Manufacturing Engineering

Requirements for Rules Interoperability

Ed Barkmeyer, Ravi Raman, Evan Wallace

Manufacturing Engineering Laboratory

NIST

NIST

General Architecture

- Multiple ontology and rules languages
- Reference concept set(s) (meta-models)
  - first-order, temporal, deontic, SCLP, data logic
  - reasoner class/style labels (DLs, SCLP, etc.)
- Rigorous foundation
  - Multiple incompatible foundations?
- Reference mapping from each language to (a set of) meta-concepts
- Standard exchange form
  - multiple standard forms?

Discipline

- Flyswatter principle:
  - don’t use a feature that requires a power reasoner just because it is easier or clearer
  - don’t require circumlocutions: automate transforms to tractable structures
- External ↔ internal transforms
  - make assumptions clear to author
  - don’t lose or add information
  - results must make sense in external form
- Logic safety
  - develop discipline for testing and verification of ontologies and rulesets
  - need to be able to exchange proofs (PML?)
  - don’t make users read traces

Common Rules Needs

- Converting measurements in different units
  - real arithmetic, exponents: mi/gal → litre/100km
- Resolving structural differences in representing the same information
  - organization of the elements of person-names
  - time intervals: (begin, end) vs. (begin, duration)
- XML Schema "restriction"
  - limit instances of class permitted in a given usage
  - limit properties permitted in a given usage
  - limit occurrences in a given usage
- Reasoning about region containment, intersection
  - both geometric and geographic

Rules in (Engineering) Models

- variables to support co-reference
  - instances of a class with the same hasLocation value
  - recognize siblings by common hasParent value
- partOf properties
  - parts derive properties from whole
  - assemblies rollup properties of parts: sets, sums
- arithmetic constraints on multiple properties
  - car.AC.weight + car.engine.weight ≤ 1000 kg
- implication between two properties for a given class
  - if car has towing package, then car has heavy duty transmission
- exclusion between two properties for a given class
  - Model X cannot have both an automatic transmission and a supercharged engine

Kinds of Rules

- Information model rules
  - describe the “business objects”
  - define validity of the information base (“consistent state”)
  - useful for inferencing
- Business logic and workflow rules
  - specify requirements for the behaviors of agents
  - guide choreography of business applications
- Semantic Webservice specifications
  - defines agent behavior (pre-condition/post-condition)
  - support dynamic integration, interoperability of software,
  - enables reliability of systems
- Need integration/interoperation of all kinds of rules

Operating Rules Environment

- External agents
  - webservices and workflows
  - rule activates function outside the engine
  - agent may be automaton or human
- Concept of time
  - relative: true now, later, before X, after X
  - rule firing must be synchronized with events
- Complex information space
  - current info from multiple sources
  - rules engine and agents maintain consistency

Rules Engine Requirements

- Decision structure language
- External events represented in the language
- External events and information communicated to engine knowledge base
- External agent invocation (protocol support)
  - termination implies post-condition
  - finite resources: delays, queues
- Need standard meta-rules language
  - meta-rules may involve external events
  - meta-rules to prevent agent interference

Questions