0 for the duration of the input presentation. Another F_2 node J is selected using choice function and code competition until a resonance is achieved. If no selected node in F_2 meets the vigilance, an uncommitted node is recruited in F_2 as a new category node selected by default.
Template learning: Once a resonance occurs, for each channel k, the weight vector $\mathbf{w}_J^k$ is modified by the following learning rule:

$$\mathbf{w}_J^{k(\text{new})} = (1 - \beta^k)\mathbf{w}_J^{k(\text{old})} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})}). \tag{4}$$

Activity readout: The chosen F_2 node J may perform a readout of its weight vectors to an input field F_1^k such that $\mathbf{x}^{k(\text{new})} = \mathbf{w}_J^k$.

A fusion ART network, consisting of different input (output) fields and a category field, is a flexible architecture that can be made for a wide variety of purposes. The neural network can learn and categorize inputs and can be made to map a category to some predefined fields by a readout process to produce the output. Another important feature of the fusion ART network regarding its use in memory is that no separate phase of operation is necessary for conducting recognition (activation) and learning. Learning can be conducted by adjusting the weighted connections while the network searches and selects the best matching node. When no existing node can be matched, a new uncommitted node is allocated to represent the new pattern. Hence, the network can grow in response to novel patterns.

B. Episodic Memory Model

The two key elements of episodic memory are events and episodes. An event can be represented as an aggregation of attributes describing a snapshot of experience in time. The event attribute values characterize the what (e.g subject, relation, action, object), where (e.g location, country, place), and when (e.g date, time, day, night) information of an event. Figure 3 shows an example of the structure of an event based on Unreal Tournament video game domain [21] used in our experiment (explained in later sections). An event may consist of information about location, states (health level, ammo level, distance to enemy, and reachable items around), behaviors (running around, collecting items, escaping from the enemy, and engaging in battle), and a reward (or punishment) level (e.g kill the enemy, being killed or damaged). These information can be represented as a vector which later can be processed as an input for episodic memory.

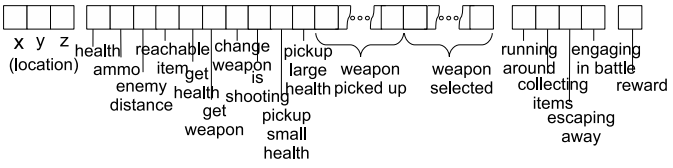

Fig. 3. Event encoding

An event can be encoded as an input vector to the fusion ART network and a category can be selected as an activated node by a bottom-up activation process. On the other hand, the top-down activation (readout operation) achieves the recall task. Figure 4(a) illustrates the bottom up and top down operations for learning, recognition, and recalling an event. The figure shows that to store and recall an event, two layers of neural fields F_1 and F_2 are involved.

On the other hand, an episode can be defined as a list of events collected in a temporal order. Our approach to encode a sequential order of events in the neural network is by maintaining a graded pattern of neural activations as each activation decays over time. Every time a new activation occurs in a neural node (neuron) j, the value is decayed so that $y_j^{(new)} = y_j^{(old)}(1-\tau)$ where $\tau \in (0,1)$ is the decaying factor and y_j is the activation value of node j. The activation values form a certain pattern such that $y_{t_i} > y_{t_{i-1}} > y_{t_{i-2}} > ... > y_{t_{i-n}}$ holds where $t_0, t_1, t_2, ..., t_n$ denote time points and y_{t_j} is a node value activated or selected at time t_j. In this case t_i is the current or the latest time point. The graded sequential pattern can also be learnt more permanently as weighted connections for example using the bi-directional activation and learning in fusion ART. Figure 4(b) illustrates the bottom up and top down operations between neural fields F_2 and F_3 for learning, recognition, and recalling an episode.

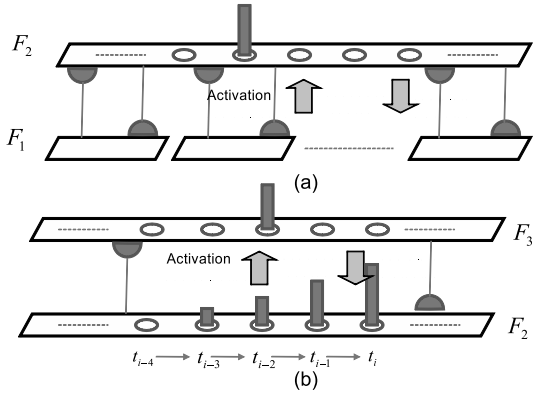

Fig. 4. (a) Operations between F_1 and F_2; (b) Operations between F_2 and F_3

Figure 5 shows the overall structure of the episodic memory model comprising three layers of memory fields. The network between F_1 and F_2 can be considered as a fusion ART for learning individual events. F_2 serves as a medium-term memory buffer for event activations that holds the graded pattern of activations for representing a sequence. The sequential activity pattern can be learnt permanently as weighted connections between F_2 and F_3, and hence they form another fusion ART for learning episodes.

