Derrick Kittler

Solutions Architect

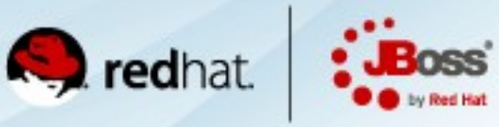
Drools and the BRMS

The Business Logic Integration Platform

redhat. | JBoss by Red Hat

Agenda

- The Concepts
- A Simple Example and Demo
- Projects Overview
- Best practices integrating Drools into **your** development process
- Learning more, getting involved and getting support



What is Drools?

Production

Drools Expert

Drools Flow (jBPM5)

Drools Fusion

Drools Guvnor

Incubation

Drools Planner

Drools Grid

Drools Semantics

Drools Chance

Business Logic integration System

redhat.

JBoss by Red Hat

What is JBoss BRMS?

JBoss Enterprise BRMS

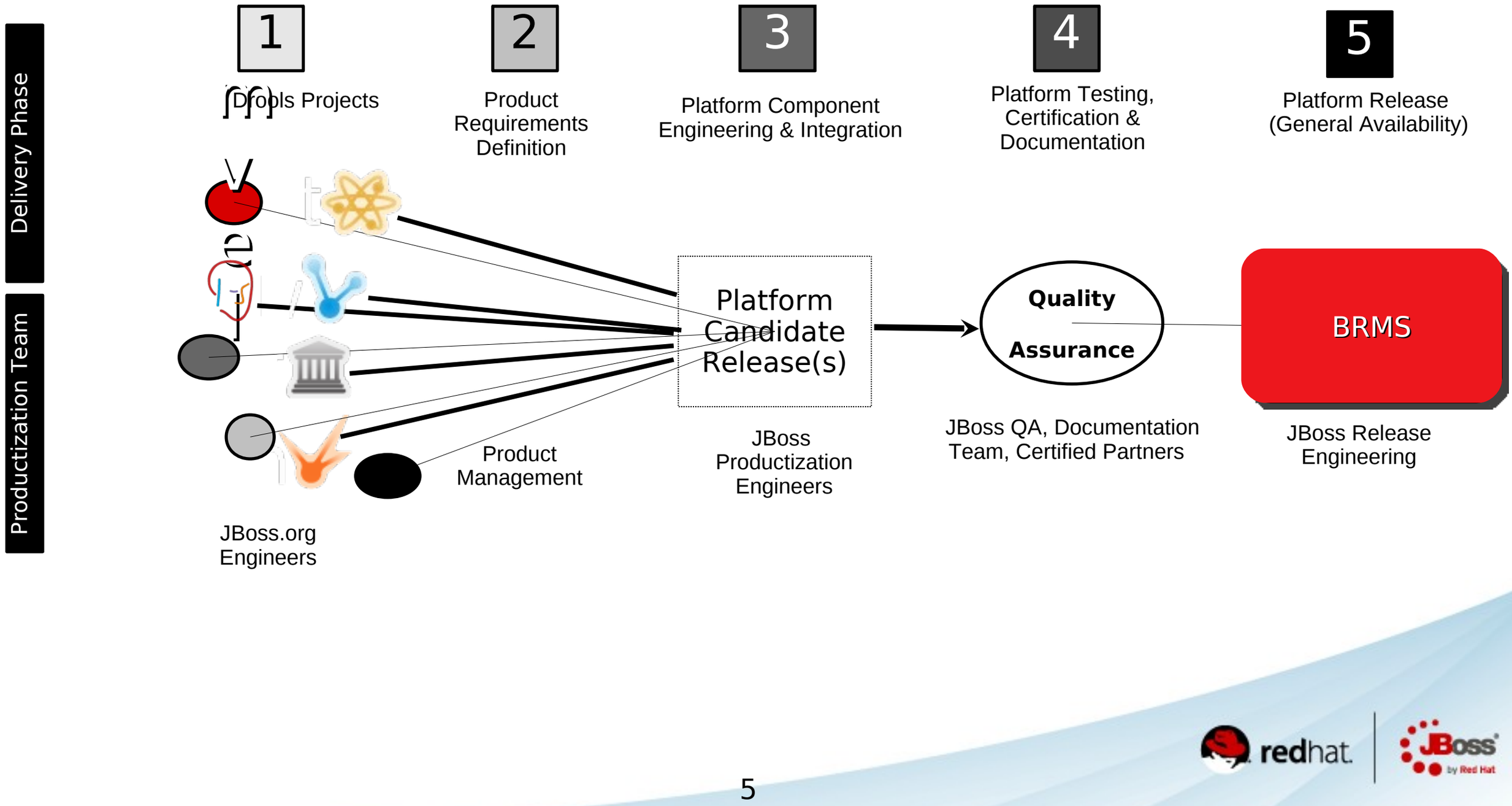
- Single, integrated, certified distributions
- Extensive Q/A Process
- Industry-leading Support
- Documentation
- Secure, Production-level Configurations
- Multi-year Errata Policy

A common and complete platform to model and govern the business logic of the enterprise

