Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium (FS-11-01)

Ziggurat: Steps Toward a General Episodic Memory

Zachary Faltersack and **Brian Burns** and **Andrew Nuxoll** and **Tanya L. Crenshaw**
University of Portland
5000 N. Willamette Blvd
Portland, OR 97217

Abstract

Evidence indicates that episodic memory plays an important role in general cognition. A modest body of research exists for creating artificial episodic memory systems. To date, research has focused on exploring their benefits. As a result, existing episodic memory systems rely on a small, relevant memory cue for effective memory retrieval.

We present Ziggurat, a domain-independent episodic memory structure and accompanying episodic learning algorithm that learns the temporal context of recorded episodes. Ziggurat's context-based memory retrieval means that it does not have to rely on relevant agent cues for effective memory retrieval; it also allows an agent to dynamically make plans using past experiences. In our experimental trials in two different domains, Ziggurat performs as well or better than an episodic memory implementation based on most other systems.

1 Introduction

Episodic memory is a long-term memory of events experienced first-hand, as identified in (Tulving 1983). For a general artificial intelligence, we offer that episodic memory can support a range of cognitive capabilities, such as noticing novel situations or predicting the success of an action. A handful of artificial episodic memory systems have been implemented, but they rely on direct comparison of small, relevant cues for memory retrieval.

For an artificial episodic memory to be effective, memory retrieval must also be done in a broader context. In humans, direct comparisons of memory cues would be sometimes strange, *"The last time I submitted a research paper about cognitive architectures was on a rainy July 15th. . . "* Direct comparisons are not enough. Memory cues also happen in broader contexts, *"The last time I submitted a paper. . . "*

contextual memory retrieval for an artificial episodic memory system. We have developed Ziggurat, an episodic memory structure and accompanying episodic learning algorithm. As episodes are recorded, Ziggurat induces a hierarchical structure over the episodes, providing for a broader context of each individual episode. This structure not only improves the quality of retrievals, as measured indirectly by agent performance, but allows the agent to dynamically construct a sequence of actions to reach a goal.

We believe that an episodic memory structure should be designed as part of a cognitive architecture. As a result, the memory structure must be strictly domain-independent, capable of operating in a variety of domains such as navigation, tactical decision making and planning. Ziggurat remains domain-independent though it is fully capable of benefiting from the same domain specific information that other systems rely upon.

The rest of the paper is organized as follows. Section 2 describes the two virtual domains we use to investigate Ziggurat's capabilities. Sections 3 and 4 detail Ziggurat and our control agent. Section 5 presents our experimental results. Section 6 describes the related research that has informed our work and Section 7 concludes the paper.

2 Domains

To demonstrate the effectiveness of Ziggurat we have tested it in two distinct environments. The first of these is Eaters, an environment used for research with the Soar cognitive architecture (Newell 1990) and its episodic memory system (Nuxoll and Laird 2007). The second is blind navigation, a virtual domain based upon a real-world robotics testbed we have built called UPBOT, pronounced *yoō-pē-bät* (Crenshaw and Beyer 2010). The agents in the UPBOT environment are iRobot Create robots, the educational version of the popular Roomba. Our virtual domain models the abilities of these robots.

Eaters Domain

In Eaters, the agent occupies a gridworld. At the beginning of the simulation, each cell in the grid contains either food or an obstacle. There are two distinct types of food that provide the agent with positive rewards of 5 and 10 respectively. Food is arranged such that the higher-valued food occurs in evenly spaced columns.

The agent begins in a random cell within the environment. At each time step, the agent can move north, south, east or west provided the destination cell does not contain an obstacle. When the agent enters a cell containing food, it gains the associated reward and the cell becomes empty from that point forward.

