**Soar-EpMem Manual**

Contributors

Jake Czyz

Nate Derbinsky

John Laird

Bob Marinier

Andy Nuxoll

Joseph Xu

Table of Contents

1. Document History 3

2. Soar-EpMem Motivation 4

3. Working Memory Structure 5

4. Storing Episodes 6

4.1. New Episode Creation 6

4.2. Episode Contents 6

4.3. Working Memory Activation 7

4.3.1. The wma Command 7

4.3.2. WMA Parameters 8

4.4. Soar-EpMem Storage 9

5. Retrieving Episodes 11

6. Soar-EpMem Parameters 12

6.1. Parameter Configuration 12

6.2. Parameter Descriptions 12

6.2.1. General 12

6.2.2. Storage 12

6.2.3. Representation 12

6.2.4. Space 13

6.3. Full Parameter Configuration 14

6.4. Parameter Behavior 14

7. Soar-EpMem Statistics 15

8. Trace Information 16

9. Soar-EpMem Programmer Reference 17

# Document History

Version 0.1

Initial specification.

# Soar-EpMem Motivation

Soar-EpMem is a task-independent, architectural integration of an artificial episodic memory (EpMem) with Soar. The EpMem mechanism will automatically record episodes as a Soar agent executes. These episodes can later be queried and retrieved in order to improve performance on future tasks.

# Working Memory Structure

Upon creation of a new state within working memory, the architecture will automatically create a structure in working memory called **epmem**. Within this structure, agents issue requests to Soar-EpMem (see ) by populating the **command** identifier with working memory elements (WMEs), and retrieve Soar-EpMem generated WMEs in the **result** identifier (see ).

# Storing Episodes

This section describes details about Soar-EpMem storage of episodes, including new episode triggering; what is stored; interactions with Working Memory Activation (WMA); as well as where and in what format the episodes are stored.

## New Episode Creation

One functional requirement of an artificial episodic memory is that recording new episodes does not require deliberate action/consideration by the agent. Soar-EpMem provides automatic storage of new episodes during the Output phase of each decision cycle as dictated by the **trigger** parameter.

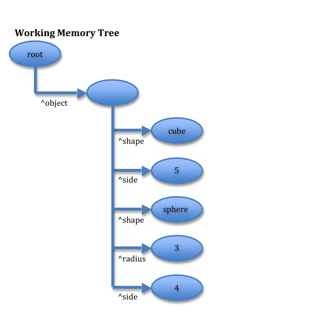
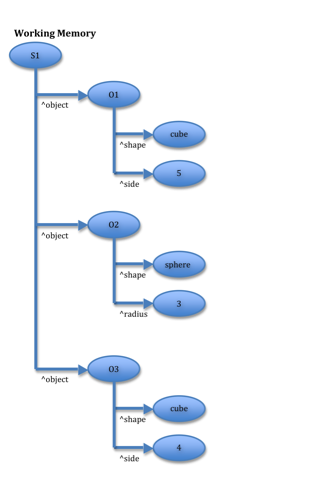
For debugging purposes, the **force** parameter allows the user to manually request that an episode be recorded at the end of the current decision cycle. Behavior is as follows:

* At the end of any decision cycle where the **force** parameter has a value of **on**, a new episode is stored irrespective of the value of the **trigger** parameter.
* The value of the **force** parameter is automatically set to **off** at the beginning of every decision cycle.

