

An Integrated Toolset for Creating Agents that Explain Themselves

Isaac G. Council

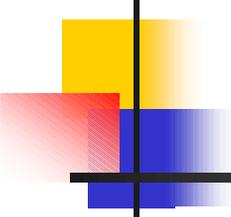
Frank E. Ritter

Steven R. Haynes

05/28/02

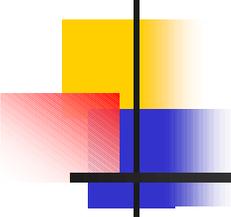
This project is supported by the US Office of Navy
Research, award number N00014-02-1-0021





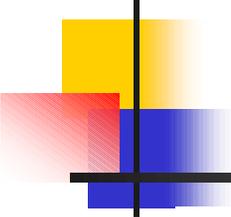
Motivation

- Easier validation of agent behavior
- Increased usability
 - Agents become easier to understand - ala Mycin
 - Provide better training - ala Clancey (Neomycin)
 - Increased potential to develop users' trust
 - Enhanced understanding may enable more creative use of tools



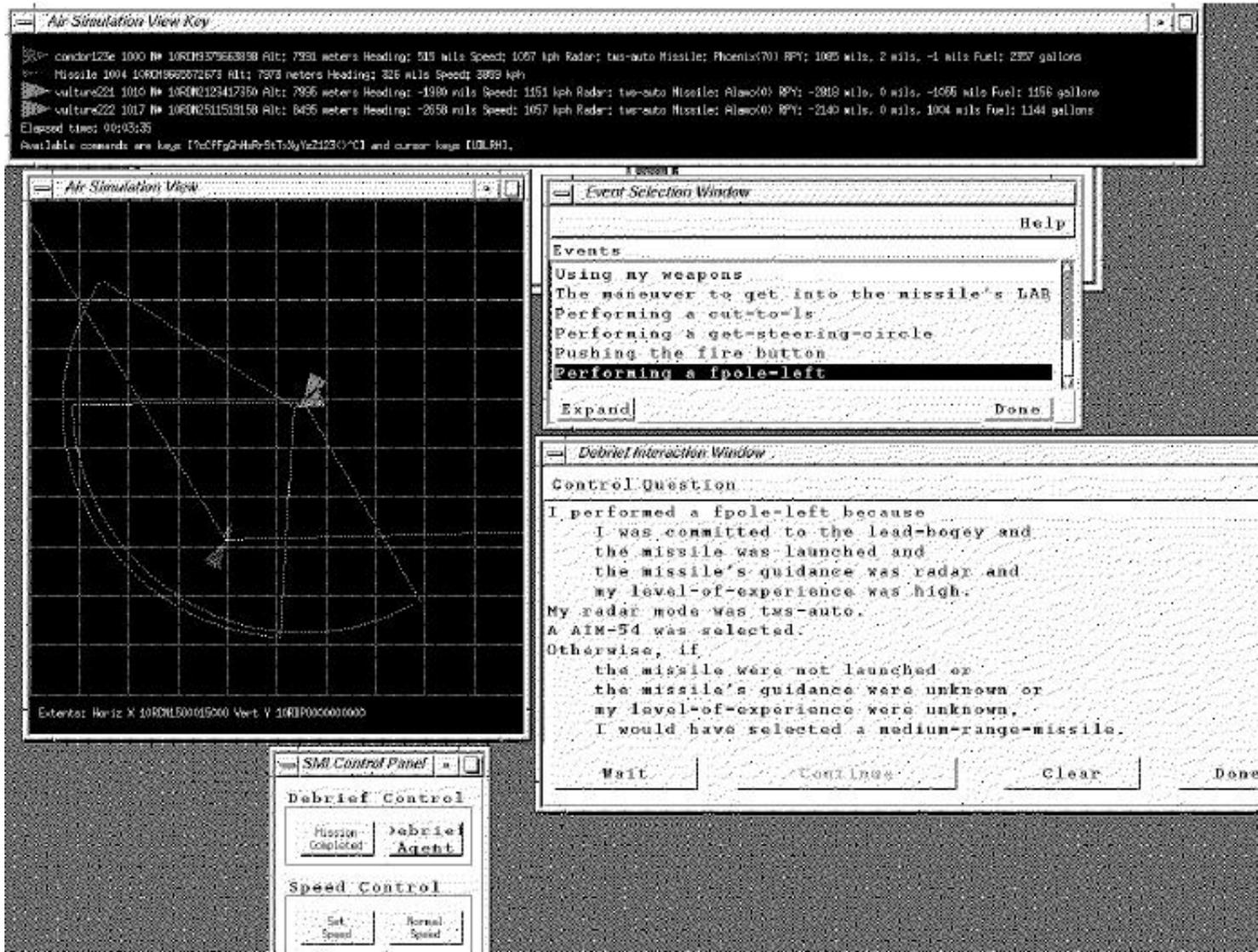
Extending Previous Work - Debrief

- Developed by Lewis Johnson at USC/ISI
- Incorporated into TacAir-Soar



What Debrief Does

- Creates episodic memory in Soar agents
- Menu-driven interface
- Allows an agent to explain:
 - What it did
 - Why it did it (limited)
 - What conclusions it drew about its situation
 - What alternative decisions it might have made, given small changes to the state

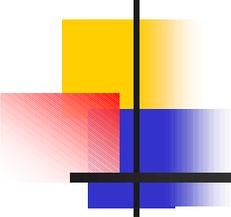


Debrief in Action



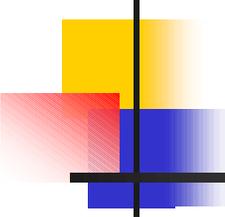
Image taken from Johnson's CGF-94 presentation





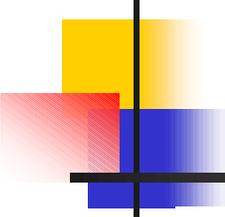
Issues to Resolve

- Currently outdated (runs in Soar 7.3)
- Ease of use
 - Documentation
 - Portability across Soar systems