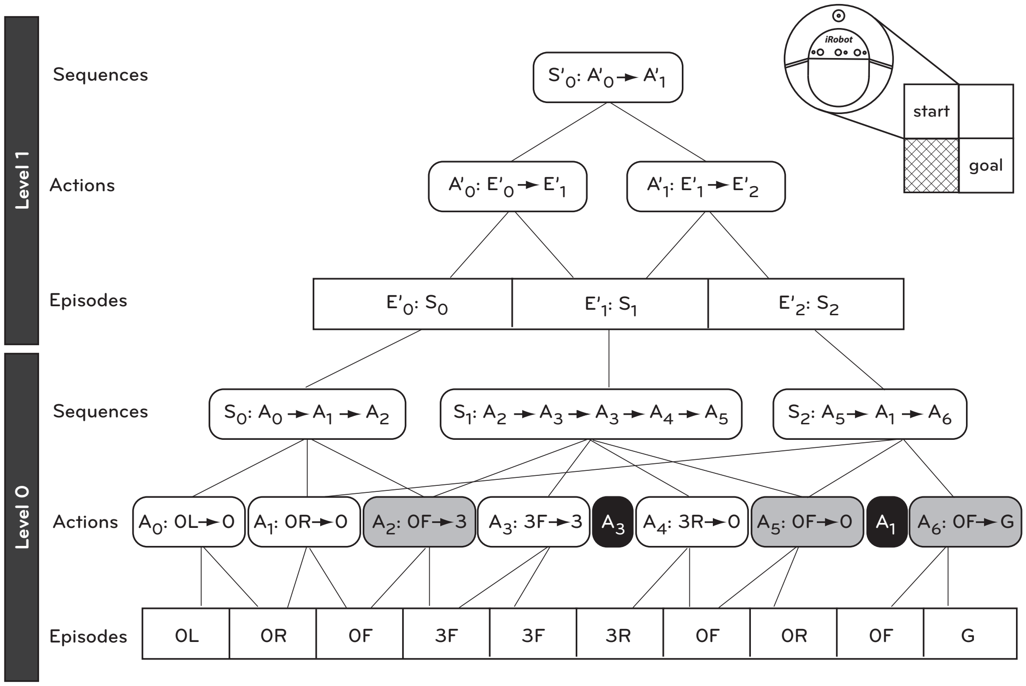


Figure 1: An example of the episodic memory structure built for the blind navigation domain during Phase 1 of the Ziggurat algorithm. A robot navigates a three-square maze. As the robot executes random commands toward the goal, its episodes are recorded in a time-ordered chain; actions and sequences are induced over the chain. Indeterminate actions are shaded grey. Repeat actions, A_3 and A_1, are shaded black. Actions and sequences are complemented by a set of replacement rules induced during Phase 2 as Ziggurat develops plans for finding the goal.

Blind Navigation Domain

In this domain, the agent must navigate from a starting position to a goal position in a maze with a severely limited set of sensors. The agent can only sense the walls around it by colliding with them and activating a pair of bumper sensors. The only other sensor available is the agent's goal sensor which acts as a reward signal.

An especially interesting challenge in this domain is perceptual aliasing (Whitehead and Ballard 1991). This occurs when there is not a one-to-one mapping between the agent's state and every possible state in the external world. In other words, many different situations in the environment look identical to the robot's sensors.

As a result, of the perceptual aliasing and a complete lack of a priori knowledge given to the agent, this domain is particularly difficult. Tests performed with a reinforcement learning agent revealed that it was unable to perform better than a random agent in this domain. However, it is possible to overcome this challenge, to a degree, by leveraging the broader context of recent, previous episodes that an agent has encountered.

3 Ziggurat Algorithm

To offer a clear explanation of Ziggurat, we consider it in two halves. The first half provides the episodic memory structure built from a chain of episodes and a hierarchical structure of building blocks. The building blocks allow for Ziggurat's contextual memory retrieval. The second half provides the episodic learning algorithm. This algorithm induces a collection of replacement rules and learns which rules provide for the best plans to achieve the goal.

That said, Ziggurat's key parts are: the episodes, a hierarchical structure of building blocks, an episodic learning algorithm, replacement rules, and confidence values. We discuss each of these in turn.

Episodes. The content of an episode varies across architectures and domains. Because it is domain-independent, Ziggurat only assumes that an episode comprises the following two parts. First, a set of attribute-value pairs representing the sensing and other internal state at the given timestep. Second, the motor command issued at that same timestep. A new episode is recorded each time the agent issues a motor command.

