OMG Production Rule Representation - an Overview

Presentation to W3C Rule Interoperability Workshop
April 2005
Agenda

Background to OMG, PRR & PRR Team
PRR Definition & Status
Potential Role in Rule Interoperability
What is OMG?

- Standards body most known for UML and CORBA
- Vendor + domain organization membership
  - Task Forces & Special Interest Groups
- Current emphasis:
  - Framework of MDA (CIM, PIM, PSM)
  - Technologies of UML2 (xUML), MOF, XMI, …
  - Domains: Finance, Space, Telco, Defense, …
  - Rule-specific: BEI (BSBR, PRR)
  - Rule-related: BPDM, ODM, OCL
What is PRR?

• Metamodel + associated representation for the class of rules typically used in production rule engines & others
  – If \(<Conditions>\) then \(<Actions>\)
  – Organized by ruleset
  – 2 subtypes considered (for 1st PRR version):
    • Sequentially processed procedural rules
    • Forward chaining inference rules (Rete-model)
  – Often used to represent business rules of various types in BRMS
• Interchange for rule modeling via XMI
• Make production rules 1st class citizens in UML
PRR positioning in MDA

- Business Motivation
- Semantics for Vocabulary & Business Rules
- Business Object Model
- Object Model
- Production Rule Representation
- Business Process Definition Metamodel
- Ontology Definition Metamodel
- Procedural code
- Query languages
- Blaze SRL
- Ilog IRL
- Pega rules

Execution layer
Why is a PRR standard required?

• Multiple representations used by industry for the same concept
  – Production Rules in rule engines / BRMS
    (eg Fair Isaac, ILOG, CA, Pegasystems…)
  – Production Rules in process engines / CASE tools
    (eg IBM, Fujitsu…)
  – Supporting technologies (eg LibRT)

• Industry need to align the “business rules approach” with UML-based OO software development best practices

• Co-development with proposed PRRuleML
Rule Model

Variables in rules can map to collections / classes

ProductionRule

Binding

Condition (part)

Action (part)

OCL expressions for conditions, actions

1

1

1

1

1..*

{ordered}

Binding is the (JRules: variable; Blaze: pattern) named entity that provides the definition of the result tuple passed from condition part to action part of an inference rule.

Condition(s) provide a filter (constraints) against the bindings to define which entities will have actions made against them. Note these are not modelled as separate conditions as this belongs to a lower level logical representation (not unique to PRR and should be re-used from elsewhere).

Actions are processed for each tuple returned by the filter part of a rule for bindings.

Variables in rules can map to collections / classes
State of PRR

• Basic metamodel defined (rulesets + rules):
  – Generic: structure compatible with
    • Multiple rule types:
      fwd / bwd chaining, sequential…
    • Multiple expression representations:
      XPath / Java / ECMAScript / …
  – Core: OMG modeling specific model
    • OCL-based expressions for fwd chaining / Rete and sequential rules

• Examples library

• Examination of OCL / OCLExpressions to define bindings + conditions + actions
Proposition: an equivalent to PRR, possibly a concrete syntax for PRR, is required for run-time rule interchange

1. PRR is for rule modeling
   • Context: OMG UML / commercial BRMS
   • Aligns with current commercial software development practices / technologies

2. PRR only loosely “related” to formal logic
   • Rule execution results in state changes
   • No backtracking semantics
   • Defines behavior, NOT a generic KRL

3. PRR works beyond web