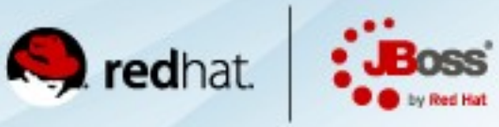
redhat. | JBoss by Red Hat

JBoss Enterprise BRMS

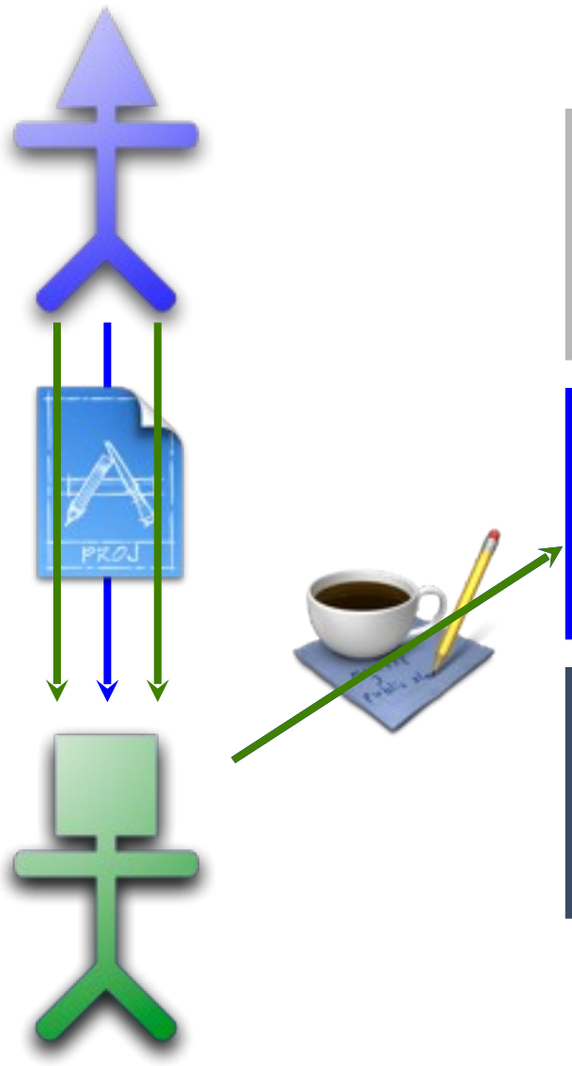
- Each JBoss Enterprise Middleware platform goes through a 5 phase delivery methodology that involves many traditional elements of the software development lifecyle:



The Concepts



The Typical SDLC



View Layer

Service Layer

Business Layer

Persistence Layer

Extract the Logic

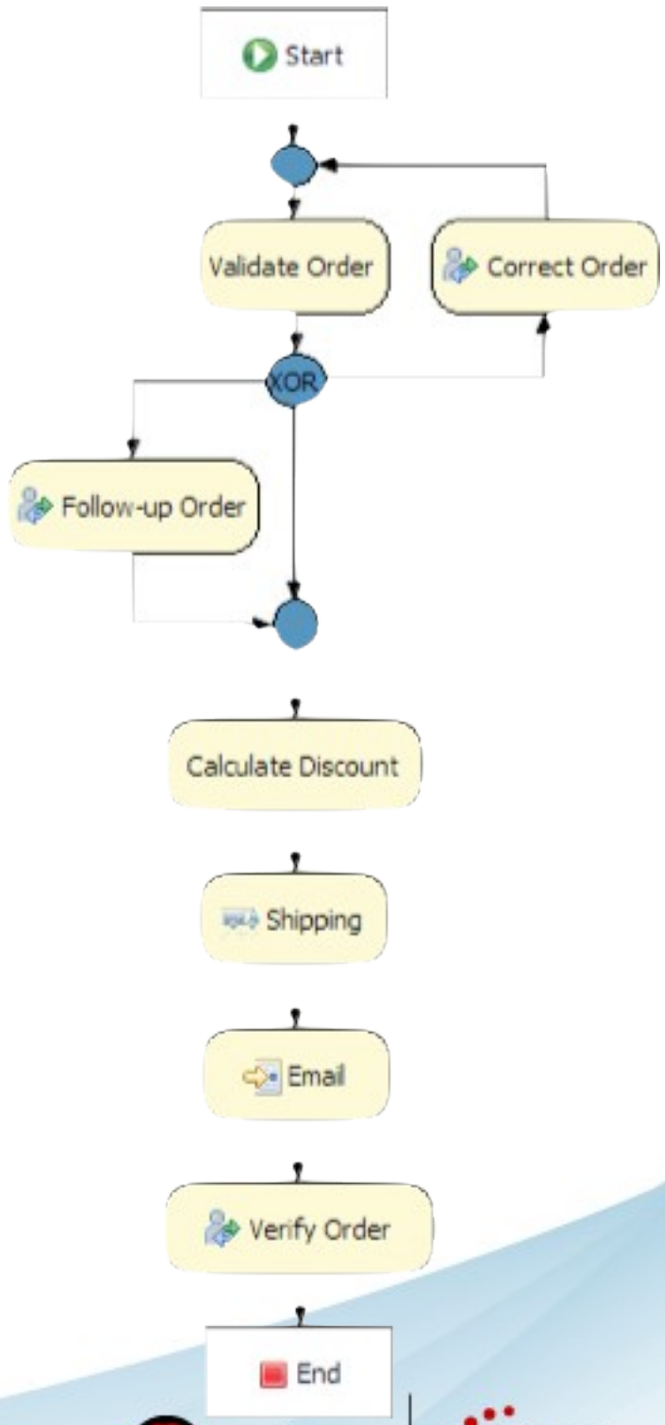
- Separate integration codebase from business logic and into a rules engine
- **Goal**: Remove the requirements churn and complexity
- **Goal**: Empower your Domain Experts to directly author rules

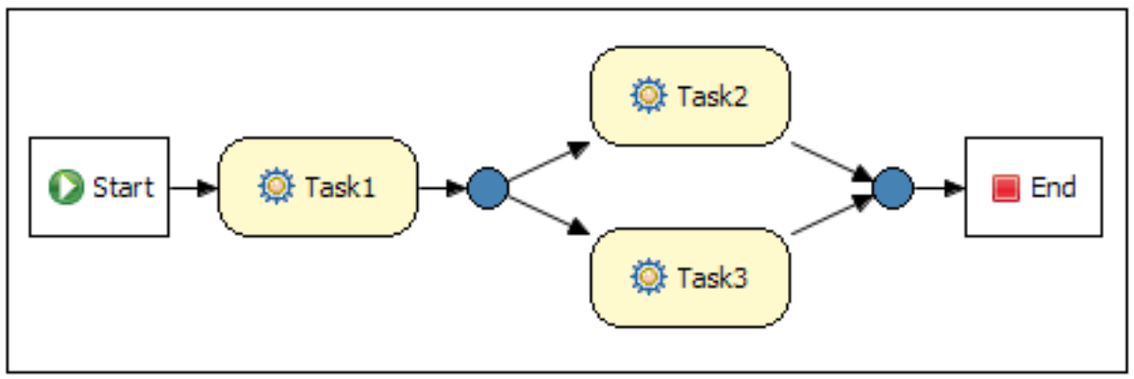
Business Logic

- Rules
- Process
- Response to Events
- Planning Problems

Why a Unified Approach?