## Episode Contents

For an artificial episodic memory to be task-independent it must record most, if not all, information available to the agent at the time of episode creation. Aside from exceptions discussed below, when Soar-EpMem records a new episode, it stores the entire contents of the top state of Working Memory. Consider the following exceptions:

* **Shared WMEs** – Soar agents can create graph structures in Working Memory using shared WMEs. To avoid cycles during storage/retrieval, Soar-EpMem does not store these shared structures. During episode creation, Soar-EpMem traverses Working Memory in a breadth-first manner and only records WMEs upon first encounter. In effect, this reduces Soar’s Working Memory graph to a Working Memory tree.
* **Multi-Valued Attributes** – Queries across multi-valued attributes can become computationally expensive. For performance reasons, storage of multi-valued attribute identifiers is controlled by the **provenance** parameter. When multi-valued attributes are not stored (**provenance** value **off**), children of multiple identifiers are collapsed into a single representative structure. As an example, consider the following illustrated conversion:



* **Manual Exclusions** – There are classes of WMEs that Soar agents may encounter that provide no benefit in context of EpMem. For instance, the “random” WME on the TankSoar input link structure provides a different random number on each update: potentially useful to an agent designer, but most likely will not contribute to effective episodic learning. Moreover, excluding WMEs from storage can provide performance benefits (reduced memory consumption and storage/retrieval time). The **exclusions** parameter allows run-time management of a list of attribute names that will be ignored during Soar-EpMem storage.

## Working Memory Activation

During the episodic retrieval process (discussed in detail later), multiple episodes may match an agent’s query. Nuxoll has produced data that demonstrates improved retrieval quality when using Working Memory Activation (WMA) of WMEs as a form of selection bias. Thus, Soar-EpMem supports integration with Soar with WMA. For a theoretical discussion of the Soar implementation of WMA, consider reading *Comprehensive Working Memory Activation in Soar* (Nuxoll, A., Laird, J., James, M., ICCM 2004).

The following sections detail configuration of WMA, including the **wma** command and WMA parameters.

### The wma Command

Management of WMA within Soar makes use of the **wma** command. Executing the command with no options will print a table of current parameter information. Executing the command with the **--print** (or **-p**) option prints a table of currently activated WMEs for debugging purposes. Finally, the **wma** command has getter (**-g** or **--get**) and setter (**-s** or **--set**) options for retrieving/manipulating parameter values (discussed in the next section).

### WMA Parameters

The following table briefly describes the parameters available for manipulation using the **wma** command’s **get**/**set** options. Further text below provides more information regarding specific parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Acceptable Values | Default | Description |
| **activation** | on  off | on | Turns on/off WMA |
| **decay-rate** | [0,1] | 0.8 | Specifies the speed at which WMEs are decayed |
| **criteria** | o-agent  o-agent-arch  all | all | Specifies what WMEs will have decay values |
| **forgetting** | on  off | off | Turns on/off removal of WMEs with activation values too low |
| **i-support** | none  no-create  uniform | uniform | Specifies the mode in which i-supported WMEs affect activation levels |
| **persistence** | on  off | off | Dictates whether an instantiation activates WMEs just once (default), or every cycle until it is retracted |
| **precision** | low  high | low | Level of precision (relates to performance) with which activation levels are calculated |

decay-rate

The **decay-rate** parameter controls the speed at which WMEs are decayed. A value of 0 will decay WMEs instantly, where 1 will not reduce initial activation level. Note that the value is internally multiplied by -1 (which is reflected upon retrieving the parameter value).

criteria

The criteria parameter specifies the classes of WMEs that will have decay values:

* **o-agent** – Only o-supported WMEs created by the agent (i.e. they have a supporting preference) are activated.
* **o-agent-arch** – All o-supported WMEs including architecturally created WMEs are activated.
* **all** – All WMEs are activated.

i-support

The i-support parameter specifies the mode in which i-supported WMEs affect activation levels:

* **none** – i-supported WMEs do not affect activation levels.
* **no-create** – i-supported WMEs boost the activation levels of all o-supported WMEs in the instantiations that test them. Each WME receives an equal boost, irrespective of "distance" (in the backtrace) from the tested WME.
* **uniform** – i-supported WMEs boost the activation levels of all o-supported WMEs in the instantiations that created or test them. Each WME receives an equal boost irrespective of "distance" (in the backtrace) from the tested WME.

