

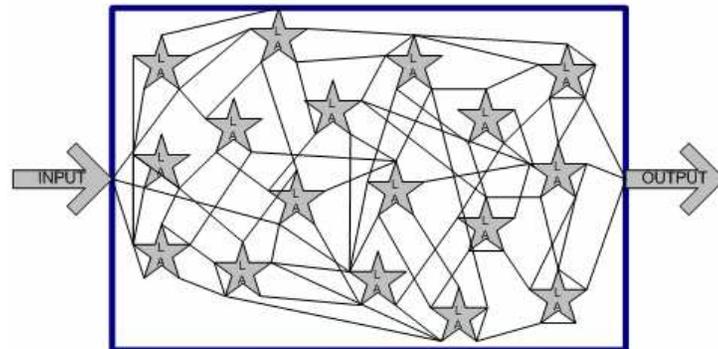
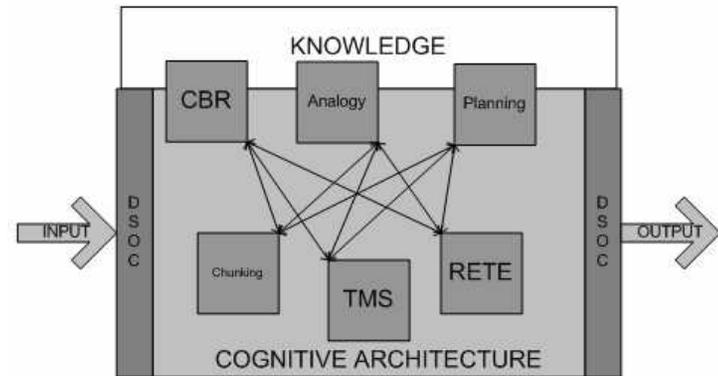
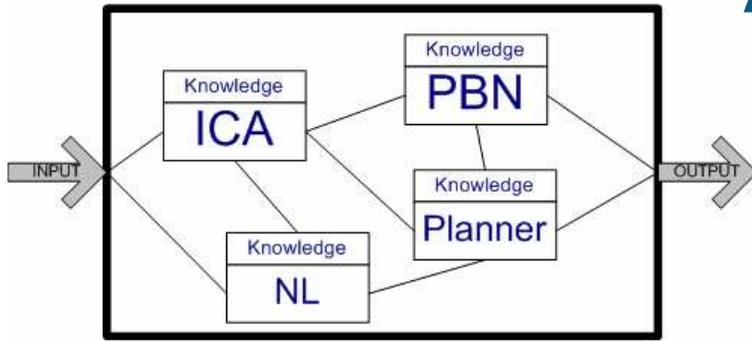


