Interfacing ACT-R 5.0 to AFRL’s Predator UAV STE

26 Jul 2003

Kevin A. Gluck
Jerry T. Ball

Air Force Research Laboratory
Warfighter Training Research Division

This research is funded by AFOSR grant 02HE01COR.
Predator UAV
Surrendering to a Predator
Ground Control Station
Predator UAV High Fidelity Sim
With what external system have you attached ACT-R?

Predator UAV Synthetic Task Environment
How was it done?

UAV STE Computer

Instructor Operator Station (C)

Pilot Station (C)

UAV Control Inputs Processes

VIT

UAV Variable Info Processes

Cognitive Model Computer

Lisp

UAV Control Inputs (C)

UAV Interface Code

AVO Model (ACT-R)

UAV State Repr. (C)

Visual Display Window (PM)

control inputs

variable info table (VIT)
Lisp-based Heads-Up Display
How is communication over the network handled?

- Communication is handled via non-blocking datagram sockets.
What information is communicated back and forth?

• All VIT data are available on the network at every 20 msec update

• **Getting state data from STE …**
  
  — Used ACL’s foreign function interface to allow a C-based process on the cognitive model computer to provide instrument data to the Lisp-based HUD
  
  — Most instrument values on the HUD are digital. Created new visual object classes for horizon line and reticle

• **Sending control inputs to STE …**
  
  — Implemented new “hand on throttle” and “hand on stick” virtual device representations by adapting existing representation for keyboard
  
  — Another C-based process on the cognitive model computer accepts device movement commands from the model and send them to the STE’s control inputs process
How do you handle time synchronization?

• We don’t.
  – The STE runs in real-time (only) and the model runs in real-time (with the appropriate parameter turned on).

• We do coordinate trial starts, however
  – Occasionally, the “start-trial” communications between the model and the STE aren’t successful and we get a bogus trial.
  – This remains an issue.
What didn’t work?  
How did we fix it?

• Model slower than humans
  – Graphics processing issue in ACL 5.0.1 (Lisp HUD)
  – Created a “jump-through-time” production that re-synched every time the model selected a new instrument to attend – not satisfactory
  – Upgrade to ACL 6.2 and more efficient graphics code from Dan Bothell solved the problem
  – Now we turn on the “real time” flag in ACT-R, to slow it down to wall-clock time

• Batch run buffer overloads
  – Model used to grind to a near stop after 20-25 trials
  – Turned out to be caused by buffer overloads in the Debug Window in ACL
  – The fix was simple: stop printing to the Debug Window
  – Now we can run 1000’s of trials without a memory issue
Questions?