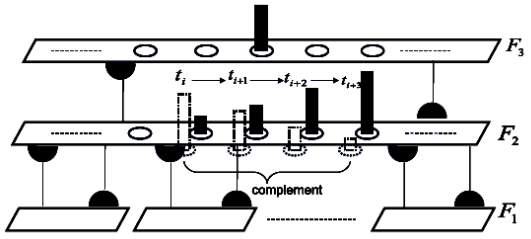

Fig. 5. The episodic memory model

To retrieve an episode, a continuous search process takes place in episodic memory in which the pattern of cue is formed in F_2 while F_3 nodes are activated and selected at the same time by the resonance search process. As long as a matching node is not found (still less than the vigilance of F_2), an F_2 node is activated for every event received (or may not be active at all if the event is absent) while all other nodes are decayed so that $y_j^{(new)} = y_j^{(old)}(1-\tau)$ where y_j is a node j in F_2 field. The continuous retrieval model described above has

been demonstrated in [22] to robustly retrieve episodic traces using different partial and noisy memory cues. For example, given a sequence A,B,C,D,E as a memory trace among other sequence entries in the episodic memory, the sequence can be correctly retrieved using cues of 'A,B,C', 'A,B', or 'D,E' as a target episode presented partially and sequentially. Figure 6 illustrates how memory cues can be used to retrieve a complete episode. Based on the presentation of a memory cue as a sequence (Figure 6i-iii), the neural network finds the best matching node in the category field F_3.

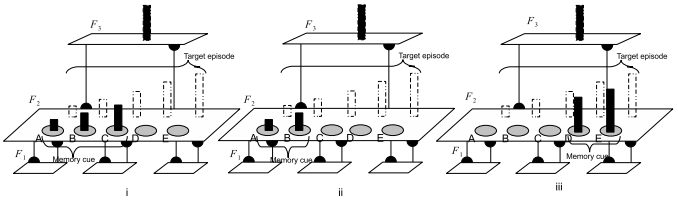

Fig. 6. Memory cues as partial target episodes (complement coding in F_2 is not shown for simplicity)

C. Semantic Memory Model

Different from episodic memory, we view that the semantic memory is not unitary. In other words, there may be different types of semantic memory network, each represents different structure of knowledge. In contrast to episodic memory, each entry in semantic memory generalizes similar inputs into the same category rather than as separate entries. Each type of semantic memory can be made as a fusion ART with each input field representing a property or an attribute of a concept. The generalization can be achieved by lowering the vigilance parameter ρ so that slightly different input patterns will still activate the same category.

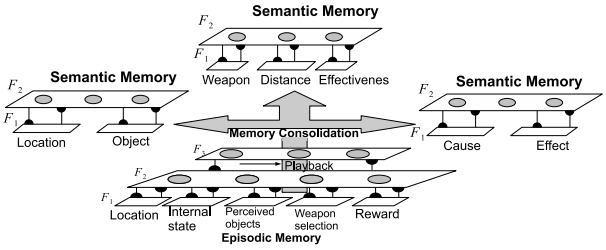

Fig. 7. Different types of semantic memory and the memory consolidation process

Figure 7 illustrates the structure of various types of the semantic memory. A semantic memory network may consist of domain specific associative rules (e.g a set of association between a certain object and its location in the environment, a set of rule associating the effectiveness of a certain weapon and the distance to the opponent) or generic causal relations associating a particular type of event to another that follows. These types of semantic knowledge can be derived by exposing the played back items from the episodic memory to the input of the semantic memory using a lower vigilance parameter and a smaller learning rate such that similar instances may gradually be clustered together regardless of their order.