Modularity in Soar-based Applications: Practical Issues

25 May 2006

Robert E. Wray
www.soartech.com

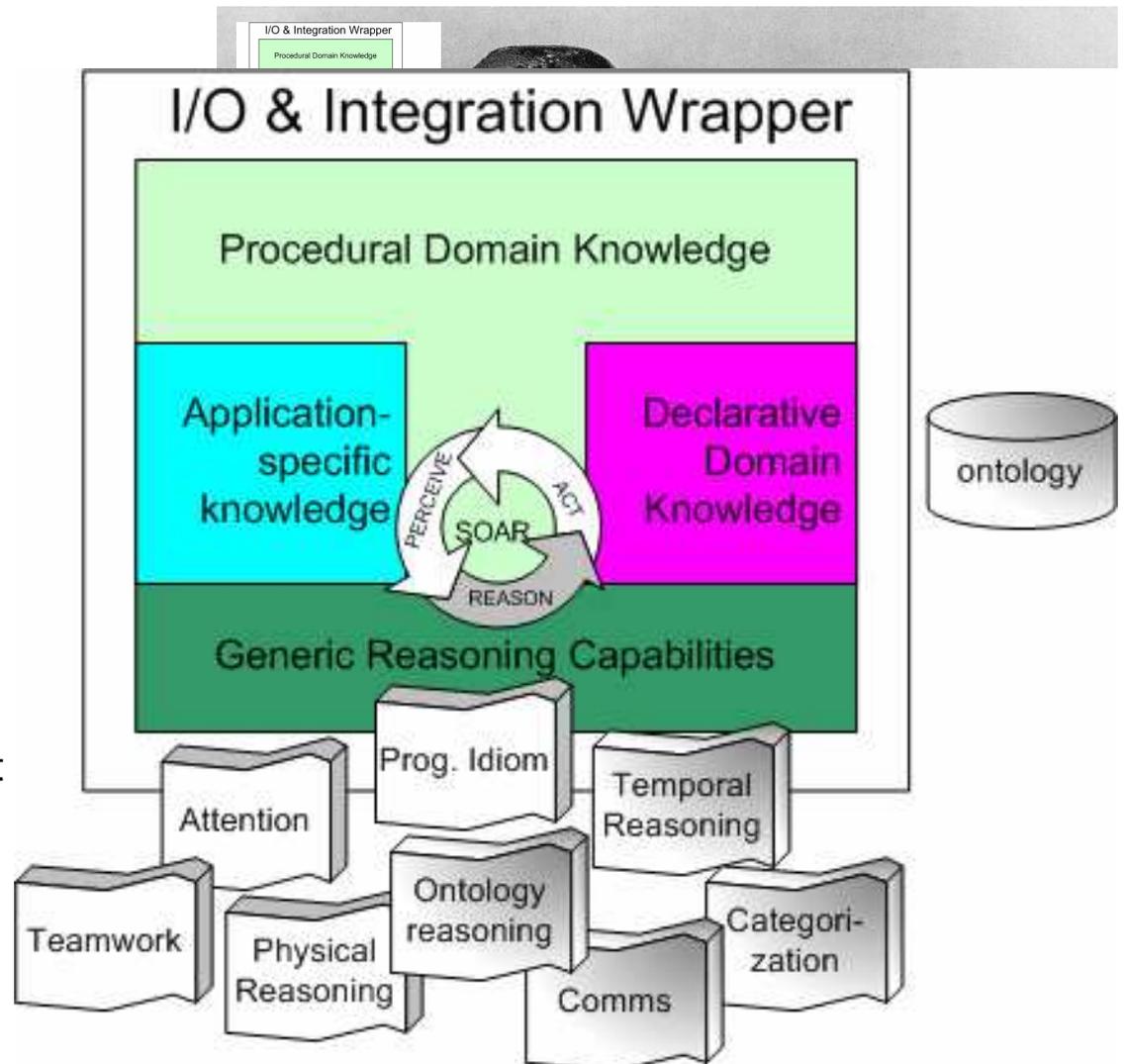
AI system engineering options



- Software Engineering (GOFAI):
 - Modular decomposition derived from functional decomposition
 - Fixed, engineered interaction & control
 - Knowledge matched to module
 - Black-box module operation
 - Includes “engineered MAS” approaches
- Cognitive architecture approach
 - Uniformly encoded knowledge
 - White-box knowledge modules
 - Least-commitment control and knowledge integration
- DAI/Multiagent System approach
 - Opportunistic, unscripted interaction
 - Distributed (“no executive”) control
 - System behavior is “emergent”

Typical Soar-based Application (HBR/CM)

- Agent maps to human actor in a physical environment
- Agent should exhibit capabilities roughly comparable to the human agent in the environment
- Typical HBR/CM implementation is consistent with Soar theory:
 - Soar is the sole “intelligence” platform
 - All knowledge is dynamically integrated at run-time within Soar
 - Examples:
 - NASA-TD, TAS, RWA (STEAM), AMBR (SCA), etc.
 - Multiple agents: multiple instances of Soar

