Toward a Standard Rule Language for Semantic Integration of the DoD Enterprise

A MITRE Sponsored Research Effort

Suzette Stoutenburg
28 April 2005
Purpose

- Identify use cases for application of a standard rule language in Department of Defense (DoD)
- Present research plan: *Toward a Standard Rule Language for Semantic Enterprise Integration*
- Discuss early findings of research
- Outline the way ahead
Motivation

- To defeat emerging threats, DoD systems must be dynamic and adaptable
- Separation of rules from application code supports the ability to dynamically modify system behavior in complex, changing environments
- To realize the benefits of rule separation, a Rule Language Standard is required to enable agility and interoperability across the enterprise
Use Cases

• Dynamic Service Oriented Architectures
• Rapid Enterprise Integration and Reuse
• Complex Semantic Integration and Synthesis
• Dynamic Information Sharing and Mediation
• Machine to Machine Interactions
• Computer Vision
Use Cases, cont.

• Dynamic Service Oriented Architectures
Use Cases, cont.

- Rapid Enterprise Integration and Reuse
  
  - New capabilities are more easily integrated
  - Rules can be reused
  - Engine can be reused
  
  - New Node
  - Legacy Sensor Rules
  - New Sensor Rules
  - Enterprise Node
Use Cases, cont.

- Complex Semantic Integration and Synthesis
Use Cases, concluded

• **Rules for Dynamic Information Sharing and Mediation**
  - Security enforcement: blocking & transformation
  - Syntactic & semantic mediation
  - Discovery
  - Event driven publication

• **Machine to Machine Interactions**
  - Allocation of battlefield capabilities
  - Battlefield event monitoring
  - Event driven, automated communication

• **Computer Vision**
Research Objectives

• Develop demonstrable recommendations for a standard rule language
• Identify DoD specific requirements by applying rules to a real world problem
• Build experiments to show how ontology and rule layers should interact and compare approaches
• Look ahead to future requirements for the evolving standard
Early Findings and Thoughts

- Standard method of expressing uncertainty and provenance is required
  Estimation of truth required in ontology & rules
- Need to distinguish battlespace objects from reports about them
- Orchestration issues exist with dynamic classification
- Need to persist derived states in time
- Closed world vs. open world predicates
  - How do we control the discovery process?
  - Do we need a hybrid?
- Translation to executable environment is difficult
- Integrated framework of tools, languages & standards must evolve
The Way Ahead

- Complete first year research
- Explore more complex, dynamic rules and how those affect the rule standard
- Examine impact of rule exchange across multiple disparate domains
- Investigate self-maintaining ontologies and rule sets
- Explore ontological closure and annotation for rule discovery
Backup Slides
Research Plan: Year 1

- Build experiment to develop recommendations for standard rule language
  - Select mission use case and capture ontology & rules
  - Build an application with layered ontology and rules
  - Build second application with integrated ontology and rules
  - Compare how each performs and observe issues with interaction and orchestration across layers
  - Keep all factors constant other than differences in languages used
Interesting Questions

• What are the advantages of an integrated vs. layered approach for specification of ontology and rules?
  If an integrated language is best, then how should it be structured to express different rule types and different logics? How tightly should they be coupled?
  If a layered approach is preferable in which rules and ontologies are separate, then how should these layers interact? Can the rules be separated from the concepts? If so, how?

• How should inferencing best be orchestrated?

• Do the approaches lead to deterministic systems?

• Would the use of metadata about the ontologies and rules help in automating the orchestration process?

• Are there certain properties of ontologies that will make them more or less tractable with certain execution environments?

• Should OWL, RuleML and SWRL be translated to executable environments? Or should integrated SWRL engines be developed and integrated into DoD systems?