- Extend Rules Engine to handle process State
- Extend the Engine to handle Events
- Integration provides
  - Simplicity
  - Performance
  - Manageability
  - Integration of Features






```
rule "RuleName"
    when
        // conditions
    then
        // actions
end
```

redhat. | JBoss by Red Hat

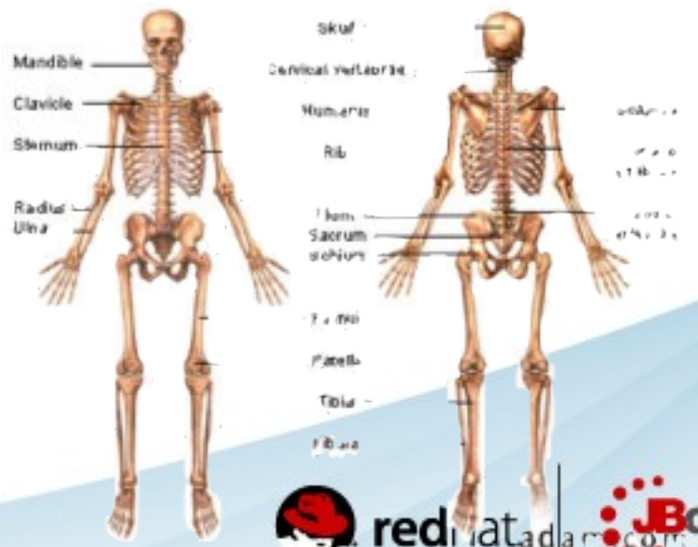
What is a Rules Engine?

- Ambiguous Term
- Inference Engine Concepts Clarify
  - Scales to a large number of rules and facts
  - Matches **facts** against **rules** to **infer** conclusions
  - Conclusions result in actions
  - Simple two-part structure:

```
rule “XYZ”
when
  <conditions>
then
    <actions>
```

Facts

- **POJOs (Plain Old Java Objects)**
  - Rules engine uses to evaluate conditions
  - Rules engine can execute POJO methods
  - Can be loaded from a database via Hibernate, JPA, etc...
  - Rules engine can modify a fact's state
- **Dynamic Facts**
  - Can be modeled in the Web UI



Fact Model Example

```
public class Room {
    private String name
    // getter and setter methods here
}
public class Sprinkler {
    private Room room;
    private boolean on;
    // getter and setter methods here
}
public class Fire {
    private Room room;
    // getter and setter methods here
}
public class Alarm {
}
```

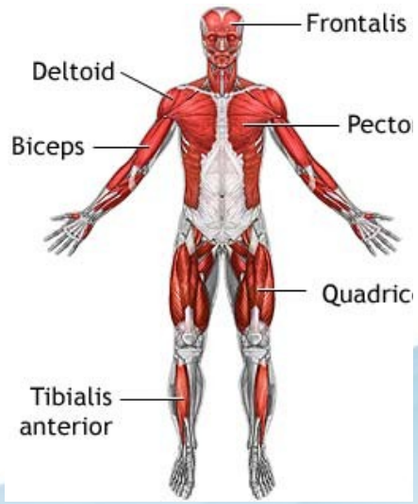
Model is multiple facts and their relationships

- Must have setters/getters
- Must have a public no-arg constructorNo mandatory inheritance/interface

Must have a public no-arg constructorNo mandatory inheritance/interface

Rules

- Form the “IF/Then” action defined as “When/Then”
- Must have a name that is unique for a rule package
- LHS (Left Hand Side)
  - Conditional part of the rule
  - Evaluate fact attributes based on criteria
- RHS (Right Hand Side)
  - Consequence or action part of the rule
  - Invoke operations
  - Modify Fact State



Rules can be authored in multiple ways!

