Rethinking JBoss
Business Rules for High
Performance Systems

A Lesson in Business Rule Processing For Rule-Based Architectures in The DoD

John Fullard
Matt Fisher

Progeny Systems

June 12, 2013
It Started with An Idea...

- Work with US Naval Community to share data
  - Transparently
  - Easily
  - Ubiquitously
- Open Source / Open Architecture technologies
- Reduce risk, cost, effort over P2P solutions
  - Certification costs especially!
It Grew With Collaboration, Experimentation

- Support multiple protocols, data models, formats
  - DDS, AMQP, CORBA, IDL, XML, XSD, ...
- Don’t reinvent the wheel every time
- Use JBoss ESB/SOA-P architecture
  - Service-based
  - Large community
  - Ability to automate
  - Flexible
Resulted in Universal Gateway (UGW)

- Dual deployment
- Drools rules were critical
  - Freedom from code changes
  - Dynamically modifiable
  - Extensible with java
- Data exchange based on message type, message content, ...

Capabilities:
- Data Mediation
- Protocol Mediation
- Data Filtering

JBoss ESB
Universal Gateway
Drools Implementation

- Design approach
  - Rule requirements
  - Environment
  - Guvnor templates
- Lessons learned
- Custom rules editor
Rule Requirements

- Message filtering
- DDS topic
- Track or resource type
- Message age
- Distance from ownship
- Message rate decimation
- Content-based message tagging
**Gateway Architecture**

- JBoss ESB/SOA-P
- Bidirectional DDS gateways and routers
- JMS messaging on one side of gateway
- Data mediation DDS <-> XML
- Message updates depending on rules firing
- Data-type agnostic processing in rules engine

```xml
<service name="DM_PurgeSystemTrackThirdPartyData" invmScope="GLOBAL" category="Drools_TDM_CStoC2" description="Dr">
  - <actions mep="OneWay">
    - <action name="BRP" class="org.jboss.soa.esb.actions.BusinessRulesProcessor">
      <property name="ruleAgentProperties" value="/home/sysadm/drools_config/ruleAgentTdmCsC2.properties"/>
      <property name="ruleClockType" value="REALTIME"/>
      <property name="stateful" value="false"/>
    - <property name="object-paths">
      <object-path esb="body.payload"/>
    </property>
    </action>
  </actions>
</service>
```
Rules Overview

- Update decimation rate
- Update ownership
- Decimation
- Message type filter
- Track type filter
- Geo filter
- Time filter
- Send message
  Application of GUID/ICISM*

*GUID/ICISM on single Drools instantiation only
**Salience**

- Used to enforce rule execution order from top to bottom
- Higher salience values in filter rules than in send rule
- Tested using jBPM Ruleflow but a bug in Drools/jBPM caused stateful sessions to leak memory
  - Bugzilla 724787 in Drools 5.2.0 and jBPM 5.1.0
  - Fixed in Drools 5.3.0
State Maintenance

- Ownership position and decimation rates updated and maintained
  - Ownership position received from certain DDS messages
  - Decimation rates set in Guvnor rules
- Original design leveraged Stateful sessions
  - Created/Updated facts for storing these values
  - Issues with rules refiring when updated facts matched rules again
Hybrid Approach

- Final design leveraged stateless sessions
- Custom state maintenance code implemented
  - Rules engine accessed singleton objects for position and decimation rates
  - Eliminated need for additional facts
  - Facts inserted or updated within Stateless Session are not reevaluated by the rules
**Guvnor Rule Templates**

- Write template for one rule
- Import values from table and generate DRL
- Allows modification of rules logic without editing DRL
- DRL is generated automatically in Guvnor
- Provides customer with fast turnaround for evolving data model
  - New rules can be added without code changes
  - Can be completed in field by non-technical user
Guvnor Rule Templates

Example of DDS Topic filter, which allows user to check and uncheck desired Topics.

The values in each row are plugged into the template to create a rule. Adding a rule is as simple as adding a row.
Drools provides powerful Complex Event Processing (CEP)

Initial design used sliding time windows for rate decimation

Required Stateful Sessions

Could not populate cron portion of timer rule with template

```java
import org.jboss.soa.esb.message.Message;
import org.jboss.soa.esb.message.format.MessageType;
import org.jboss.soa.esb.message.format.MessageFactory;
import org.jboss.soa.esb.client.ServiceInvoker;

global Message message;
declare Pldm
    @role(event)
end

rule "decimation"
timer (cron: * 5/5 * * *)
    when
        $newest : Number()
            from accumulate( Pldm ( messageType=="RM_AIS_Status", allowResource==true, allowMessageType==true, $ts : receivedTime ) over window:time($n),
                        max ( $ts ) )
    then
        $pldm = Pldm ( $ts==receivedTime )
        Message esbMessage = MessageFactory.getInstance().getMessage();
        esbMessage.getBody().add("pldm", new getFullMsg());
        ServiceInvoker serviceInvoker = new ServiceInvoker("RM_AIS_Status", "RM_AIS_StatusRouter");
        serviceInvoker.deliverAsync(esbMessage);
end
```

Template rules cannot populate these values.
Universal Gateway Drools Implementation

- Design approach
- Rule requirements
- Environment
- Guvnor templates
- Lessons learned
- Custom rules editor
Stateful Sessions