Consider an example in the blind navigation domain. In Figure 1, a robot is navigating a simple maze, depicted top-right. In this domain, the values 0, 1, 2 and 3 refer to the bumper sensing values, from *no bumpers activated* to *both bumpers activated*. At Level-0, the chain of episodes recorded over time is $\{$0L, 0R, 0F, 3F, $\dots\}$. This may be read as *no bumpers activated, turned left, no bumpers activated, turned right, no bumpers activated, drove forward, both bumpers activated, drove forward* and so forth. The episode in which the goal is achieved is labeled simply, 'G'.

Hierarchical Structure. The hierarchical structure induced by Ziggurat comprises multiple levels of the following building blocks: episodes, actions, and sequences.

Each *level* in the hierarchy is made up of the three kinds of building blocks. At the bottom of each level are the episodes, then actions, then sequences. There are two levels shown in Figure 1, Level-0 and Level-1. Level-0 is the bottom-most level, with the episodes at the very bottom.

Note that at higher levels in the hierarchy, episodes are sequences from the previous level. For example, in Figure 1, the first episode at Level-1 is the first sequence at Level-0. For the results presented in this paper, the number of levels is set at four.

An *action* is constructed from an overlapping pair of episodes. It is a triple consisting of the initial sensing and internal state, the motor command, and the resulting sensing and internal state.

In Figure 1, Level-0, the action A_0 is built from the first two episodes. It may be read as, *if no bumpers have been activated, and the robot turns left, expect that no bumpers will be activated.* Notice that actions A_2, A_5, and A_6 are shaded grey. These are *indeterminate* actions. The left-hand sides of the actions match, the right-hand sides do not. The remaining actions in the figure are *determinate*.

A *sequence* is an ordered collection of actions. Every sequence ends in an indeterminate action, with no more than two indeterminate actions in a sequence. In Figure 1, Level-0, sequence S_0 is constructed from actions A_0, A_1, and A_2. Notice that S_0 ends in action A_2 since it is an indeterminate action. Sequence S_1 also begins with action A_2.

Episodic Learning Algorithm. In the initial phase of constructing the episodic memory, Ziggurat's episodic learning algorithm chooses random commands to find the goal. During this initial phase, episodes are recorded and actions and sequences are induced. In subsequent phases, the episodic learning algorithm uses the set of actions to dynamically construct plans to reach the goal. As the set of actions increases, the agent may build more effective plans. For the purposes of our experiments, we employed a simple planning algorithm that uses breadth-first search to find the shortest plan.

Replacement Rules. Ziggurat's episodic learning algorithm learns how to construct better plans to reach the goal. It does so by learning a set of replacement rules. In the blind navigation domain, a replacement rule might replace three left turns with a single right turn. Such a replacement would reduce the number of moves to reach the goal.

More precisely, replacement rules are constructed from the previously induced set of all actions. A pair of actions is replaced by a new single action. Rules have the form $\{A_{o_1}, A_{o_2} \rightarrow A_{new}\}$ where the left-hand side of A_{o_1} matches that of A_{new} and the right-hand side of A_{o_2} matches that of A_{new}. That is, the initial and expected perceptions of the action-pair matches the new action. Because these replacement rules can be constructed for any level in the hierarchy, one action may actually represent a collection of actions.

Confidence. Ziggurat's episodic learning algorithm maintains a confidence about the agent's state, called *self-confidence*. While this state reflects the agent's position and orientation, it is simply a subsequence of episodes in the chain.

Self-confidence ranges from 0 to 1. As the agent follows the induced actions and sequences, and as the expected sensor data matches the actual sensor data, the agent's self-confidence increases. A higher self-confidence signifies a higher likelihood that the agent is moving towards its goal.

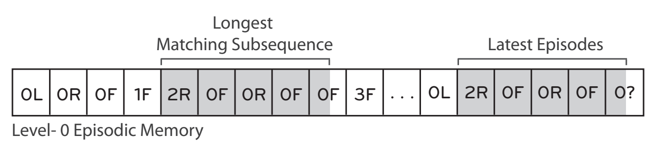
Ziggurat's episodic learning algorithm also maintains a confidence for each replacement rule. Each rule's confidence ranges from 0 to 1, and signifies the rule's past effectiveness. Replacement rules have an initial confidence of 0.5. Suppose three left turns are replaced by one right turn. If the expected sensor outcome is the same, the replacement rule's confidence increases.

