Real-time Modeling of Reaction Time in Soar: Support for Psychological Models of Fatigue

Randolph M. Jones (rjones@colby.edu)
Computer Science, Colby College
Soar Technology, Inc.
AI Lab, University of Michigan

Basic Goals

- Create a capability to build uniform, real-time models of behavior
- Incorporate a current model of fatigue as an optional element of Soar models

Real-time Psychological Models

- The “Soar Theory of Cognition” claims that decisions in Soar map to a particular time-scale of human information processing
- It appears that we now have fast enough technology that we can make Soar models take as long to make decisions as they are “supposed to”.
- This may prove especially important for models of interactive behavior

Changing the Architecture

- A user-configurable parameter sets the maximum delay per decision
- Each decision incorporates a delay (unless there is a time overrun)
- The parameter can be set to zero (the default), leaving current models unchanged

Incorporating a Model of Fatigue

- Fatigue has been found empirically to impact cognitive behavior in two primary ways
  - General slowing of optimal response time
  - Probabilistic distribution of “attentional lapses”
    - Both are functions of number of hours awake and the cosine of the time of day

General Slowing From Fatigue

- Slowing affects optimal response time, so applies to all cognitive processing
- Simply adjust the time-per-decision parameter dynamically, according to fatigue factors
- User-configurable parameters:
  - Time of day
  - Number of hours without sleep

Attentional Lapsing

- Attentional lapsing occurs with varying frequency and lapses have varying duration
  - Same fatigue parameters as before
- Lapsing *does not* affect "stimulated" behavior
  - Operators with "^fatigue-lapse *yes*" attached will be subject to lapses
- I/O will be "turned off" during lapse
  - Input-link values will be "frozen"

Implementation

- Parallel implementations under construction
- Tcl interface
  - Introduces delays before each decision (using callbacks)
  - Requires Tcl-based I/O
- Kernel interface
  - Effects compiled into Kernel using C code