D. Memory Consolidation From Episodic to Semantic Memory

As two integral components of declarative memory system, episodic memory and semantic memory have been commonly recognized to be intrinsically related [18]: while semantic memory can be considered to be high level concepts and knowledge extracted from specific experiences stored in episodic memory, semantic memory influences daily activities and guides the formation of new episodic memory. Essentially, in this proposed memory system, episodic and semantic memory run and learn independently in parallel but at different paces. Episodic memory serves as a long-term temporary buffer for rapidly storing events and episodes. The stored events and episodes then can be recalled at a later time through a memory consolidation process to gradually extract and learn general facts and rules as semantic memory.

In our proposed system, the knowledge transfer process from episodic to semantic memory starts with a playing-back of the stored episodes from episodic memory model. During each episode reproduction, each associated event in the episode will be presented as both the episodic memory output (via F_1) and the activation pattern of the working memory. The ordering of event reproductions should be consistent with the temporal information stored (via weights of F_2). As each event is presented, it will be reevaluated and checked against its relevance with the current knowledge transfer process. In case the presented event describes the experience of interest, the event representation (shown in Figure 3) held in the working memory is forwarded to semantic memory as a training sample for learning the specific semantic knowledge. Otherwise, the content of working memory is discarded and the reproduction is continued from the next stored event.

E. Procedural Memory Model

The procedural memory model is based on a 3-channel Fusion ART model, also known as Temporal Difference-Fusion Architecture for Learning, COgnition, and Navigation (i.e. TD-FALCON). FALCON learns action and value policies through reinforcement learning across the sensory, motor, and feedback channels. As shown in Figure 8, FALCON [17] comprises a cognitive field F_2^c and three input fields, namely a sensory field F_1^{c1} for representing current states, an action field F_1^{c2} for representing actions, and a reward field F_1^{c3} for representing reinforcement values.

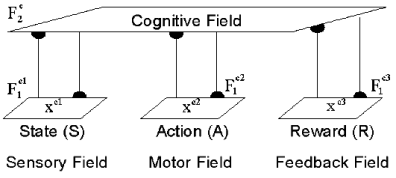

Fig. 8. The procedural memory model

TD-FALCON [15] incorporates Temporal Difference (TD) methods to estimate and learn value functions of action-state pairs $Q(s,a)$ that indicates the goodness for a learning system to take a certain action a in a given state s. Such value functions are then used in the action selection mechanism, also known as the *policy*, to select an action with the maximal payoff.

Given the current state s, the proposed model first decides between exploration and exploitation by following an action selection policy. For exploration, a random action is picked. For exploitation, the proposed model searches for optimal action through a direct code access procedure [15]. Upon receiving a feedback from the environment after performing the action, a TD formula is used to compute a new estimate of the Q value of performing the chosen action in the current state. The new Q value is then used as the teaching signal for the procedural memory model to learn the association of the current state and the chosen action to the estimated Q value. The details of the action selection policy, the direct code access procedure, and the Temporal Difference equation are elaborated as follow.

Action Selection Policy: Through a direct code access procedure, TD-FALCON searches for the cognitive node which matches with the current state and has the maximal reward value. For direct code access, the activity vectors x^{c1}, x^{c2}, and x^{c3} are initialized by $x^{c1} = S$, $x^{c2} = (1, ..., 1)$, and $x^{c3} = (1, 0)$. TD-FALCON then performs code activation and code competition according to equations (5) and (6) to select a cognitive node.

Upon selecting a winning F_2^c node J, the chosen node J performs a readout of its weight vector to the action field F_1^{c2} such that

$$\mathbf{x}^{c2(\text{new})} = \mathbf{x}^{c2(\text{old})} \wedge \mathbf{w}_J^{c2}. \tag{5}$$

An action a_I is then chosen, which has the highest activation value

$$x_I^{c2} = \max\{x_i^{c2(\text{new})} : \text{for all } F_i^{c2} \text{ node } i\}. \tag{6}$$

Learning Value Function: A typical Temporal Difference equation for iterative estimation of value functions $Q(s,a)$ is given by

$$\Delta Q(s,a) = \alpha TD_{err}. \tag{7}$$

where $\alpha \in [0,1]$ is the learning parameter and TD_{err} is a function of the current Q-value predicted by TD-FALCON and the Q-value newly computed by the TD formula.