redhat. | JBoss by Red Hat

Rules in DRL

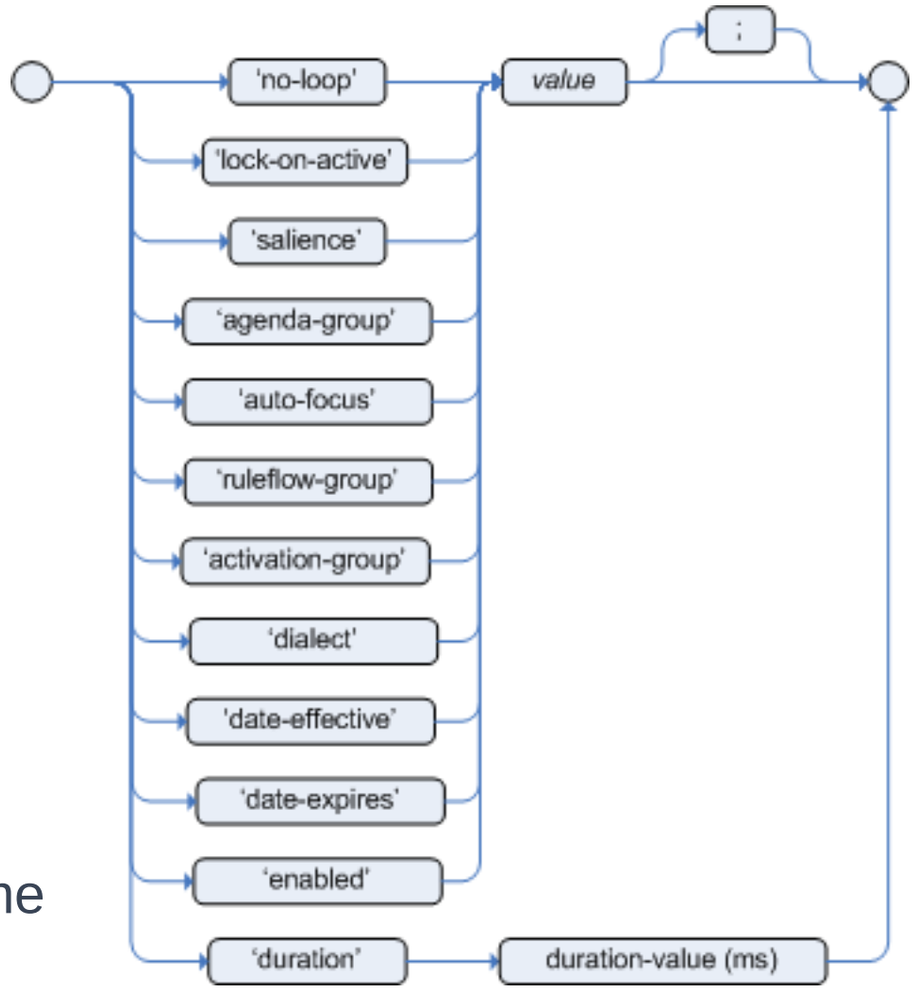
```
package com.sample

import com.sample.*;

rule "When there is a fire turn on the sprinkler"
dialect "mvel"
when Fire( $room: room)
            $sprinkler: Sprinkler( room  == $room, on == false )
then
            modify ($sprinkler) {on = true};
            println("Sprinkler activated in " + $room.name);
end

rule "When the fire is gone turn off the sprinkler"
dialect "mvel"
when
    $room : Room()
    $sprinkler : Sprinkler( room == $room, on == true )
    not Fire( room == $room )
then
    modify( $sprinkler ) { on = false };
    println( "Turn off the sprinkler for room " + $room.name );
end
```

Conditional Elements

- **and**
  - ✓ all attributes match (default of comma separated list)
- **or**
  - ✓ either attribute matches
- **eval**
  - ✓ catch all element
  - ✓ wraps any primative returning semantic code
- **not**
  - ✓ attribute does not match
- **exists**
  - ✓ checks for the existence of something
- **collect, memberOf, accumulate**
  - ✓ elements to reason over collections of data
- **from**
  - ✓ element to retreive data from external sources like DBs, WebServices, etc.
- **matches, soundslike,**
  - ✓ regular expressions and English language phonetics

Rule Control

- **no-loop**
  - ✓ short circuit rule recursion
- **salience**
  - ✓ numeric value that represents rule importance
- **agenda-group**
  - ✓ fire rules in group only when in focus
- **auto-focus**
  - ✓ trigger focus change to the rule's agenda-group
- **activation-group**
  - ✓ first matching rule fires
  - ✓ all other rules in group are ignored **lock-on-active**
- **date-effective, date-expires**
  - ✓ define rules that are only active at certain points in time
- **template**
  - ✓ define templates that may be reused in multiple rules



Avenues for Authoring Rules

Web Broswer

- Dynamic Facts
- Decision Tables
- Guided Rules
- English/Industry Specific Rules
- Guided Tests
- Scenario Tests
- Asset Search
- Browse by Category
- Browse by Status