Algorithm. We now describe the Ziggurat algorithm. Our actual implementation may be downloaded from our public code repository, http://code.google.com/p/upbot/. Ziggurat executes in two phases. Phase 1 executes only once, during the first attempt at the goal. Phase 2 executes for each subsequent attempt at reaching the goal.

Phase 1. The agent's self-confidence, or C_s, is initialized to 0.5. Ziggurat chooses random commands until the agent reaches the goal. As these commands are executed, the agent's episodes are recorded in a time-ordered chain. During phase 1, the agent's self-confidence does not change, but actions and sequences are induced from the episodes.

Phase 2.

1. **Determine state.** Starting at the highest level of episodic memory in the hierarchical structure, Ziggurat searches for a state which best matches its current state. State matching is performed by searching for the longest matching subsequence in the episodic memory between the most recent subsequence of memories and the remaining memory. Since state-matching begins at the highest level of memories, matches of length 1 take precedence over longer matches at lower levels. The example below demonstrates state matching at Level-0 for the blind navigation domain:



2. **Form a plan.** Based on the match identified in Step 1, Ziggurat uses its induced actions to construct a sequence of actions from its current state to the goal state. In the example above, the first command of the plan would be to travel forward, according to action A_2 in Figure 1.

   (a) **Guess.** If no plan results from Step 2, Ziggurat randomly selects a command that will induce a new action. This command is executed, the resulting episode and action are recorded, and the algorithm returns to Step 1.

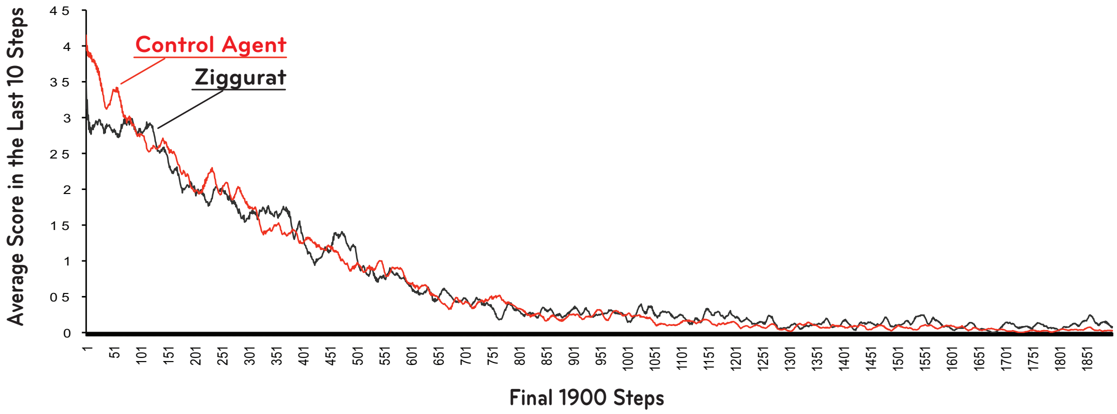


Figure 2: Ziggurat and the control agent were placed in the Eaters World and simulated for 10,000 steps. Every 2000 steps, the world's food sources were refilled. The graph shows the average of 100 executions of the simulation, the last 1900 steps of the averaged executions. In the last 1900 steps, both Ziggurat and the control agent leveraged their knowledge from the previous four food refills, resulting in equal performance.

(b) **Follow the plan.** Ziggurat attempts to follow the plan to the goal. For each action in the plan, Ziggurat may apply a replacement rule. A replacement rule is applied if the following two conditions are met:

- No replacements have yet been applied.
- Self-confidence is greater than (1 - replacement rule confidence).

A new replacement rule is created if the following properties hold:

- No applicable replacement rules exist.
- The self-confidence is greater than (1 - default replacement rule confidence), or 0.5.
- There is more than one step remaining in the plan.

For the domains we employed for this research, the number of possible replacement rules is finite, so a new replacement rule will not be created if all possible replacements already exist.