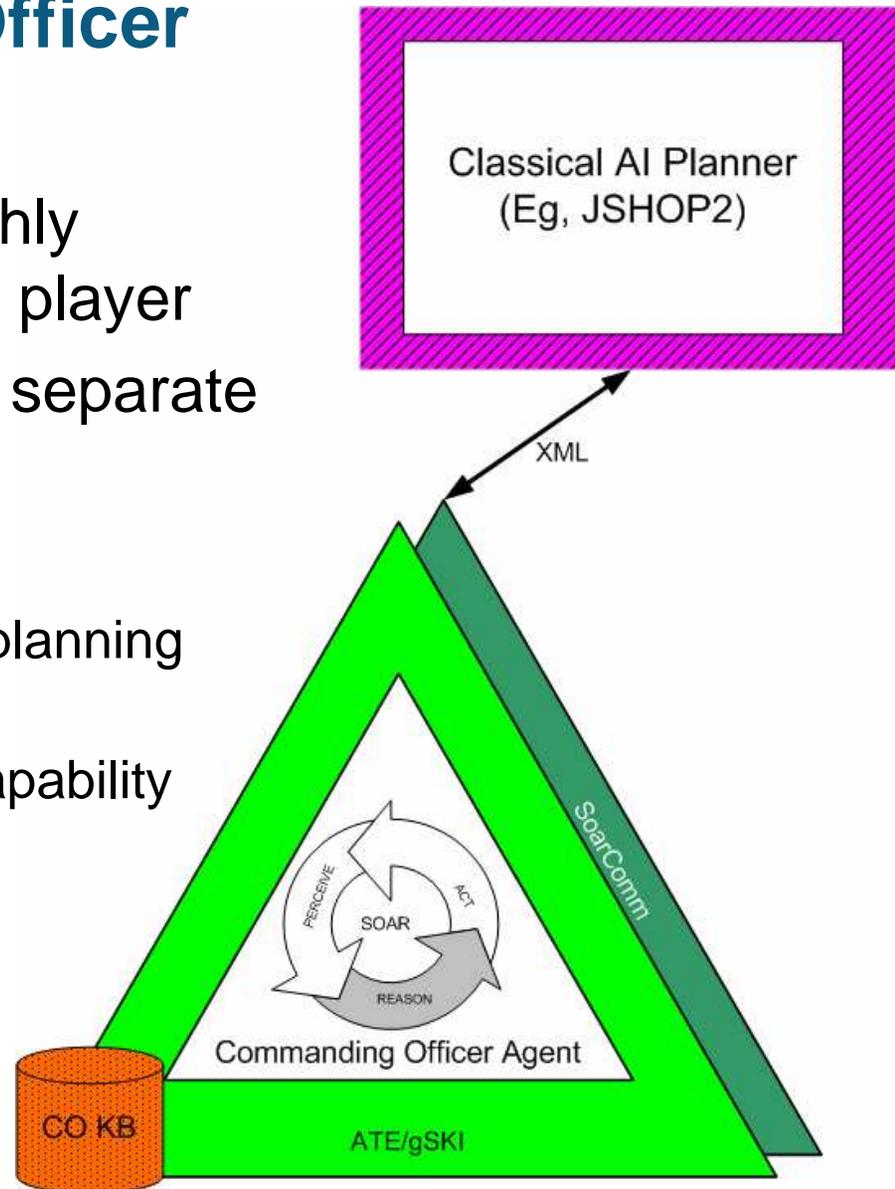


Recent Soar Technology Application Architectures

- Current application development at Soar Tech demonstrates strong reaction to some practical constraints of Soar
 - Result: System architectures beginning to look more like GOFAI systems than systems constrained by Soar theory

