Interfacing ACT-R and the X-Plane Flight Simulator

Michael D. Byrne
Departments of Psychology and Computer Science
Rice University
Houston, TX
byrne@acm.org
http://chil.rice.edu/
Acknowledgments

- NASA
  - Funding
  - Guidance and SMEs

- Rice collaborators
  - Postdoc: Volkan Ustun
  - Grad student: Jeff Zemla

- External collaborators
  - UIUC: Alex Kirlik, Kenyon Riddle, Ron Carbonari, Jonathan Silver
  - Aptima: Amy Alexander, Dave Bauer
  - SMEs: Don Talleur, Colleen Cardoza, R. K. Creighton, James Blaisdell
Overview

- The research problem
- The environment
- Our solutions
- Open issues
Surface traffic management is a critical concern for NextGen

- Goal: optimize timing and route for each plane
- Too computationally difficult for human controllers
- Exacerbated by increasing amount of surface traffic

FAA and NASA developing algorithms to calculate optimal routes

HITL experiments with ground controllers

- Require many participants to act as “pilots” to ensure simulation fidelity

Large-scale computer simulations

- No dynamic models of human pilots
- Simulated pilots react in zero time, perfectly predictably
The Environment

- Taxiing a 737 from gate to takeoff
- Unique opportunity to validate ACT-R model against not subjects in simulations, but actual operational data
  - Compare with extensive data taken from real operations at DFW airport (SODAA data)
- How to use these extensive data?
- Environment plays a huge role in shaping behavior
  - Aircraft physics
  - Location of signs and routes
  - Other agents: aircraft, ground controllers, surface traffic
- How to hook ACT-R up to this rich and complex environment?
Commercially available and affordable medium-fidelity ("award winning" too, I’m sure) flight simulation environment

Has plugin architecture; we use two plugins:

- Load and run SODAA data, reproducing a slice of real time at DFW airport
  - Can “take out” one real aircraft, replace with plane driven by ACT-R
- Communicate with Lisp environment
  - Location and state of ownship controlled by ACT-R
  - Locations, orientations, and some properties of all other aircraft
  - The **big** limitation: cannot “see” out the window!
Our Solution

X-Plane

Airport environment

SODAA

PC

Aircraft state
Flight controls

Virtual Cockpit

Surface & signs

Airport Database

network

Virtual information
Physical actions

ACT-R

Knowledge & strategies

Task Analysis

Mac
The “Virtual Cockpit”

A software object in between ACT-R and X-Plane
  • An ACT-R “device”

Gets information from X-Plane
  • Location and state of aircraft, detailed info about ownship

Renders visual world for ACT-R to see
  • Constructs instrument panel based on X-Plane values
  • Renders out-the-window view
    ❖ Driven by database describing taxiways, signage, etc.
    ❖ Combined with location information from X-Plane

Relays commands given by ACT-R to X-Plane
  • Throttle, brake, and yoke adjustments
Issues

• Mostly, it “works” pretty well
  • That leaves rather a lot of room for improvement

• Two biggest problems are labor and time

• Labor
  • Mapping out sections of the airport taxi surface is labor-intensive
    ❖ Not just taxiways and signs, but locations of intersections and other visual markers that are needed by the model
  • Not a lot of shortcuts here
X-Plane runs only in real time

ACT-R generally has no problem keeping up with this

• Even running RMCL on an Intel machine

Biggest problem is indeterminism (and lag) in communication between Lisp and X-Plane

• Sometimes, requests come back very quickly

• Other times, it can take X-Plane/the network hundreds of milliseconds to return with a state update

• Doing a full redraw and PROC-DISPLAY isn’t terribly fast, either

We handle this by periodically launching another Lisp process to ask for an update

• That process updates when ready, so ACT-R doesn’t wait
So, How’s It Working?
So, How’s It Working?
So, How’s It Working?
So, How’s It Working?
Questions?