TD-FALCON employs a Bounded Q-learning rule, wherein the temporal error term is computed by

$$\Delta Q(s,a) = \alpha TD_{err}(1 - \Delta Q(s,a)). \tag{8}$$

where $TD_{err} = r + \gamma max_{a'} Q(s', a') - Q(s,a)$, of which r is the immediate reward value, $\gamma \in [0,1]$ is the discount parameter, and $max_{a'} Q(s', a')$ denotes the maximum estimated value of the next state s'.

F. Decision Making Through Procedural-Declarative Memory

As shown in Figure 1, all memory modules are connected through a medium term working memory which holds all possible details of the current perception and internal status. The decision making of a memory-enhanced agent is conducted through interactions among all existing memory modules. At any time of point, by feeding memory patterns of working memory as input cues, each memory module retrieves and outputs its best matching memory pattern. These activated memory patterns represent the most relevant knowledge stored in different memory modules based on the current situation, combining together to form the current decision making of the agent. Take an example on a shooting game, given the observed situation at some time point (shared by working memory), the current decision making of the agent may consist of an "engaging in battle" action selected from procedural memory and the best weapon choice based on semantic memory. Additionally, during the online learning of different memory modules, the knowledge learned and activated from declarative memory influences the agent's behaviors, which will eventually link to the discovery of new procedural and declarative knowledge.

IV. Case Study On Unreal Tournament

In order to evaluate the integrated procedural-declarative memory model, we embed the proposed memory system into an autonomous non-player character (NPC) agent playing the Unreal Tournament (UT) game. The experiments using UT are conducted to see if the proposed memory model can produce useful knowledge for the agent that improves its performance. The scenario of the game used in the experiment is "Death match", wherein the objective of each agent is to kill as many opponents as possible and to avoid being killed by others. In the game, two (or more) NPCs are running around and shooting each other. They can collect objects in the environment, like health or medical kit to increase its strength and different types of weapon and ammunition for shooting.

In the experiment, all agents that we evaluate in the experiment play against a baseline NPC agent called *AdvanceBot* that behave according to a set of hardcoded rules. There are four different hard-coded behavior modes in *AdvanceBot* (i.e. running around, collecting items, escaping away and engaging in battle). *AdvanceBot* always chooses one of the four behaviors based on a set of predefined rules. Under the battle engagement behavior, the agent also always tries to select the best weapon available for shooting based on some heuristics optimized for a certain environment map used in the game.

A. Memory Enhanced Agents

To investigate how the individual memory modules contribute to the overall agent performance, different agents with different memory modules embedded are tested and compared. This experiment employs two memory-based agents, namely

TABLE I
SAMPLE RULES LEARNT IN PROCEDURAL MEMORY

```
IF health is around 19, and not being damaged,
   and not seen enemy,
   and has adequate ammo,
   and currently in running around state;
  THEN go into collecting items state;
  WITH reward of 0.556.

IF health is 0.4, and being damaged,
   and opponent is in sight,
   and has adequate ammo,
   and currently in collecting item state;
  THEN go into engaging in battle state;
  WITH reward of 0.718.
```

TABLE II
SAMPLE RULES LEARNT IN SEMANTIC MEMORY

```
IF distance is not so far [1800 2099]
  THEN ASSAULT_RIFLE effectiveness 0.07

IF distance is very near [300 599]
  THEN ASSAULT_RIFLE effectiveness 0.048

IF distance is extremely near [0 299]
  THEN SHOCK_RIFLE effectiveness 0.946

IF distance is very near [300 599]
  THEN ROCKET_LAUNCHER effectiveness 0.932
```

Note: the largest visible distance to enemy is 300

RLBot with procedural memory embedded and *MemBot* incorporating the full integrated procedural-declarative memory system.

a) Agents with Procedural Memory: The agent embedding procedural memory module (i.e. *RLBot*) is made by employing the same set of behaviors as *AdvanceBot*. The agent learns and performs its behavior selection through the reinforcement learning algorithm as stated in Section III-E. The state, action, and reward vectors in Figure 9 correspond to the input fields in a multi-channel ART network of *RLBot*. Behavior pattern (i.e. running around, collecting items, escaping away and engaging in battle) in the state vector represents the behavior currently selected. The action vector indicates the next behavior to be selected. Based on the state field and the reward (set to the maximum), the network searches the best match category node and reads out the output to the action field indicating the behavior type to be selected. The network then receives feedbacks in terms of the new state and any reward given by the environment.