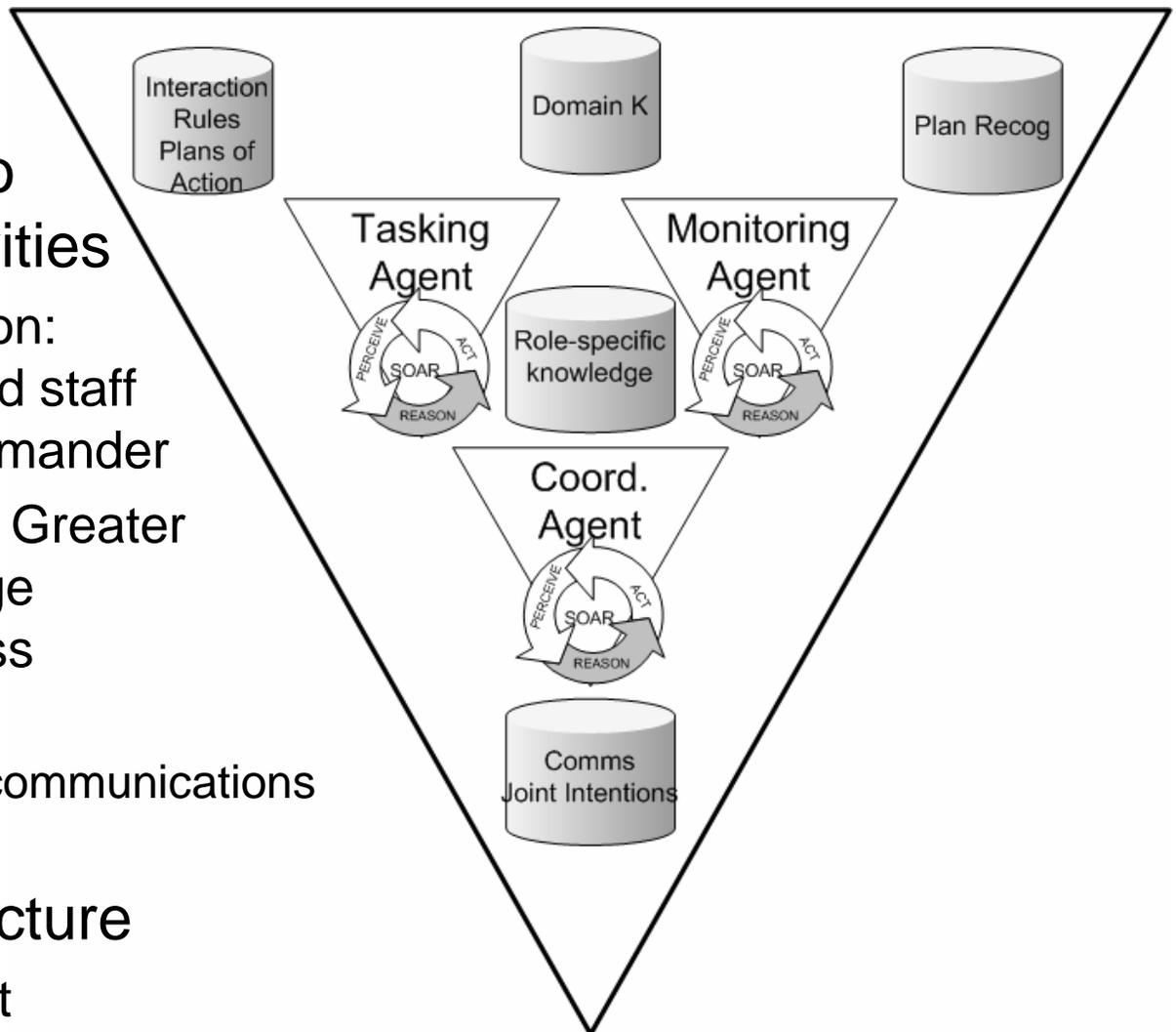
JFETS Commanding Officer

- “Command” function roughly comparable to RTS game player
- Realized as Soar agent + separate planning system
- Motivation:
 - Maturity & performance of planning systems relative to Soar
 - No off-the-shelf planning capability