3. **Recover.** If an action is followed that does not result in the expected perception, then the plan must be reformulated. Perform the adjustments below and return to Step 1.

(a) **Reduce self-confidence.** For the results presented in this paper, self-confidence was reduced by 50%.

(b) **Reduce rule confidence.** If a replacement was attempted, that replacement rule's confidence is reduced. For the results presented in this paper, rule confidence was reduced by 50%.

4. **Declare victory.** If the plan is followed to the goal, perform the adjustments below and return to Step 1.

(a) **Increase self-confidence.** $C_s = C_s + (1.0 - C_s) * d$. For the results presented in this paper, d = 0.5.

(b) **Increase rule confidence.** If a replacement was attempted, that replacement rule's confidence is increased according to the same formula used to increase self-confidence.

4 Control Agent

Our previous research has implemented an episodic memory for the Soar cognitive architecture (Nuxoll and Laird 2007). For the experiments presented in this paper, we compared Ziggurat to a stand-alone episodic memory implementation modeled after that used by Soar. That said, the control agent implementation is also similar to that used by most other episodic memory implementations.

In the control agent implementation, episodic retrieval is activated by a memory cue provided by the agent. Using a nearest-neighbor match, the memory cue is compared to the episodes to find the best match. Ties are broken by choosing the more recent episode. The matched episode is returned to the agent for processing.

At each time step, the agent follows three steps to choose its next action:

1. **Construct a Cue.** The agent constructs a memory cue for each possible action. Each cue contains the agent's current sensing and the candidate action.
2. **Score the Episodes.** For each cue constructed, the episodic memory returns an episode. The agent scores each episode. The score is determined by examining subsequent episodes in the episodic memory to determine the next reward that the agent received. If the reward did not occur immediately, the agent applies a discount factor. If no reward occurs after projecting a certain number of steps in the future, then a score of 0 is applied.
3. **Choose the Action.** The agent chooses the action from the episode with the highest score. Ties are broken randomly.

For our experiments, we used the same discount factor and projection limit that were used in previous experiments with the Soar episodic memory.

Our control agent does lack some features that are present in Soar. Of these, the only one that could affect the results of this research is Soar's use of working memory activation to weight the features of a given memory cue (Nuxoll and Laird 2004). Because of this, we provided the agent with knowledge of an ideal memory cue at each time step, rather than a cue that contained any irrelevant or misleading features. As a result, the activation bias would likely have a negligible effect if it were present.

5 Results

Eaters Domain

Figure 2 compares the performance of Ziggurat to our control agent in the Eaters domain. Ziggurat and the control agent were placed in the Eaters World and simulated for 10,000 steps. Every 2000 steps, the world's food sources were refilled. The graph shows the average of 100 executions of the simulation, the last 1900 steps of the averaged executions. In the last 1900 steps, both Ziggurat and the control agent leveraged their knowledge from the previous four food refills, resulting in equal performance.

In general, the two agents are equally effective in this domain. The control agent resorts to a random move in any situation where all moves are equal, providing for its quick and early short-term gains. Meanwhile, Ziggurat sometimes attempts to make long-term plans that avoid these short-term gains in exchange for a longer-term reward. However, these results support the hypothesis that Ziggurat's ability to make long-term plans does not disrupt its ability to make good short-term decisions.

Results from the Blind Navigation Domain

Figure 3 compares the performance of Ziggurat to the control agent in the blind navigation domain for three different mazes. For each approach, we list the average steps to the goal on the 50^{th} trial based on the knowledge obtained from all the previous trials. Each of the values listed is the average of ten repetitions of the experiment.

As the data show, Ziggurat performs consistently better. As mentioned in the definition of this domain, perceptual aliasing is a serious challenge for an agent in this particular domain. We observed that Ziggurat's sequences made for better retrievals and better plans than the control agent.

