Episodic Memory in a Cognitive Model

David Ménager

University of Kansas, Lawrence KS 66045, USA,
dhmenager@ku.edu

Abstract. The ability to remember events plays an important role in human life. People can replay past events in their heads and often make decisions based on the retrieved information. In this paper, we describe a novel extension to a cognitive architecture, ICARUS, that enables it to store, organize, generalize, and retrieve episodic cases that can help the agent in a variety of manners. After briefly discussing previous work on the related topic, we explain the new extension to the architecture. Then we discuss three architectural implications of the new capability and list some future work before we conclude.

Keywords: episodic memory, case-based reasoning, cognitive architectures, learning from observation, expectations, impasse resolution

1 Introduction

Computational models of episodic memory in the context of cognitive architectures are not discussed very frequently [1]. Our aim is to address this by presenting our theory that accounts for the segmentation of raw experience into episodes, the organization of episodes inside of a larger episodic structure, the retrieval of episodes, and a broad range of episodic memory-based learning within the confines of one framework. In our work, we aim to build a psychologically inspired episodic memory module, and propose it as a type of case-based reasoning. The implementation is integrated within a cognitive architecture, ICARUS [5]. Our work resulted in three new or improved capabilities in our system.

In the sections below, we first describe the background in the literature that serves as basis for our work followed by a brief review of the ICARUS architecture. We then provide a description of how an episodic memory has been implemented in the architecture and discuss how our system maps onto the case-based reasoning framework. We conclude after discussing some architectural implications of our extension.

2 Background

Since the inception of episodic memory theory, researchers have held that any model of it must support encoding and retrieval of experience [8]. Encoding is the process of recording and organizing experiences into the episodic memory. Retrieval is the process of performing similarity-based retrieval to find relevant experiences in this memory.

Table 1. Two sample ICARUS concepts and a skill for blocks world.

```
((on ?o1 ?o2)
 :elements (?o1 is (block ?o1 ^x ?x1 ^y ?y1 ^len ?len1)
            ?o2 is (block ?o2 ^x ?x2 ^y ?y2 ^len ?len2
                                    ^height ?height2))
 :tests ((*overlapping ?x1 ?len1 ?x2 ?len2)
         (= ?y1 (+ ?y2 ?height2))))

((clear ?block)
 :elements (?block is (block ?block))
 :conditions ((not (on ?another ?block))))

((look-right ?robot)
 :elements (?robot is (robot ?robot ^looking ?looking
                                    ^holding ?holding))
 :conditions ((not (eq ?looking 'right)))
 :actions ((*look-right ?robot))
 :effects (?robot is (robot ?robot ^looking right
                                   ^holding ?holding)))
```

Psychological evidence suggests that episodic memory is an index-based long-term memory that supports cue-based retrieval [3][8]. Based on our understanding of this literature, we implemented our extension to ICARUS. Before we describe the implementation, however, it will be useful to review the architecture to facilitate our discussions.

3 ICARUS Review

As a cognitive architecture, ICARUS provides a framework for modeling human cognition. The architecture makes commitments to its representation of knowledge structures, the memories that store these structures, and the processes that work over them. In this section, we provide a brief review of the architecture to facilitate our discussion on the new episodic memory module afterwards.

3.1 Representation and Memories

ICARUS distinguishes two main types of knowledge. These are *concepts* and *skills*. Concepts, in a generic way, describe certain aspects of a situation in the environment. These are stored hierarchically in the conceptual long-term memory. The lowest level of this hierarchy is formed by *primitive* concepts, defined directly from the sensory input, and the *higher-level* concepts are defined in terms of lower-level concepts. For example, the first two entries in Table 1 are concept definitions. The first one, `(on ?o1 ?o2)`, describes a primitive situation where a block is on top of another block. The other `(clear ?block)`, depicts a higher-level situation where nothing is on top of a block.

The other type of knowledge is *skills*. These describe, in a generic way, procedures for achieving certain concept instances in the environment. These are essentially hierarchical versions of STRIPS operators [2] stored in the skill long-term memory. The last entry in Table 1 shows a *primitive* skill definition, `(look-right ?robot)`, that describes a procedure to get the robot to look right. *Higher-level* skills have a similar syntax, except that they include sub-skills instead of direct actions in their body.

3.2 Inference and Execution

The ICARUS architecture operates in cycles. At the beginning of each cycle, the system receives sensory input from the environment. Based on this information, the architecture infers all the concept instances that are true in the current state by matching its concept definitions to perceived objects and other concept instances.

Once all the beliefs are inferred, the system finds all the relevant skill definitions for the current goal(s). ICARUS then chooses one or more of them and executes them in the world. The architecture will continue its cycles in this manner until all of its goals are achieved or its operations are terminated for any other reasons. With this brief review, we now continue our main discussion on the new episodic module in ICARUS.

4 Episodic Module in ICARUS

In the context of episodic memory, ICARUS shares some architectural assumptions with Soar [4]. More specifically, both the architectures assume that episodic memory is a long-term, cue-based system that maintains cues in the agent's working memory, and the agent deliberately encodes experiences as episodes and is able to retrieve them. However, the episodic memory module in ICARUS also has some unique characteristics:

- Episodic memory is a compound structure composed of an episodic cache, an episodic generalization tree, and a concept frequency tree.
- Episodic generalization tree is indexed by similarity.
- Episodes represent durative experiences of variable length.
- Episodic memory is a dynamic structure that supports generalization among sufficiently similar episodes.

Our system encodes new episodes to construct the generalization tree. On each cycle, the agent stores the current state and actions into the episodic cache. It also uses the state to update the count of the concepts in the concept frequency tree. When the agent identifies one or more *significant concept instances*, it begins to encode a new episode. Significance is defined by the notion of rarity. The frequency tree can show which concepts are rare according to the agent's location.