## Soar-EpMem Storage

EpMem currently uses SQLite to facilitate efficient and standardized storage and querying of episodes. The episodic store can be maintained in memory or on disk (per the **database** and **path** parameters). If the store is located on disk, users can use any standard SQLite programs/components to access/query its contents.

Because the most efficient organization and representation of the episodic store is currently under investigation, Soar-EpMem exposes different storage modes via the **indexing** parameter. Our current data shows that the default method, **bigtree\_rit**, maintains high efficiency in most environments, and thus other settings are reserved for experimentation. To facilitate external access to the store, the text below represents the relational schema of the default organizational method:

ids (WME name/value registration)

* child\_id (integer, primary key) - element id
* parent\_id (integer) - element id of parent
* name (text) - WME name
* value - WME value, NULL for identifiers
* hash (integer) - hash of the name/value pair

times (Efficient lookup of valid temporal ids)

* id (integer, primary key) - valid temporal id

now (Registry of “now” element ranges)

* id (integer, primary key) - element id, relates to ids (child\_id)
* start (integer) - temporal id when element instance started

points (Registry of completed valid element ranges lasting one decision cycle)

* id (integer) - element id, relates to ids (child\_id)
* start (integer) - temporal id when element instance started/ended

episodes (Registry of completed valid element ranges)

* id (integer) - element id, relates to ids (child\_id)
* start (integer) - temporal id when element instance started
* end (integer) - temporal id when element instance ended
* node (integer) - RIT node value, expedites intersection searches

left\_nodes (RIT left table, temporary)

* min (integer)
* max (integer)

right\_nodes (RIT right table, temporary)

* node (integer)

weights (Temporary look-up table of element weights during query transactions)

* id (integer, primary key) - element id, relates to ids (child\_id)
* weight (real) - WMA value

It should be noted that currently only the **bigtree\_instance** indexing mode stores WMA values of episode WMEs. The remaining methods make use of the WMA values of cue WMEs at retrieval time.

# Retrieving Episodes

Good retrieval stuff.

# Soar-EpMem Parameters

The following sections discuss how to configure the Soar-EpMem parameters discussed in previous sections.

## Parameter Configuration

Individual configuration parameters are retrieved and manipulated using the **get** and **set** switches of the **epmem** command:

epmem [-g|--get] <parameter>

epmem [-s|--set] <parameter> <value>

Agents can retrieve and change parameters in the actions of rules using the **cmd** function.

## Parameter Descriptions

All Soar-EpMem parameters are organized below. The *Protected* field is discussed in Section 6.4).

### General

|  |  |
| --- | --- |
| Purpose | Enable or disable Soar-EpMem |
| Parameter | **learning** |
| Values | |  |  | | --- | --- | | **off** | Disable Soar-EpMem | | **on** | Enable Soar-EpMem | |
| Default | **on** |
| Protected | no |

### Storage

|  |  |
| --- | --- |
| Purpose | Specifies whether the episodic store will be maintained in memory or on disk |
| Parameter | **database** |
| Values | |  |  | | --- | --- | | **file** | Episodic store is maintained on disk | | **memory** | Episodic store is maintained in memory | |
| Default | **file** |
| Protected | yes |

|  |  |
| --- | --- |
| Purpose | Specifies where on disk the episodic store will be saved |
| Parameter | **path** |
| Values | |  |  | | --- | --- | | **<empty>** | Soar-EpMem will create a temporary database file on disk during execution (and delete it after use) | | **<valid path>** | Soar-EpMem will use the specified path for its database file on disk. If the file doesn’t exist, it will create it. | |
| Default | **<empty>** |
| Protected | yes |

### Representation

|  |  |
| --- | --- |
| Purpose | Specifies the store indexing mode |
| Parameter | **indexing** |
| Values | |  |  | | --- | --- | | **bigtree\_hybrid** | Supplements Nuxoll’s Interval method with a bit-string episode representation for retrievals | | **bigtree\_instance** | Implements Nuxoll’s Instance method | | **bigtree\_range** | Implements Nuxoll’s Interval method | | **bigtree\_rit** | Supplements Nuxoll’s Interval method with a Relational Interval Tree for retrievals | |
| Default | **bigtree\_rit** |
| Protected | yes |

|  |  |
| --- | --- |
| Purpose | Specifies whether multi-valued attribute structure is stored (not implemented) |
| Parameter | **provenance** |
| Values | |  |  | | --- | --- | | **off** | Multi-valued attribute structure is not stored | |
| Default | **off** |
| Protected | yes |