6 Related Work

Our work is motivated by the importance of context in human memory, though researchers are not in agreement regarding its exact role. Tulving proposed the encoding specificity principle; the greater the overlap of information at the time of encoding and retrieval, the greater the success of retrieval (Tulving 1983). Though Tulving's principle is too complex to be testable, such context-dependent memory has been demonstrated by numerous studies. One classic study used divers. Words learned on land were more difficult to recall underwater; recall was impaired by a change in context (Godden and Baddeley 1975). Moreover, the implicit memory effect known as priming suggests that recent activation of a memory's representation means that it is more likely to be retrieved (Baddeley 1998).

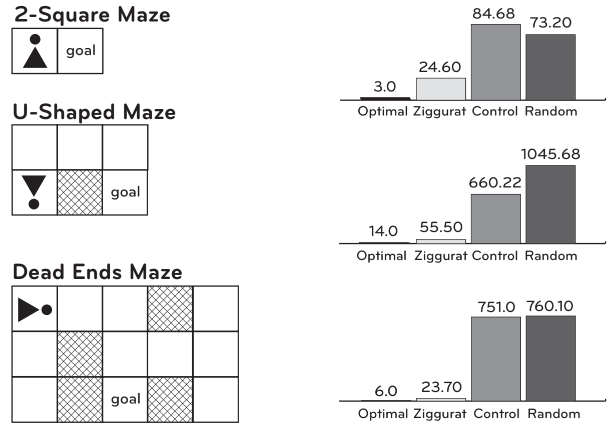


Figure 3: Ziggurat and the control agent were simulated ten times for 50 trials on different mazes. In each maze, the triangle icon indicates the starting position and orientation of the robot (e.g. the robot began facing north in the 2-Square maze). Each bar graph summarizes the average number of steps it took to find the goal in the 50^{th} trial using the knowledge from the previous trials. For comparison, we show the optimal number of steps and a random selection of steps.

While context seems to play a role in much of memory, we focus on context-based retrieval for episodic memory in artificial intelligence. In (Nuxoll and Laird 2007), we demonstrate the capabilities that episodic memory offers to cognitive architectures, from noticing novel situations to predicting the success of an action. Other researchers have recognized the value of episodic memory, and a number of artificial episodic memory systems have been developed. An early episodic memory was developed for an artificially intelligent submarine in a simple aquatic world in (Vere and Bickmore 1990); among its other abilities, it recorded its experiences that could later be queried by researchers. Those developed more recently in (Tecuci and Porter 2007) and (Ho, Dautenhahn, and Nehaniv 2008) have demonstrated their utility in a variety of domains. As in our work, the system presented in (Brom, Lukavsk, and Kadlec 2010) stores memories in a hierarchical format, but it uses domain-specific information to do so. Moreover, we are not aware of any other episodic memory systems that perform context-based retrieval.

Aside from the episodic memory in Soar, explicit episodic memory implementations have appeared in two other cognitive architectures that we are aware of: ACT-R (Altmann and Gray 1998) and LIDA (Franklin et al. 2005). Neither of these implementations bear any resemblance to Ziggurat, but they do reinforce the hypothesis that episodic memory is an important part of human cognition.

Episodic Memory is often compared to case-based reasoning (CBR). While CBR research is rarely entirely task independent, results from using general case representations may yield insights on building a better episodic memory (Bergmann, Kolodner, and Plaza 2005) (Tecuci and Porter 2009).

Our work is also informed by (McCallum 1995) who uses a blind navigation domain to evaluate his variant Q-learning algorithm named Nearest Sequence Memory, or NSM. State is represented as a triple, action, perception, reward. All triples are recorded in a time-ordered chain; the next future action is selected, in part, based on the longest match-length of subsequences in the chain.

Ziggurat induces a hierarchical structure over the episodes to form an episodic memory. There are multiple approaches to inducing some structure from a sequence. One example is (Nevill-Manning and Witten 1997), in which the authors present an approach to learning grammatical rules in a sequence of discrete symbols. Their algorithm, SEQUITUR, forms a grammar from a sequence based on repeated phrases in the sequence. We are not aware of any episodic memory implementations that do this.

7 Conclusion

This paper presented Ziggurat, an episodic memory structure and accompanying episodic learning algorithm. Ziggurat features a novel hierarchical structure that is induced over episodes, allowing for context-based memory retrieval. Like previous episodic memory systems, Ziggurat is domain independent. The only information that must be supplied to Ziggurat in advance is a goal test or reward signal.

