Knowledge Integration Environment

Aka “Knowledge is Everything”

D.Sottara, PhD
OMG Technical Meeting – Spring 2013, Reston, VA
Part I – The Consolidated Past: Drools 5.x

- **Drools Expert**
  - Object-Oriented, Production Rule engine
- **Drools Flow**
  - Complex Event Processing engine
- **JBPM**
  - BPMNv2-compliant process engine
- **Decision Tables**
  - Waiting for DMN
- **Drools Planner**
  - Metaheuristics Constraint Optimization
- **Guvnor**
  - Repository and Authoring BRMS
Part II – The Upcoming Future: Drools 6.x

- KIE
  - Knowledge Integration and Execution Environment
- Drools – PMML
  - Predictive Analytics
- Drools Chance
  - Fuzzy and other “non-boolean” reasoning styles
- Drools Shapes
  - Semantic Web Technology integration
- Drools MAS
  - FIPA-compliant agent implementation
Knowledge Bundles
Knowledge Bases are Composite:

- Rule Engine
- Temporal Extension
- Process Engine
- Optimization Engine

Facts (Objects)

Process Instances

Events

Rules

Business Processes

Decision Tables

Constraints

Rule Engine
+ Temporal Extension
Process Engine
Optimization Engine
• Knowledge Bundle: package of Knowledge Assets
  - “virtual directories” by package
• Domain models (Classes + Interfaces)
• Business Processes
• Decision Tables
• ...
  - Distributed with G.A.V. Indentifiers
  - Deployed (“injected”) in a runtime environment
    • Running the appropriate engines
API4KB (work in progress)
API4KB : Drools binding (to date)
There's **much** more than Rules and Processes

Just focusing on rule languages:
- RIF + dialects
- RuleML
  - Deliberation RuleML
  - Reaction RuleML
- SBVR
- ...

- Variations:
  - *Modal, Epistemic, Fuzzy, Deontic, Defeasible,* ....
Drools Shapes
Ontologies

- Ontologies for knowledge representation and reasoning:
  - Domain Description
  - Inference
    - Classification
    - Querying
    - ...
  - Persist / Retrieve Information

Problem: An ontology KB is not object oriented...
  - Drools as a triple processor? Not convincing...
Domain Models

- Step I: Ontology → (static) domain model
  - **OO model**
  - **Classes + Interfaces**
    - Deals with multiple inheritance
    - Separates type from implementation
Drools Shapes – Model generator

- Ontology (abstract model)
- Interfaces (src + jar)
- ... (additional interfaces)
- XSD (schema)
- XJB (bindings)
- HyperJaxB III
- JAVA (classes)
  - JAXB
  - JPA
  - Empire
  - XML
  - DB
  - TS
Rule / Ontology integration

• Step II: Semantic Reasoning
  - Classification / Subsumption (T-box reasoning)
    • PoC
  - Classification / Recognition (A-box reasoning)
    • Rule style (CWA, UNA): work in progress
    • Ontology style (OWA): PoC
Rule / Ontology integration

- Step III: Rule Integration
  - Ontology-grounded rules
    - Write rules using interfaces
    - Ontology defines the rule vocabulary
  - Strong + Dynamic Typing
    - “Traitng”
      - Proxy-assisted interface injection
      - Transparency
“Traitig”

- Instance-level operation:
  - Add Type information
  - Apply Restrictions
  - Apply Extension

```java
declare Person
    name : String
    age : int
end

declare trait Patient
    name : String
    mrn : String
end

when
    $p : Person( ... )
    ...
then
don( $p, Patient.class )
end
```

Demo...

Drools PMML
AI Debate

- **Symbolic AI**
  - Based on an explicit, *qualitative* representation of knowledge (rules, processes, ontologies, ...)
  - *Shared* and “*Taught*”

- **Sub-symbolic AI**
  - Based on an implicit, quantitative representation
  - *Mined* and “*Learned*”

Which one is used for:

- **Classification** ?
- **Inference** (prediction) ?
Predictive Models

- Parametric, connectionists models
  - General structure is predefined
  - Parameters encode quantitative knowledge
  - Infer *quantitative* knowledge
    - Neural Networks
    - Decision Trees
    - Support Vector Machines
    - Regression Models
    - Association Rules
    - Clusters
    - ...
PMML

- DMG Standard for predictive model interchange
  - XML Based
  - Supported by major data mining tools
- Predictive Models as knowledge assets
  - Data Pre-processing
  - Model structure and behavior → evaluation
  - Result Post-processing
Drools PMML

- Integrates predictive models into knowledge runtime
  - PMML-encoded model
    - → Rule Set (behavior)
    - → Facts (model structure)
  - Chaining-driven integration:
    - Model I/O bound to object field(s) – or – event streams
    - Additional helper facts provide additional information
  - Homogeneous structure:
    - Control Meta-rules can modify the “predictive model”
• Demo : Neural Network evaluation

https://github.com/droolsjbpm/drools-chance/tree/master/drools-pmml/src/test/java/org/drools/pmml/pmml_4_1
Drools Chance
A simple rule

• A rule reads:
"If a young Patient has an abnormal XYZ Test, then they might be diagnosed with Problem ABC"

• when
  a Person has the status of Patient, has age < 18, and there exists a Test with type XYZ and value > x%
then
insert a diagnosis of Problem, type ABC

• when
  a Person isA Patient (whose) age is YOUNG, and there exists an XYZTest and (its) value is ABNORMAL
then
insert a Problem with type ABC that isA ABCProblem
Imperfection

- Imperfection
  
  "... is a condition where Boolean truth values are unknown, unknowable, or inapplicable . . ."

  (W3C Incubator Group for Uncertainty on the Web)

- Uncertainty
- Vagueness
- Imprecision
Drools Chance

- **Express** Imperfect Rules
- **Match** Imperfect Data
- **Extend** the default, "perfect" behavior

- Data Properties: Distributions $<Value \rightarrow Degree>$
- Rule Evaluators: Compute Degrees
- Logic Operators: Combine Degrees

```
Person

name
"john" 0.6 0.4 "mark"

"Crisp"

age
32
```

Distributions:
- Domain<String>
- Domain<Integer>
Drools MAS
Agent-oriented approach

- “Self-managed” knowledge runtime
  - Application-controlled →  
    - The application owns and controls the runtime
  - Service →  
    - Independent, but “bare” runtime
  - Agent →  
    - Filtered access  
    - Mediated by a communication layer
Agent architecture

- External communication layer: **FIPA**
  - ACL performatives extend working memory operations
    - Inform
      - Confirm / Disconfirm
    - Query
    - Request
    - ...
  - OWL → OO conversion for message content
    - XML compatibility: message serialization
Agent Architecture

- “Main” working memory for communication
  - Parallel conversations
  - Message interpretation
- “Mindsets” for reasoning
  - Dedicated slave working memories
  - Content is Application-specific
    - Content-based routing applied internally
  - Created/Destroyed as needed
- Examples:
  https://github.com/droolsjbpm/drools-mas/tree/master/examples
Integration
Hybrid Reasoning ??

value ABNORMAL

patient OLD

Collect Historic Data → Predict → Request Ack → Email

ack
Hybrid Reasoning!!
Integration

Knowledge Base

Extended KB

Native KB

Data Model (OO) + Facts

Rule Engine

Process Engine

DRL

Data

Working Memory

Session

APIs (Java)

Agents (DRL)

OWL2

PMML

DT

RuleML

XSD

PRR

RIF

Fuzzy RuleML

"Imperfect" DRL

BPMN2

DRL

Rule Engine

CEP Engine

Session