 - Requires relevant operators to be found and marked by developer
 - Domain specificity?



SAP Usability Data

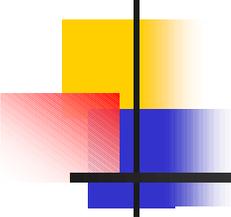
- ~8 hours video recordings of experts interacting with the TacAir-Soar SAP being transcribed (about 70% complete)
- Transcriptions will be analyzed to determine the types of questions users ask during TacAir-Soar sessions
- These explanation requests will be used to validate/extend Debrief functionality



Extending Previous Work - Visual Soar

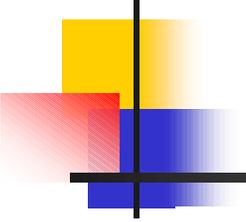
Visual Soar provides an excellent starting point for making Debrief more usable

- May be extended to support integrating Debrief into Soar agents
- May ease the process of sorting and marking explanatory operators through its graphical UI



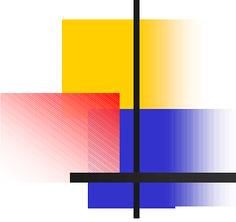
What's Needed

- Means to add graphical marking facility
- Better methods of sorting, searching, and annotating Soar systems - database backend to Visual Soar?
- Something like the Developmental Soar Interface approach, or the Tcl/Tk Soar Interface?



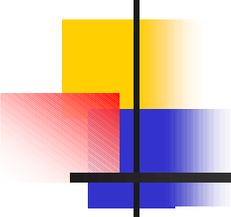
Nuggets

- Potential to make Soar systems more usable
- Code reuse



Coal

- Potential for TacAir-Soar specificity



Web Resources

- Debriefable Agents

<http://www.isi.edu/soar/johnson/debriefable.html>

(contains links to CGF-94 paper and presentation)

- Visual Soar

<http://ai.eecs.umich.edu/~soar/projects/visualsoar/>

Questions?