As demonstrated by the results of our experiments, Ziggurat is more robust than previous episodic memory systems in some environments, particularly when the context of retrieval and planning is important to succeed at the task.

Ziggurat is memory intensive; as it records episodes, it consumes memory indefinitely. As published in (Nuxoll et al. 2010), a robust forgetting mechanism may resolve this issue and may even improve the algorithm's performance.

For future work, we have begun work on a robust feature-weighting algorithm for Ziggurat's episodes (Walker et al. 2011); we plan to expand that research in the immediate future. Ziggurat might also improve by modifying or replacing the simple confidence-based approach to identifying effective parts of its memory. An established machine learning algorithm might be a more effective way to manage these preferences.

Acknowledgments

This project was made possible thanks to the 2009 and 2010 Arthur Butine grants and grants from the Ed and Sharon Sweo Chair of Engineering at the University of Portland.

References

Altmann, E. M., and Gray, W. D. 1998. Pervasive episodic memory: Evidence from a control-of-attention paradigm. In *Proceedings of the twentieth annual conference of the Cognitive Science Society*, 42–47. Erlbaum.

Baddeley, A. 1998. *Human Memory: Theory and Practice*. Allyn and Bacon.

Bergmann, R.; Kolodner, J.; and Plaza, E. 2005. Representation in case-based reasoning. *Knowl. Eng. Rev.* 20:209–213.

Brom, C.; Lukavsk, J.; and Kadlec, R. 2010. Episodic memory for human-like agents and human-like agents for episodic memory. *International Journal of Machine Consciousness (IJMC)* 2(2):227–244.

Crenshaw, T. L., and Beyer, S. 2010. UPBOT: A testbed for cyber-physical systems. In 3rd Workshop on Cyber Security Experimentation and Test (CSET).

Franklin, S.; Baars, B.; Ramamurthy, U.; and Ventura, M. 2005. The role of consciousness in memory. *Brains, Mind, Media* 1:1–38.

Godden, D. R., and Baddeley, A. D. 1975. Context dependent memory in two natural environments: On land and underwater. *British Journal of Psychology* 66:325–331.

Ho, W.; Dautenhahn, K.; and Nehaniv, C. 2008. Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connection Science* 20(1):21–65.

McCallum, A. 1995. Instance-based state identification for reinforcement learning. In *Advances in Neural Information Processing Systems 7*, 377–384. MIT Press.

Nevill-Manning, C. G., and Witten, I. H. 1997. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research* 7:67–82.

Newell, A. 1990. *Unified Theories of Cognition*. Cambpridge, MA: Harvard University Press.

Nuxoll, A. M., and Laird, J. E. 2004. A cognitive model of episodic memory integrated with a general cognitive architecture. In *Proceedings of the International Conference on Cognitive Modeling*.

Nuxoll, A. M., and Laird, J. E. 2007. Extending cognitive architecture with episodic memory. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, 1560–1565. Vancouver, Canada: AAAI Press.

Nuxoll, A. M.; Tecuci, D.; Ho, W. C.; and Wang, N. 2010. Comparing forgetting algorithms for artificial episodic memory systems. In *Thirty Sixth Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB)*.

Tecuci, D., and Porter, B. 2007. A generic memory module for events. In *Proceedings to The Twentieth International FLAIRS Conference*.

Tecuci, D., and Porter, B. 2009. Memory-Based Goal Schema Recognition. In *Proceedings to the 22nd Florida Artificial Intelligence Research Society Conference (FLAIRS22)*.

Tulving, E. 1983. *Elements of Episodic Memory*. Clarendon Press.

Vere, S., and Bickmore, T. 1990. A basic agent. *Computational Intelligence* 6:41–60.

Walker, B.; Dalen, D.; Faltersack, Z.; and Nuxoll, A. 2011. Extracting episodic memory feature relevance without domain knowledge. In *Proceedings of the Biologically Inspired Cognitive Architectures Society*.

Whitehead, S. D., and Ballard, D. H. 1991. Learning to perceive and act by trial and error. *Machine Learning* 7(1):45–83.