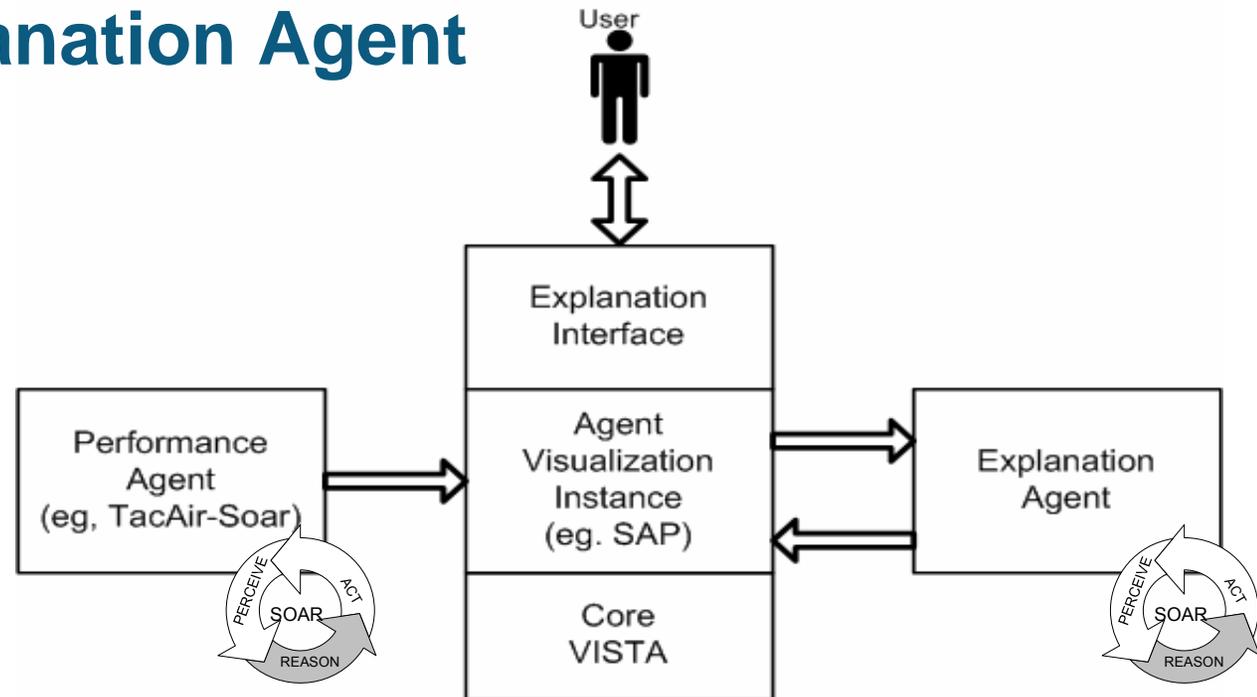


Commanding Officer Decomposition

- Decomposes command role into three distinct activities
 - Original assumption: model of command staff vs. individual commander
 - Partial motivation: Greater reuse of knowledge components across different domains
 - joint-intentions communications knowledge
- Extensible architecture
 - visualization agent