Spread sheet

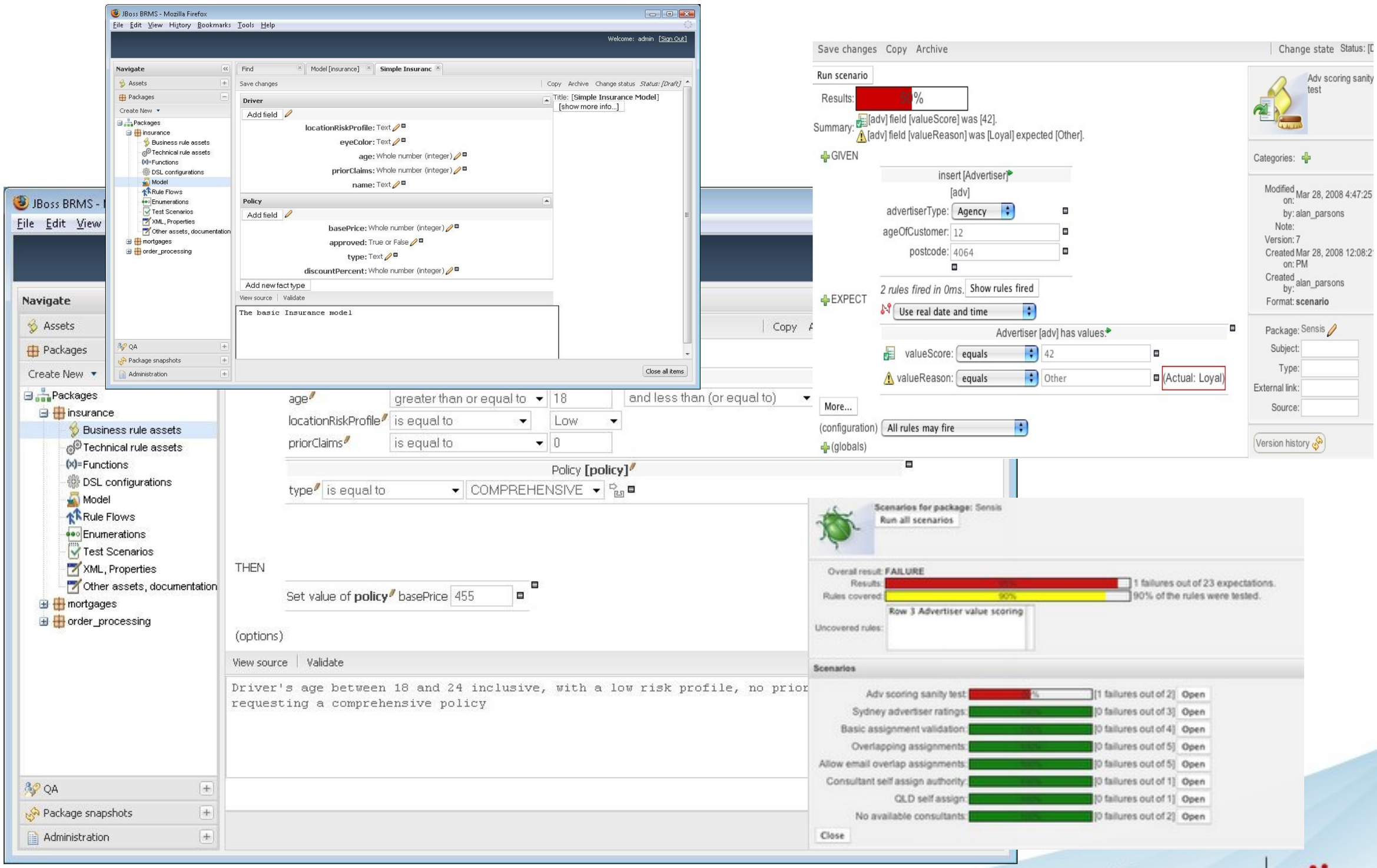
- Spreadsheet Decision Tables

JBoss Developer Studio

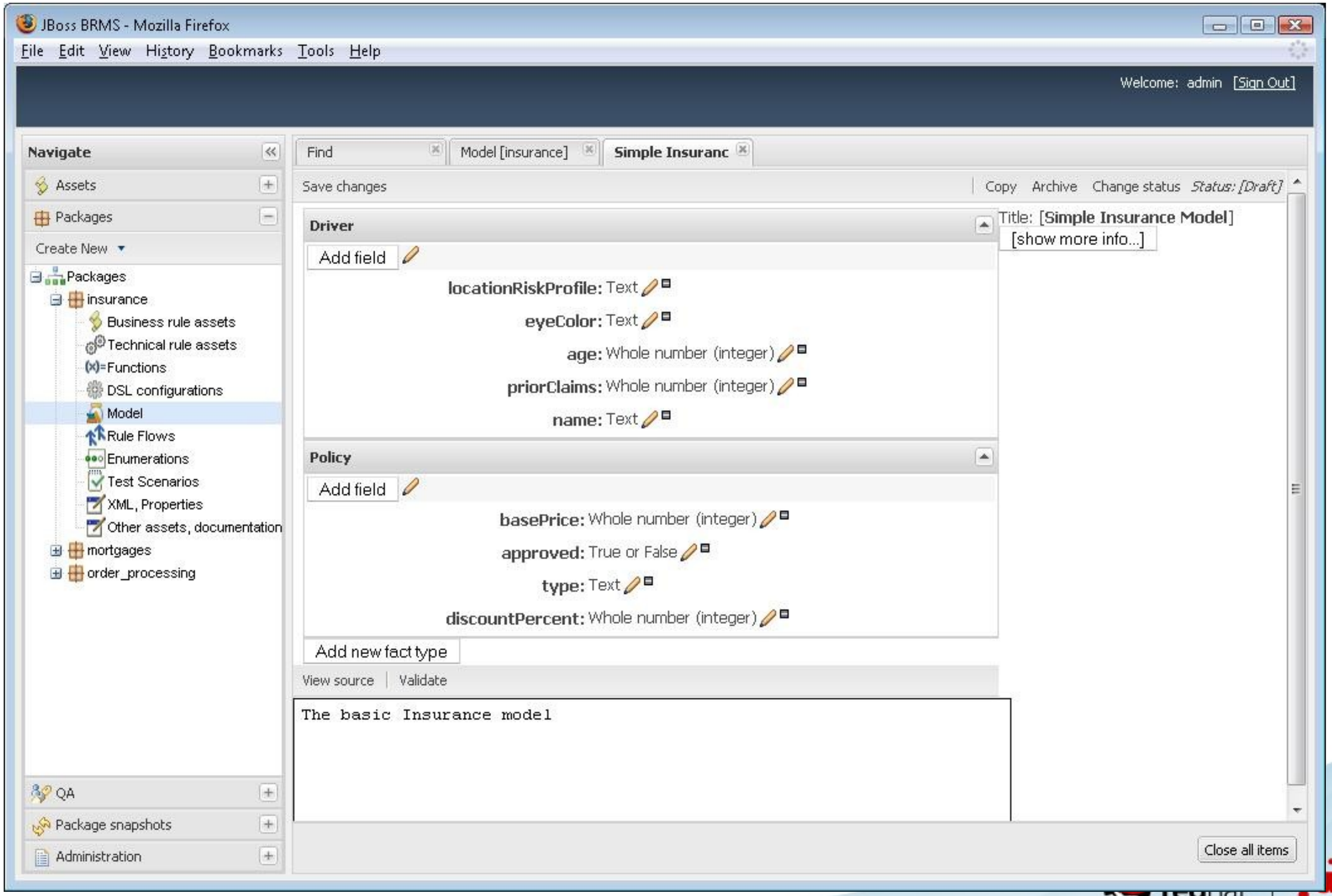
- Technical Rules (.drl)
- Enumerations
- Domain Specific Language
- Templates
- Rule Flow
- Step-debugging
- Agenda Views and Inspection
- Working Memory Inspection
- Rule Engine Audit Trails
- Technical Tests
- Technical Test Suites

