COSA² – A Cognitive System Architecture with Centralized Ontology and Specific Algorithms

{stefan.brueggenwirth|axel.schulte}@uniwb.de

Application: UAV Guidance in Manned-Unmanned Teaming

- UAV operator located in a transport helicopter cockpit
- Guidance of multiple UAVs
- UAVs perform reconnaissance of routes and landing sites semi-autonomously
- Real-time data delivered to helicopter crew
- UAV operator assistant system

Solution: Cognitive Automation

- Human centered automation approach
- Inspired by Rasmussens theory on human performance and Endsley’s work on situation awareness
- Collaboration with a human operator on the basis of shared work objectives
- Artificial Cognitive Units are intelligent agents using the COSA Framework

Implementation: COSA

- Originally implemented using Soar as inference engine and CORBA as middleware
- Reimplemented using graph transformation techniques with an evolutionary search plan optimizer

Implementation: COSA²

- Using reimplemented inference engine
- External PDDL planner
- Linear optimizer CPLEX for plan adjustment and scheduling
- Spread-based middleware

Identification

- Rule-based inference step
- Transforms low-level, subsymbolic sensory information into matching high-level concepts
- Object oriented, typed syntax
- Based on Eclipse Plugin CML (Cognitive Modeling Language)

Goal Determination

- Rule-based goal instantiation step
- Activates currently pursued goals dependent upon matched concepts
- Support for hard goals (removed-at-end, may-never-occur type)
- Support for soft goals (penalty on occurrence)

Planning

- Projection of feasible tasks candidates against identification and goal determination rules
- Generates task agenda that satisfies hard-goals while optimizing soft-goals
- Strips based notation with preconditions and expected effects
- Uses PDDL 2.2 AI planner

Procedure Based Behavior

- Reactive layer with prestored behavior decreases response time
- A matching pattern immediately fires associated procedure
- Concept-based behavior is suspended, but may project expected outcome using forward chaining to evaluate benefit of reactive rule

Plan execution and Monitoring

- Task-Agenda contains tuple with task and triggering cue
- Using ILOG CPLEX for linear optimization and scheduling of parameters
- Deviation between actual environment situation and expected preconditions or effects triggers replanning