4.1 Encoding

Once the significant concept instances are identified, the architecture tries to explain each one by analyzing information stored in the episodic cache to find a logical process that causes the concept instance to exist. If the explanation attempt is successful, the state–action sequence that explains the significant concept instance will be stored as the new episode. Otherwise, the significant concept instance(s) and the time when they occurred are stored as the new episode. The episode is then encoded into the episodic generalization tree.

Episodes are retained in a top-down, level-order fashion. At each visited episode, ICARUS analyzes the similarity between the episode in the tree, and the episode to be inserted. If the episode in the tree can instantiate the new episode, the system tries to insert the new episode in the existing episode's subtree. This process continues recursively until the episode to be inserted cannot be instantiated by any child episodes of the parent. Once this happens, the episode becomes a sibling of its dissimilar episodes.

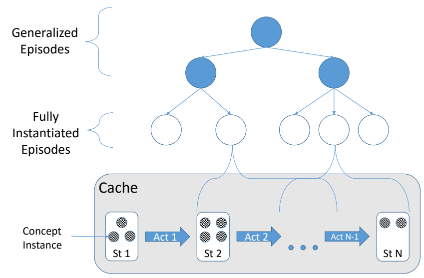

Fig. 1. A block diagram that shows the cache and episodes in the generalization tree.

4.2 Generalization

Once an episode is retained inside the generalization tree, the architecture attempts to find any sibling episodes that can generalize with it. This process plays an active role in creating a hierarchy in the generalization tree. Two episodes can be generalized if they have the same structure, but describe different situations. The generalized episode becomes the parent of the episodes that it generalizes. The root (top-level) node of this tree is the most general episode. Episodes become more specific at each decreasing level of the tree toward the leaf nodes, where there are fully instantiated episodes. This structural organization reflects our understanding of [7], and the notion of episodes representing durative events is consistent with all the literature on episodic memory we are aware of. This hierarchy, illustrated in Figure 1, provides an interesting indexing scheme for the retrieval and storage of episodes.

5 Relevance to Case-based Reasoning

In this section, we illustrate how ICARUS's episodic processes map onto relevant counterparts in case-based reasoning. Some aspects are more closely aligned than others, but we believe that our approach matches the framework well.

5.1 Retrieval and Reuse

ICARUS supports cue-based retrieval of the episodic memory. Given a cue, which can be partially instantiated, the architecture performs a top-down, level-order search of the generalization tree to return one or more matching episodes. An episode in this tree is sufficiently similar to the retrieval cue, if it has the same structure and can instantiate as the cue. Once such an episode is retrieved, ICARUS can adapt it for the current situation by instantiating it with relevant bindings derived from the current state.

5.2 Revision and Retention

On every cycle, ICARUS records the current state and executed actions. If the agent is using a retrieved episode to solve a problem, the executed solution steps (as suggested by the retrieved episode) will also be recorded in the episodic cache. Then the system can create a new episode that may, in fact, be a revised version of the originally retrieved episode and retain it in the episodic memory. The current theory does not support forgetting, but in order to keep the episodic memory from violating space and time constraints, we will need to consider maintenance strategies like flexible feature deletion [6] that account for this phenomenon.

6 Architectural Implications

In this section we describe three architectural implications of the episodic memory as they pertain to case-based reasoning. We plan to demonstrate more applications of our work, but for the moment we supply examples for learning from observation, episodic reuse and making expectations.

Learning from Observation When an agent enters a new environment it may not know how to characterize it, and as a result may be unsure of what to do. As it collects more experiences it may come to knowledge of specific patterns about the environment and formally characterize them in terms of rules.

Episodic Reuse ICARUS encounters an impasse when it does not know how to reach its goal. In such cases it may be advantageous to recall a similar experience to the current one and repeat the actions that yielded the goal. Reasoning with experience in such a manner is one of the ways for which ICARUS gathers support to formalize experiences into rules as discussed above.

Expectations Our system uses the concept frequency tree to create expectations of what concept instances should be true relative to a given environment. As an agent collects more experiences it forms the conditional probability that a concept instance is true, given the agent's experiences of the location.

7 Conclusions

In this work, we described a computational model of episodic memory within the context of a cognitive architecture, ICARUS. We founded our approach on psychological evidence concerning the nature of episodic memory. The extension serves as an important basis for future research, and we argued that our work yields three new or improved cognitive functions for ICARUS related to case-based reasoning.

References

1. Brom, C., Pešková, K., Lukavskỳ, J.: What does your actor remember? towards characters with a full episodic memory. In: International Conference on Virtual Storytelling. pp. 89–101. Springer (2007)
2. Fikes, R., Nilsson, N.: STRIPS: a new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, 189–208 (1971)
3. Hellerstedt, R.: From Cue to Recall: The Temporal Dynamics of Long-Term Memory Retrieval. Ph.D. thesis, Lund University (2015)
4. Laird, J.E.: The Soar Cognitive Architecture. MIT Press, Cambridge, MA (2012)
5. Langley, P., Choi, D.: A unified cognitive architecture for physical agents. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence (2006)
6. Leake, D., Schack, B.: Flexible feature deletion: Compacting case bases by selectively compressing case contents. In: International Conference on Case-Based Reasoning. pp. 212–227. Springer (2015)
7. Schiller, D., Eichenbaum, H., Buffalo, E.A., Davachi, L., Foster, D.J., Leutgeb, S., Ranganath, C.: Memory and space: Towards an understanding of the cognitive map. The Journal of Neuroscience 35(41), 13904–13911 (2015)
8. Tulving, E.: Elements of episodic memory. (1983)