redhat. | JBoss by Red Hat

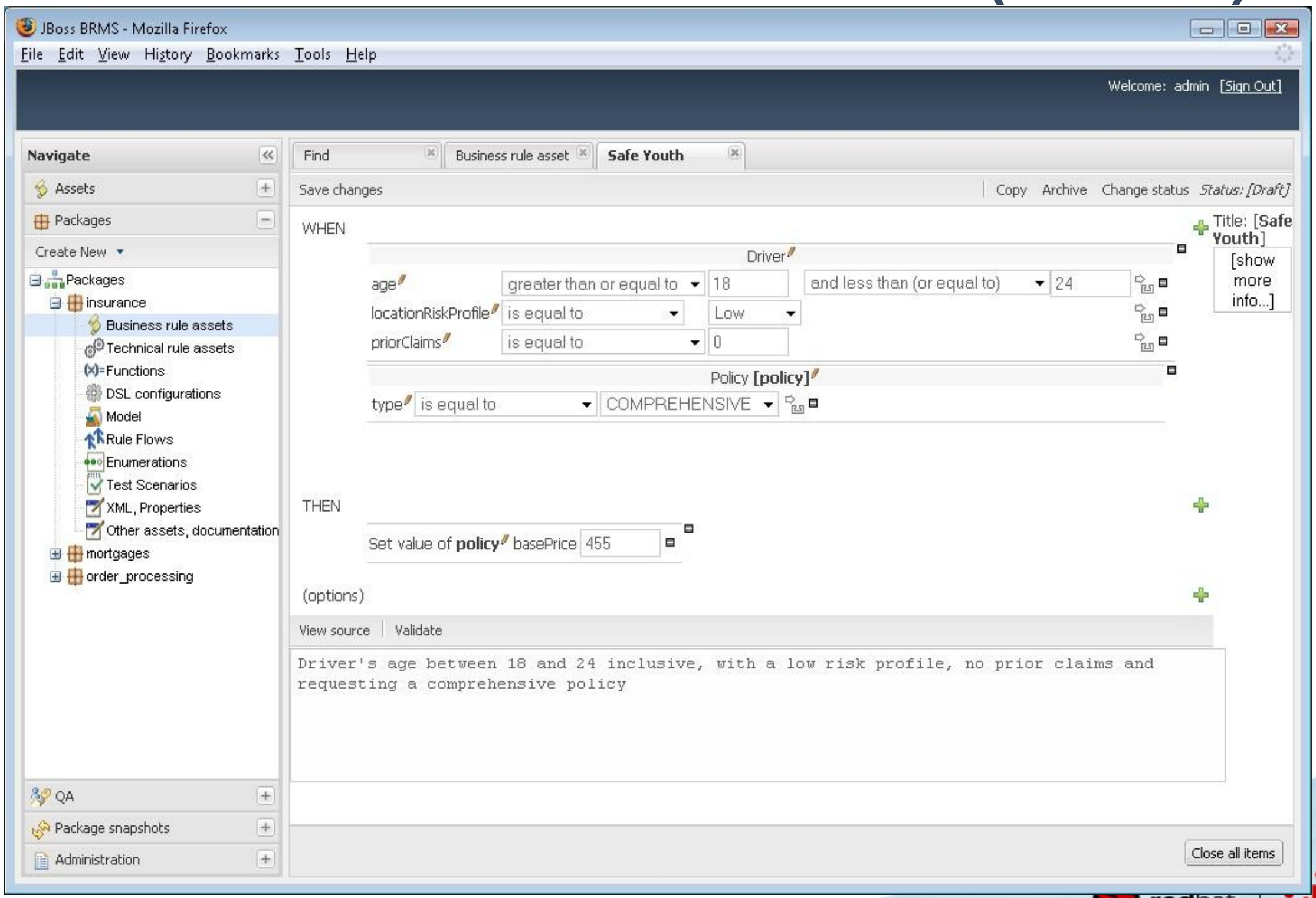
BRMS – Rich UI for Business Rules



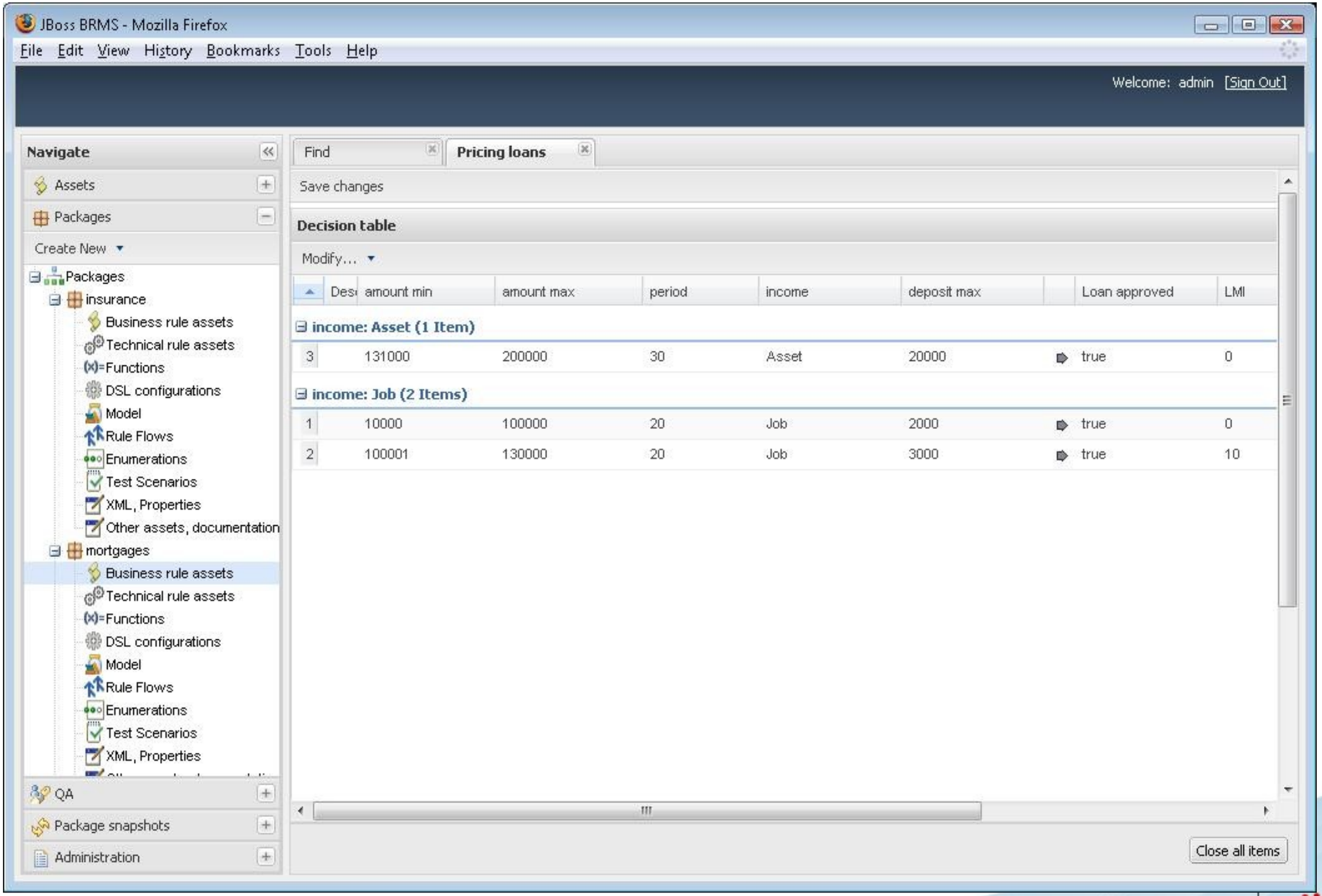
Fact Model Editor



Guided Rule Editor (Web)



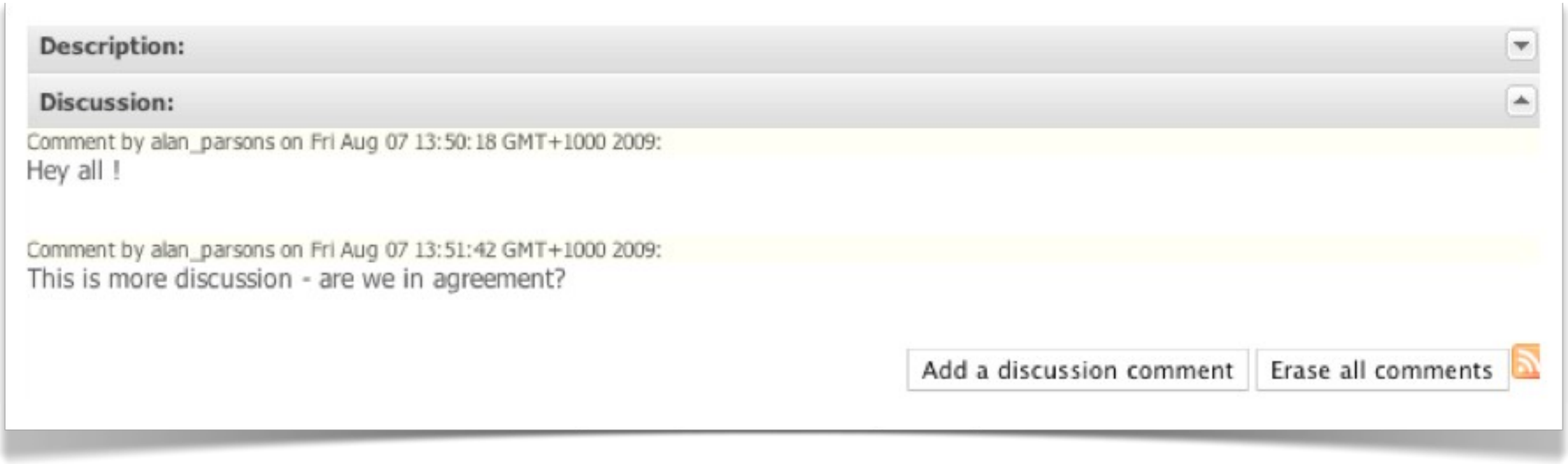
Web Decision Table Editor

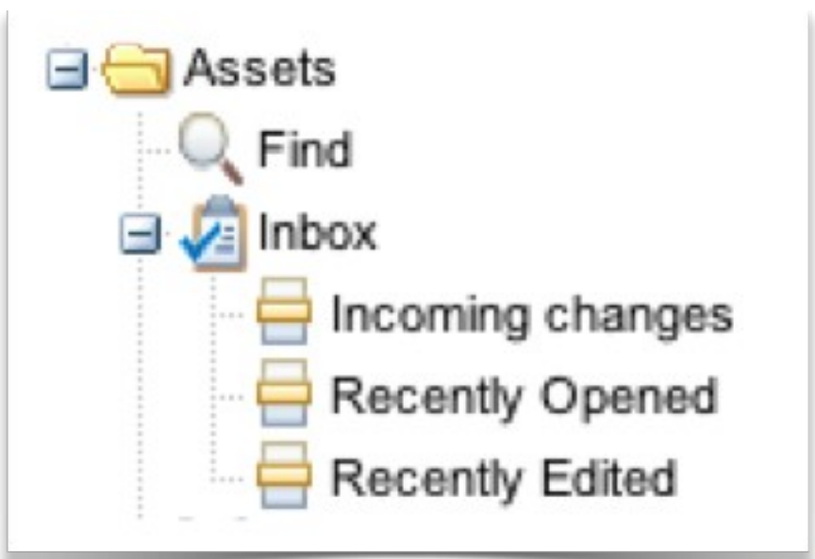


Decision Tables

	B	C	D	E	F	G	H
1							
4							
9	**Base pricing rules**	**Age Bracket**	**Location risk profile**	**Number of prior claims**	**Policy type applying for**	**Base $ AUD**	**Record Reason**
10	Young safe package	18, 24	LOW	1	COMPREHENSIVE	450	
11			MED		FIRE_THEFT	200	Priors not relevant
12			MED	0	COMPREHENSIVE	300	
13			LOW		FIRE_THEFT	150	
14			LOW	0	COMPREHENSIVE	150	Safe driver discount
15	Young risk	18,24	MED	1	COMPREHENSIVE	700	
16		18,24	HIGH	0	COMPREHENSIVE	700	Location risk
17		18,24	HIGH		FIRE_THEFT	550	Location risk
18	Mature drivers	25,30		0	COMPREHENSIVE	120	Cheapest possible
19		25,30		1	COMPREHENSIVE	300	
20		25,30		2	COMPREHENSIVE	590	
21		25,35		3	THIRD_PARTY	800	High risk