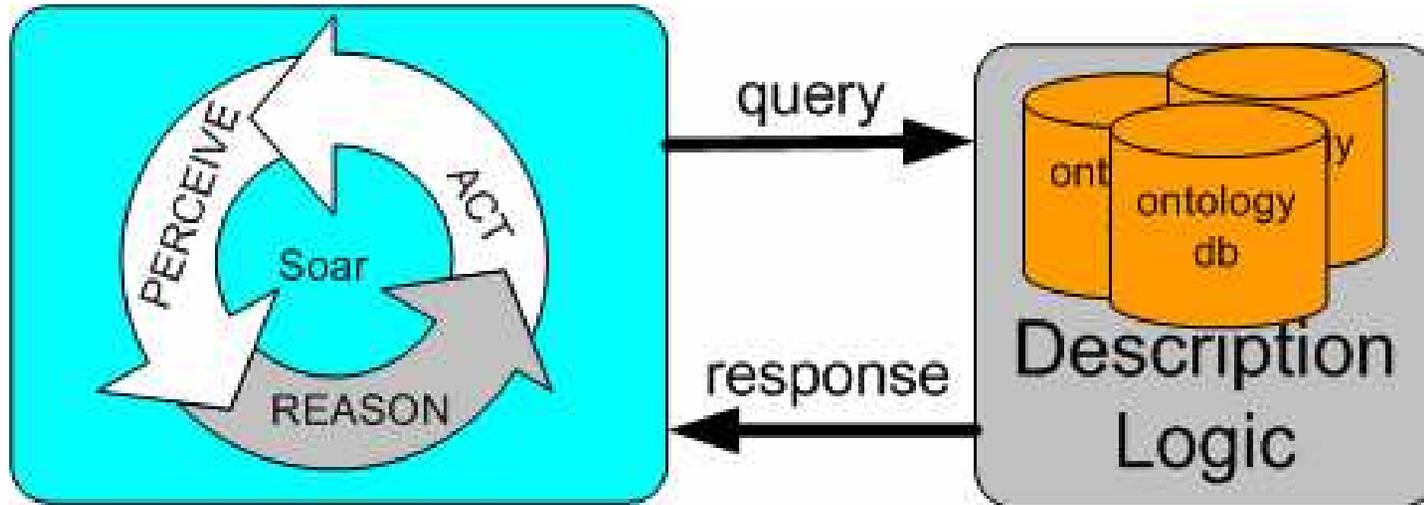


VISTA Explanation Agent



- Generation of explanations realized in a separate agent
- Architecture facilitates explanations from non-Soar agents
- Motivation:
 - Reuse of explanation capability
 - No methodology/tools for (consistent) aspectual encoding (Debrief)

Onto2Soar (proposed 2nd-generation system arch)



- Move formal ontologies outside representation within Soar and use description logic tools to resolve ontology queries
- Motivation:
 - Incompleteness of Soar (associative, semantic retrieval)
 - Maturity of description logic tools
 - Improved performance, scalability
 - “Abuses WM” with obvious consequences...

Practical Issues in Soar Application Development

What is driving divergence from theoretical assumptions?

- Soar theory:
 - Completeness in functionality
 - Knowledge reuse across applications (especially knowledge for general capabilities)
 - Architectural assumptions enable run-time knowledge integration (interleaving and open, not encapsulated, knowledge dependencies)
- Application-development constraints:
 - Current release of Soar is not complete (research in progress)
 - Special-purpose mechanisms (eg, planners) offer significant raw performance improvements
 - Knowledge reuse is rare, not the rule
 - Knowledge development cost tends to scale super-linearly
 - Large knowledge bases do not necessarily provide adequate performance
 - No knowledge packaging methodology/tools
 - Accessibility of Soar
 - Soar is used for many non-HBR/CM applications
 - Are “Soar claims” specific only to human-inspired models, or to intelligent systems generally?

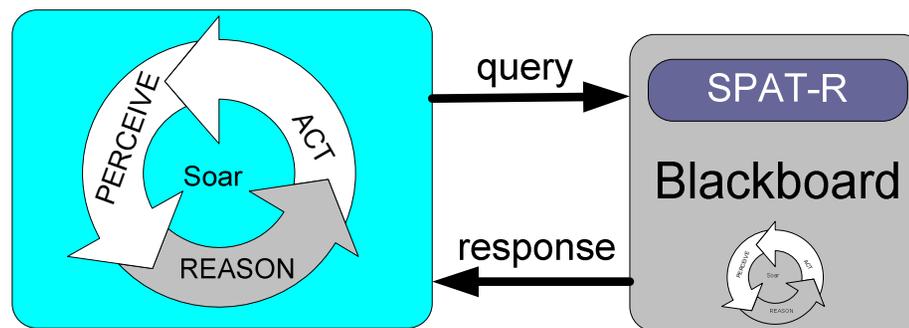
Potential Limitations of SE/GOFAI Approach

- Functional decompositions are ad hoc
 - Right functional decomposition for one problem is not the “right” functional decomposition for the next problem → poor reuse, lots of re-engineering, very costly
- Black-box modules
 - Scalable software engineering (divide-and-conquer; strong encapsulation) results in pre-defined, limited context → Limited insight/reach into another module
 - (Likely) violates “represent once” goals
- System-level engineering
 - Pre-defined interactivity and behavior → High brittleness when application requirements change (little adaptiveness to new requirements)

