

The ACT-R v5.0 temporal module

Niels Taatgen, Hedderik van Rijn & John R. Anderson

15 March 2005

1 Introduction

At the ICCM 2004 conference, Taatgen, Van Rijn and Anderson introduced an additional ACT-R 5.0 module for time perception (the Temporal buffer). The idea behind this module is that time can be represented by “ticks” which can be thought of as markers that indicate how much time has passed, and which are distributed in time with increasingly wider intervals. This results in high resolution time estimation for short intervals, but less accurate timing for longer intervals. The conceptual underpinnings and additional empirical data are presented in the ICCM’04 paper (<http://www.ai.rug.nl/~niels/temporal.html>) and a CogSci ’05 paper (currently under review), this document describes how to use ACT-R implementation for the Temporal module that can be found at <http://www.ai.rug.nl/~niels/temporal.html>.

Note that the Temporal module does **not** work in ACT-R versions before 5.0.

2 Structure of the Temporal Buffer

The Temporal buffer contains a single slot: `ticks`. The value of this slot is an integer representing how many ticks have passed since last reset of the Temporal buffer. This value is automatically¹ updated as time passes.

3 Initiating the Temporal Module

A call to `initialize-temporal-module` is required *after* calling `clear-all` as ACT-R has to know about the existence of the Temporal module. This not only instantiates the Temporal buffer, but also initializes the initial tick increment, see below under Temporal module parameters.

Repeated calls to `initialize-temporal-module` reinitialize the initial tick increment, adding some noise to the default value. Thus, to simulate multiple subjects, call this function multiple times.

4 Operations on the Temporal Buffer

4.1 `+temporal>`

By issuing a `+temporal>` on the RHS of a production, the Temporal buffer, that is, the current tick count, is initialized to zero. Therefore, `+temporal>` should be thought of as a “reset” of the Temporal buffer count. Currently, issuing a `+temporal>` is the only situation in which the tick count is reset.

¹The automatic updating is accounted for in the function `next-time-tick`, which is scheduled for execution using `queue-command` calls.

4.2 =temporal>

Using =temporal> on the LHS of a production can both be used to harvest the current ticks:

```
(p harvest-temporal
  =goal>
    isa      demo-time
    curtime nil
  =temporal>
    isa      time
    ticks    =ticks
==>
  =goal>
    curtime =ticks)
```

and to match the current tick value against a set value:

```
(p match-temporal
  =goal>
    isa      demo-time
    curtime =ticks
  =temporal>
    isa      time
    ticks    =ticks
==>
  !output!  ("~S ticks have passed!~%" =ticks))
```

4.3 -temporal>

This operation clears the Temporal buffer, removing the current time representation from the Temporal buffer. Moreover, it also turns off the ACT-R internal calculation of time ticks. It can therefore be economical to issue an -temporal> command when a model does (temporarily) not use the Temporal module.

5 Temporal Module Parameters

The length of the ticks is defined as $t_n = a \cdot t_{n-1} + \text{noise}(\text{mean}=0, sd = b \cdot a \cdot t_{n-1})$, given that the first tick is $t_0 = \text{starttick} + \text{noise}(\text{mean}=0, sd = 5b \cdot \text{starttick})$. Note that this is different from the ICCM paper, in which the starttick was referred to as being constant instead of noisy. The value of t_0 is set in initialize-temporal-module.

The table below links the equation's parameters to ACT-R code variables. Note that these variables have to be set using standard lisp setf calls, there is no "sgp :xyz ..." equivalent.

Equation	Default value	Variable
a	1.1	*time-mult*
b	0.015	*time-noise*
starttick	1.1	*master-start-increment*

6 Miscellaneous

- As the Temporal module adds events to the ACT-R/PM queue, it might be wise to set the auto-dequeue PM-parameter (i.e., (pm-set-params :auto-dequeue t)).

- The contents of the Temporal buffer can be examined using the `(temporal)` function.