### Space

|  |  |
| --- | --- |
| Purpose | Specifies what triggers new episode creation |
| Parameter | **trigger** |
| Values | |  |  | | --- | --- | | **dc** | Episodes are recorded every decision cycle | | **none** | Episodes are not automatically recorded | | **output** | Episodes are recorded decision cycles when there is a change to the output link | |
| Default | **output** |
| Protected | no |

|  |  |
| --- | --- |
| Purpose | Forces creation of a new episode |
| Parameter | **force** |
| Values | |  |  | | --- | --- | | **off** | Episode recording is dependent upon the current trigger | | **on** | An episode will be recorded this decision cycle | |
| Default | **off** |
| Protected | no |

|  |  |
| --- | --- |
| Purpose | Specifies the degree to which cardinality and WMA are weighted in query processing |
| Parameter | **balance** |
| Values | Numeric, **[0,1]** |
| Default | **0.5** |
| Protected | no |

|  |  |
| --- | --- |
| Purpose | Specifies a list of WME attribute names that are ignored during episode creation |
| Parameter | **exclusions** |
| Values | |  |  | | --- | --- | | **<any string>** | If the supplied value does not currently exist within the exclusion list it is added, otherwise it is removed from the list. | |
| Default | **<empty>** |
| Protected | no |

## Full Parameter Configuration

Entering simply the **epmem** command (with no switches) will return full parameter configuration information. For example, assuming default configuration, the result of executing **epmem** is as follows:

>epmem

EpMem learning: on

Storage

-------

database: file

path:

Representation

--------------

indexing: bigtree\_rit

provenance: off

Space

-----

trigger: output

force: off

balance: 0.5

exclusions:

## Parameter Behavior

Upon attempting to set a Soar-EpMem parameter, the new value is validated. If the value is found to be invalid, the system will use the previous value.

The set of parameters listed above that have a “yes” in the *Protected* field cannot be changed once the Soar-EpMem system has been “initialized.” The Soar-EpMem system initializes during recording of the first episode since starting Soar or issuing the **close** switch of the **epmem** command.

# Soar-EpMem Statistics

Feedback from the Soar-EpMem system is retrieved using the **stats** switch of the **epmem** command:

epmem [-S|--stats] <statistic>

If a **statistic** argument is provided, the command returns the value of a specific statistic. The valid statistic arguments are listed below.

|  |  |
| --- | --- |
| Statistic | **time** |
| Description | Current episode id (starts at 1, increases) |

|  |  |
| --- | --- |
| Statistic | **mem\_usage** |
| Description | Current SQLite memory usage in bytes |

|  |  |
| --- | --- |
| Statistic | **mem\_high** |
| Description | Greatest SQLite memory usage in bytes since last database initialization |

When using the **bigtree\_rit** indexing method, the following additional statistics may be requested for debugging performance issues in the Relational Interval Tree: **rit\_offset**, **rit\_left\_root**, **rit\_right\_root**, **rit\_min\_step**.

Agents can retrieve specific statistics in rule actions using the **cmd** function.

Entering the **epmem --stats** command with no statistic, or an invalid statistic, will return all statistics. A sample execution may look as follows:

>epmem --stats

Time: 0

Memory Usage: 0

Memory Highwater: 0

# Trace Information

To view Soar-EpMem debugging information, use the following watch switch:

watch [-e|--epmem]

This function is not enabled by default or through any watch level. At present, this watch level generates a message when an episode is recorded.

# Soar-EpMem Programmer Reference

Reference stuff