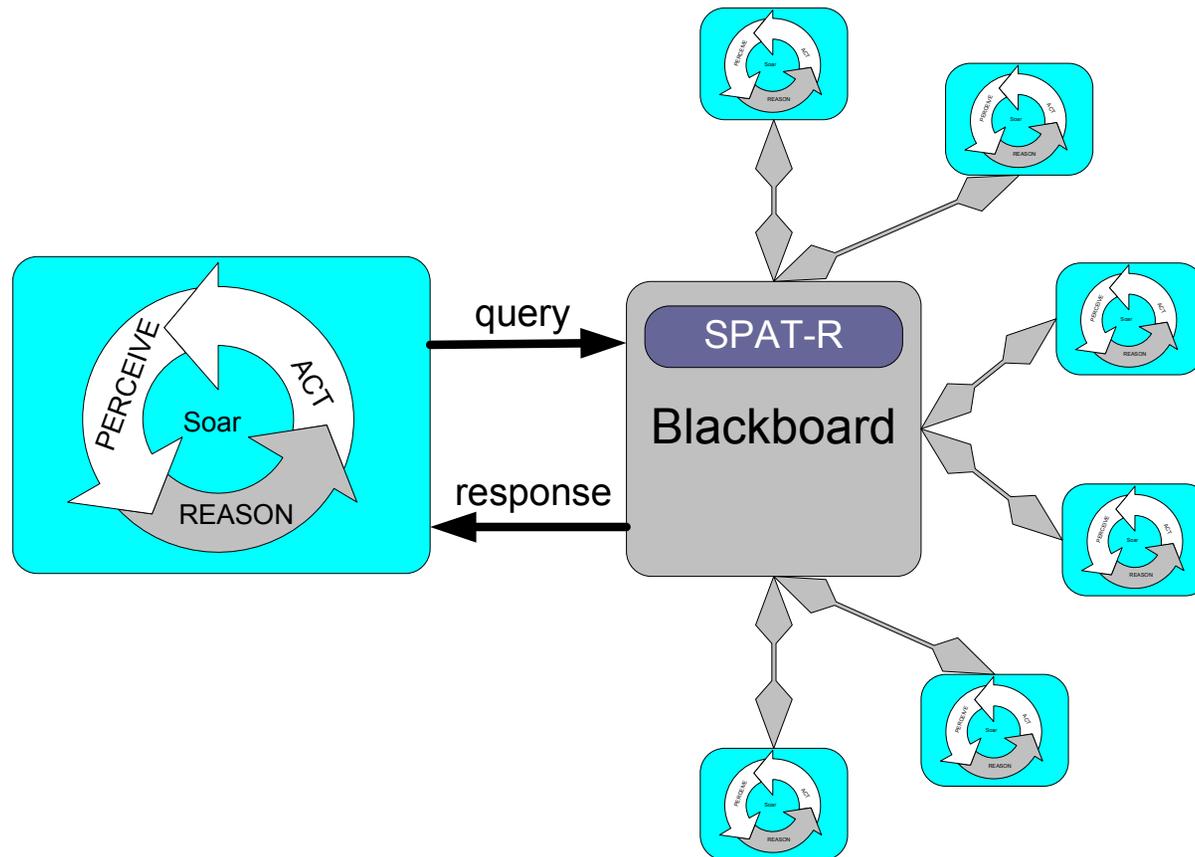
Directions for application-development tools

- Research in knowledge packaging
 - Initial explorations/lessons: SCA (impasses), STEAM (annotations)
 - High utility for both traditional and GOFAI approaches
 - Preserves/enables “white box” modularity?
- Research/engineer “semantic interfaces”
 - Define good abstractions for Soar-Module information exchange
 - Examples:
 - Soar-planner interface, SPAT-R; Visual Imagery? Semantic memory?
- Research/engineer enabling technology for interfaces
 - Blackboards
 - Agent memory as blackboard (ala JESS)
 - Distinct shared memory component
 - Soar agent as blackboard (ala AIS)
 - Communication infrastructure
 - Understand and document trade offs!