- Rules were reworked so that each would fire only once and the relative order did not matter
  - Rules cannot refire in stateless session
  - All rules would fire, but any rule resulting in a drop was given higher priority than the rule to send a message
  - Relative order of those filter and decimation rules is not important, only the priority of drop versus send
- Final ruleset eliminated message drops and duplication
  - Original design allowed rule refiring which resulted in numerous tracks with either multiple or no GUID and ICISM tags applied
  - Updates to the message body would occur after all filter rules had fired
  - Improved CPU utilization by only applying GUID and ICISM to messages that had already passed all filters
Guvnor Data Model Issue

- Guvnor requires data model jars for building rule package
- When jar is uploaded, list of imported types is populated
- Could not easily verify when Data Model is successfully loaded in Drools
- Data model with ~7000 classes was updated frequently
  - GUI became unresponsive loading list of types
  - Building rule package took minutes instead of seconds
- Clearing list of imported types removed these problems, but was tedious to do repeatedly
Wrapper Classes

- Complex data model of ~7000 classes was difficult for Guvnor to support
  - Simplifying data model was not an option
- Solution involved development of partially-visible wrapper classes
  - Wrapper classes hid underlying data model from Guvnor
- Any necessary parsing done before object enters Rules Engine
  - Provided the accessor methods required by Drools, which underlying model did not
  - No XML transformers needed
  - Fewer rules needed because all incoming protocols and data models had same internal representation

```java
public TrackMessage(
    DDS.TrackMessages.CEC_IdentificationRecommendationData trackMsg,
    String topicName) {

    messageType = topicName;
    trackType = trackMsg.sourceTrackKey.trackType.name();
    sentTime = trackMsg.sentTime.nanosecondsSinceEpoch;
    index = trackMsg.sourceTrackKey.index;
    useCount = trackMsg.sourceTrackKey.useCount;
    payload = trackMsg;
}
```
After DDS <-> XML data mediation, XML messages were between 5,000 and 175,000 bytes.

Tested using XPath to determine the type from an XML message then unmarshalled to a POJO.

Observed slow performance, due largely to the overhead of unmarshalling.

This approach was investigated for the rules engine, but the unmarshalling was found to be too expensive.

Mitigated by performing rules execution before data mediation (vs. after).
Guvnor Complexity

- Guvnor provides more capabilities than UGW requires
- Updating table values for rule templates involved numerous steps:
  - Select Package, click Business Rule Assets, open rule, click Load Template Data, edit table, click Save and Close Template Data, click Save Rule
- Added complexity meant more user training
- Additional configuration required to lock down access to certain users on the resource level
- Custom Rules Editor developed in place of Guvnor
Universal Gateway
Drools Implementation

- Design approach
- Rule requirements
- Environment
- Guvnor templates
- Lessons learned
- Custom rules editor
Custom Rules Editor

- Written in SmartGWT and modeled after Guvnor
- Limited to UGW requirements
  - Simple with focus on minimal steps
  - Stricter input validation
- Package load, save, verification, and build capabilities
# Custom Rules Editor

## C5 to C2 Tracks

### Allowed Types

- System Tracks
  - TT_SYSTEM_TRACK
  - TM_NewSystemAirTrack
  - TM_NewSystemSurfaceTrack
  - TM_NewSystemLandTrack
  - TM_NewSystemSuburfaceTrack
  - TM_NewSystemElectronicWarfare
  - TM_NewSystemAcousticContact
  - TM_NewSystemReferencePoint
  - TM_UpdateSystemAirTrack
  - TM_UpdateSystemSurfaceTrack
  - TM_UpdateSystemLandTrack
  - TM_UpdateSystemSubsurfaceTrack
  - TM_UpdateSystemElectronicWarfare
  - TM_UpdateSystemAcousticContact
  - TM_UpdateSystemReferencePoint

### 14/3S

- Select All
- Deselect All

## C2 to C5 Tracks

### Readiness
- C5 to C2 UCS
- C2 to C5 UCS

### Build Package

- Verify

### Load Package

### Save Package

## Source Tracks

- TT_ASW_TRACK
  - SM_NewASW_AcousticContact
  - SM_UpdateASW_AcousticContact
  - TT_CEC_COMPOSITERADARFF_TRACK
  - SM_NewCEC_ComposedRadarFF_Track
  - SM_UpdateCEC_ComposedRadarFF_Track
  - TT_EWS_COMPOSITE_TRACK
  - SM_NewEWS_CompositeTrack
  - SM_UpdateEWS_CompositeTrack
  - TT_MH60R_REPORTED_SOURCE_TRACK
  - SM_NewMH60R_ReportedReferencePoint
  - SM_UpdateMH60R_ReportedReferencePoint
  - TT_MH60R_SELFREPORTED_TRACK
  - SM_NewMH60R_SelfReportedTrack
  - SM_UpdateMH60R_SelfReportedTrack
  - TT_SLG32_TRACK
  - SM_NewSLG32_Track
  - SM_UpdateSLG32_Track
Custom Rules Editor

- Hierarchy of types, topics in data model reflected in editor
- Rule settings propagate through tree
- Parent nodes can show partial selection
- Stricter input validation with user notification
Drools provides high performance filtering capability
- Sustained track data rate of approximately 325 messages/sec
- Message backup would begin at 400-500 messages/sec
- New features and bug fixes released frequently
- Active Drools development community was key
  - Message boards assisted development
- Best solution involving Java, non-proprietary, robustness, performance
- Overall, integration is easy but not trivial
  - Initial CEP implementation more challenging
  - Initial template setup in Guvnor challenging
  - Drools documentation improves with each release
QUESTIONS?

jfullard@progeny.net
mfisher@progeny.net