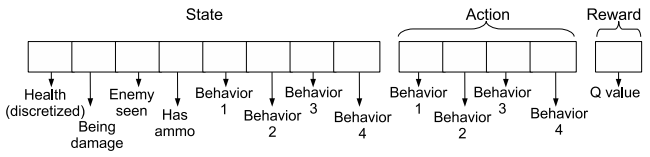

Fig. 9. State, action and reward representation for procedural memory model

The network learns by updating the weighted connections according to the feedback received and applying temporal difference methods as described by III-E to update the reward field. The agent receives the reward signal (positive or negative) whenever it kills or is killed by another agent. In this way, *RLBot* continually learns and acquires procedural knowledge on behavior selections (as illustrated by the sample rule shown in Table I) while playing the games. In contrast to *AdvanceBot*, *RLBot* chooses an available weapon randomly in the battle engagement behavior. Another agent called *RLBot++* is also used to employ the same reinforcement learning model as *RLBot* but select the weapon based on the optimized predefined rules just like in *AdvanceBot*.

b) Agent with Procedural-Declarative Memories: The proposed declarative model is embedded in an agent (i.e. *MemBot*) which has the same architecture as *RLBot* but with the episodic and semantic memories running concurrently. The episodic memory captures episodes based on the event information in the working memory. An event from the UT game is encoded as a vector shown in Figure 3. There are four input fields in episodic memory for location, state, selected behavior, and the reward received. In the experiment, the vigilance of all input fields (ρ_e) and the F_2 field (ρ_s) are set to 1.0 and 0.9 respectively so that it tends to always store distinct events and episodes in response to the incoming events.

As described in Section III-C, the semantic network is applied to learn weapon effectiveness in the experiment. The network has three input fields: the Weapon field representing the identity of the weapon (F_1^a); the Distance field representing the distance between the agent and its opponent at the time of shooting (F_1^b); and the Effectiveness field representing the chance to kill the enemy (F_1^c). In the experiment, the vigilance of the Weapon (ρ^a), Distance (ρ^b), and Effectiveness (ρ^c) fields are 1.0, 0.9, and 0.8 respectively. The learning rate β^a, β^b, and β^c are 1.0, 0.1, and 0.2 respectively. Similar to the action selection process with procedural memory, the agent reasoning system can use the knowledge in the semantic memory by providing the current distance to the opponent while setting up the effectiveness to maximum (the greatest chance of killing) as memory cues. The retrieved values support the agent to decide which weapon to select during the battle. If the cue is not recognized, a random weapon is selected.

Table II illustrates the sample learned rules of weapon effectiveness in symbolic forms. Each rule corresponds to a category node in F_2 layer of the semantic memory. The generalization employed using Fuzzy operators makes it possible to represent the values of antecedents with a range of values. Table II also shows the symbolic categorization of the distance range for interpreting the rules.

B. Results and Discussion

Experiments are conducted by letting *RLBot*, *RLBot++* and the memory-based *RLBot* (i.e. *MemBot*) to individually play against *AdvanceBot*. A single experiment run consists of 25 games or trials, which is counted whenever the agent kills or

is killed by another agent.

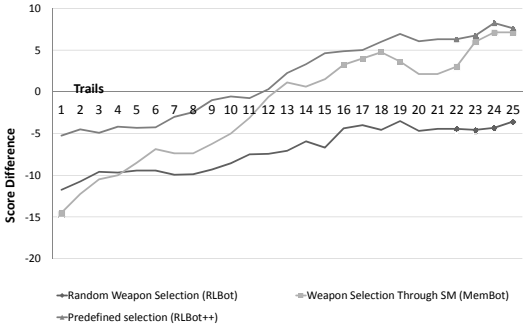

Fig. 10. Performance of *RLBot*, *RLBot++*, and *MemBot* over 25 trials

Figure 10 shows the performance of both *RLBot*, *RLBot++* and *MemBot* in terms of game score differences against *AdvanceBot* averaged over four independent runs. From the performance plotting of *RLBot*, it dominates over its hard-coded opponent gradually. It shows that the procedural memory facilitates the agent through interacting with the environment and enhances its learning capability. By comparing its performance with *MemBot*, the experiment also confirms that the incorporation of the episodic and semantic memory module further improves the learning which results in a much better performance than using the reinforcement learning alone (i.e. *RLBot*). This indicates that the semantic memory successfully learns useful knowledge for the weapon selection portion of the reasoning mechanism. The performance of *MemBot* can eventually reach the same level as the weapon selection optimized rules (i.e. *RLBot++*).