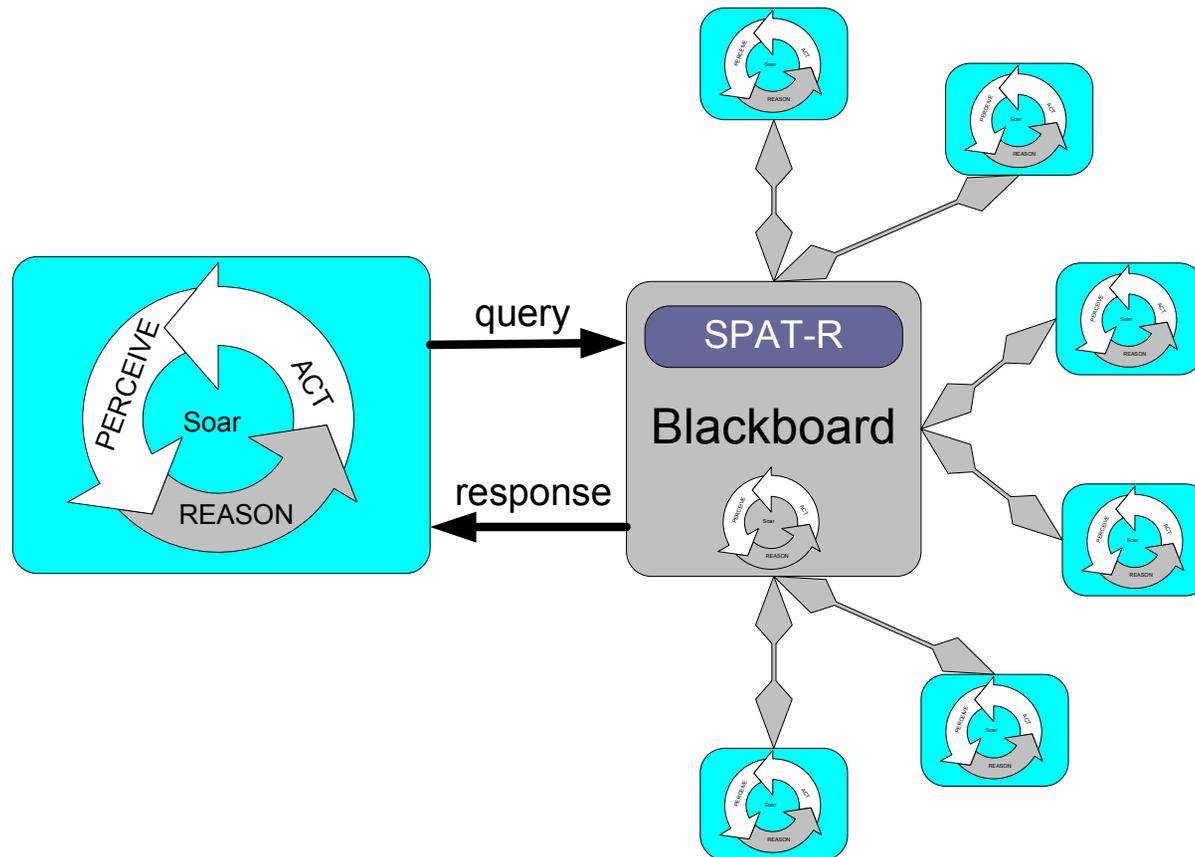
External functional component...



External shared functional component...



External shared intelligent component...



Conclusions

- Nuggets
 - The cognitive architecture approach is a hybrid of GOFAI and MAS:
 - Open, “emergent” least-commitment interactions at symbol level
 - Tightly integrated, fixed, optimized interactions at architecture level
 - Soar theory offers compelling story for least-commitment control and dynamic knowledge integration
- Coal
 - Many practical limitations impede realization of theoretical benefits
- Unresolved questions
 - Which system engineering approaches are most appropriate for what kinds of applications?
 - What are “natural” units of agency for different kinds of applications?
 - How are Soar constraints in non-HBR systems useful? Informative?
 - What specific research and tools are needed to support the different directions for supporting application development?
 - How could Soar best be applied in DAI systems (if at all)?