V. Conclusion

In this paper, we have presented a multi-memory system supporting the distributed knowledge representations and parallel memory learning across procedural and declarative memory. Based on fusion Adaptive Resonance Theory, the proposed system promotes the rapid and robust learning on declarative traces as well as effective online reinforcement learning. Within the proposed system, each individual memory module performs independent learning in different paces. The memory modules cooperate closely with each other though memory consolidations and knowledge transfers in order to realize their individual roles and functionalities, as well as to facilitate the process of decision making. The proposed memory system has been evaluated in an online shooting game, wherein the proposed system is used to learn the declarative and procedural memories of an agent. Our experiments also confirms that the proposed system improves the learning of the agent in real time. In this paper, we show that the cooperations among different memory modules emerge more intelligence compared to modeling them in isolation. For the future development of this multi-memory system, we aim to study on more general forms of semantic network, as well as enhancing the dynamic management of each individual memory module.

Acknowledgement

This research is supported by the DSO National Laboratories under research grant DSOCL11258.

References

[1] J. R. Anderson. *Rules of the mind.* Lawrence Erlbaum Associates, Hillsdale, 1993.

[2] A. Cleeremans and J. L. McClelland. Learning the structure of event sequences. *Journal of Experimential Psychology: General*, 120(3):235–253, 1991.

[3] M. A. Conway. *Exploring episodic memory*, volume 18, chapter 1.2, pages 19–29. Elsevier, 2008.

[4] M. J. Farah and J. L. McClelland. A computational model of semantic memory impairment: Modality specificity and emergent category specificity. *Journal of Experimental Psychology: General*, 120(4):339–357, 1991.

[5] G. E. Hinton. Implementing semantic networks in parallel hardware. In Geoffrey E. Hinton and James A. Anderson, editors, *Parallel Models of Associative Memory*, pages 161–187. Lawrence Erlbaum Associates, Hillsdale, 1981.

[6] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24:65–76, 1986.

[7] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[8] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning memory. *Psychological Review*, 102(3):419–457, 1995.

[9] S. T. Mueller and R. M. Shiffrin. REM-II: a model of the development co-evolution of episodic memory and semantic knowledge. In *Proceedings of International Conference on Development and Learning*, volume 5, 2006.

[10] A. Nuxoll and J. E. Laird. Extending cognitive architecture with episodic memory. In *Proceedings of the 22nd national conference on Artificial Intelligence*, pages 1560–1564. AAAI Press, 2007.

[11] P. Perruchet. and A. Vinter. PARSER: A model for word segmentation. *Journal of Memory and Language*, 39:246–263, 1998.

[12] G. J. Rinkus. A neural model of episodic and semantic spatiotemporal memory. In *Proceedings of the 26th Annual Conference of Cognitive Science Society*, pages 1155–1160, Chicago, 2004. LEA.

[13] L. Shastri. Episodic memory and cortico-hippocampal interactions. *TRENDS in Cognitive Sciences*, 6(4):162–168, April 2002.

[14] R. Sun. *Duality of the Mind: A Bottom-Up Approach Toward Cognition*. Lawrence Erlbaum, Mahwah, 2002.

[15] A.-H. Tan. Direct code access in self-organizing neural architectures for reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 1071–1076, 2007.

[16] A.-H. Tan, G.A. Carpenter, and S. Grossberg. Intelligence Through Interaction: Towards A Unified Theory for Learning. In *International Symposium on Neural Networks*, volume LNCS 4491, pages 1098–1107, 2007.

[17] A.-H. Tan, N. Lu, and D. Xiao. Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. *IEEE Transactions on Neural Networks*, 9(2):230–244, 2008.

[18] H. S. Terrace and J. Metcalfe. *The Missing Link in Cognition: Origins of Self-reflective Consciousness*. Oxford University Press, 2005.

[19] E. Tulving. *Elements of episodic memory*. Oxford University Press, 1983.

[20] S. Vere and T. Bickmore. A basic agent. *Computational Intelligence*, 6(1):41–60, 2007.

[21] D. Wang, B. Subagdja, and A.-H. Tan. Creating human-like autonomous players in real-time first person shooter computer games. In *Proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence*, pages 173–178, 2009.

[22] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk. A self-organizing approach to episodic memory modeling. In *Proceedings of 2010 International Joint Conference on Neural Networks*, pages 447–454, 2010.