Discussions & Inbox





Information and important URLs

Last Modified: Mon Nov 29 2010 16:18:44 GMT-0600 (CST)
Last contributor: mic
Date created: Mon Nov 29 2010 16:18:37 GMT-0600 (CST)
Show package source: Show package source
URL for package documentation: http://127.0.0.1:8080/jboss-brms/org.drools.guvnor.Guvnor/package/mortgages/LATEST/documentation.pdf
URL for package source: http://127.0.0.1:8080/jboss-brms/org.drools.guvnor.Guvnor/package/mortgages/LATEST.drl
URL for package binary: http://127.0.0.1:8080/jboss-brms/org.drools.guvnor.Guvnor/package/mortgages/LATEST
URL for running tests: http://127.0.0.1:8080/jboss-br
Change Set: http://127.0.0.1:8080/jboss-b
Status:

mortgages

Rule Bankruptcy history

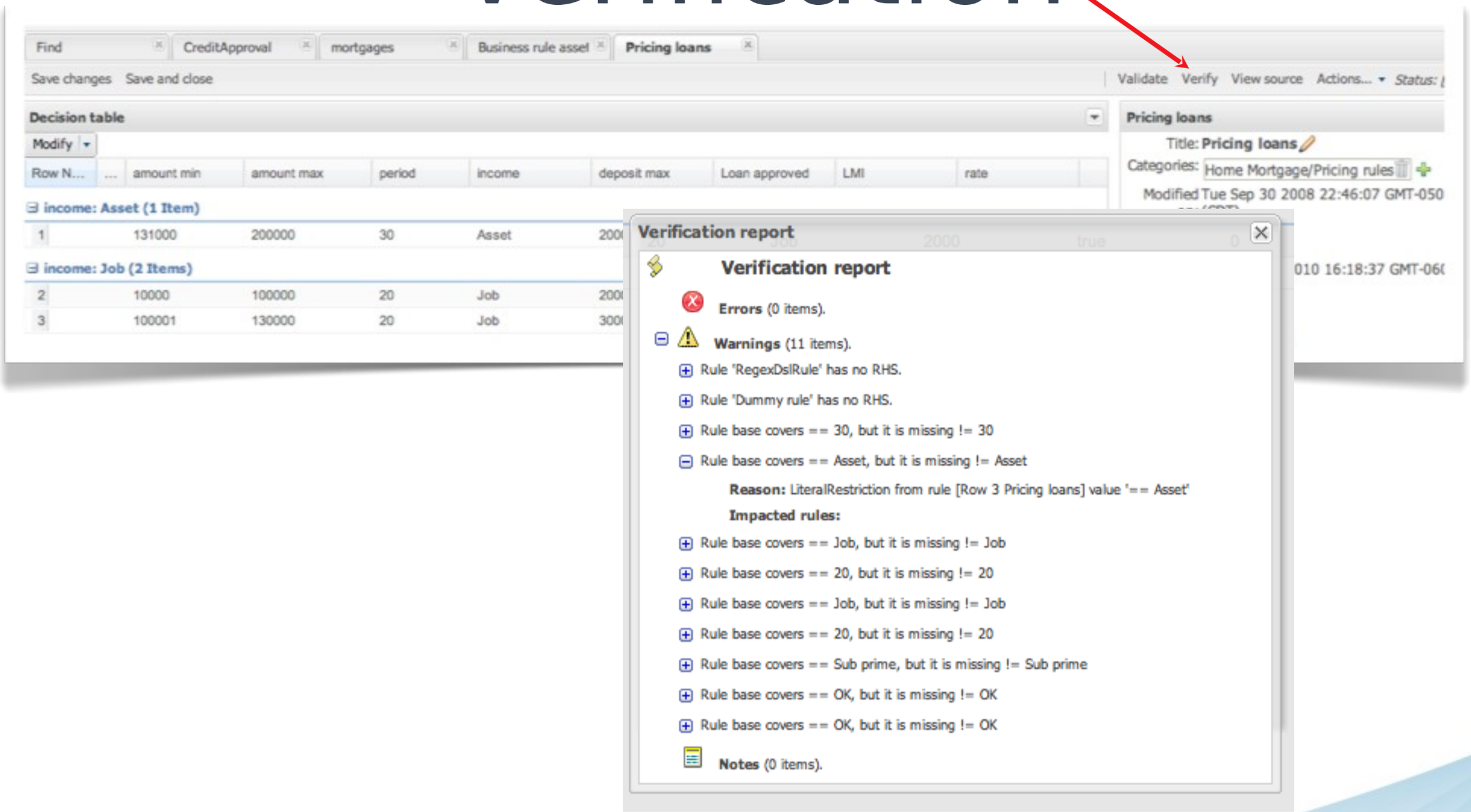
Description
-

Attributes
salience 10
dialect "mvel"
WHEN
a : LoanApplication( )
exists Bankruptcy(yearOfOccurrence > "1990" || amountOwed > "10000")
THEN
a.setApproved(false);
a.setExplanation("has been bankrupt");
retract(a);

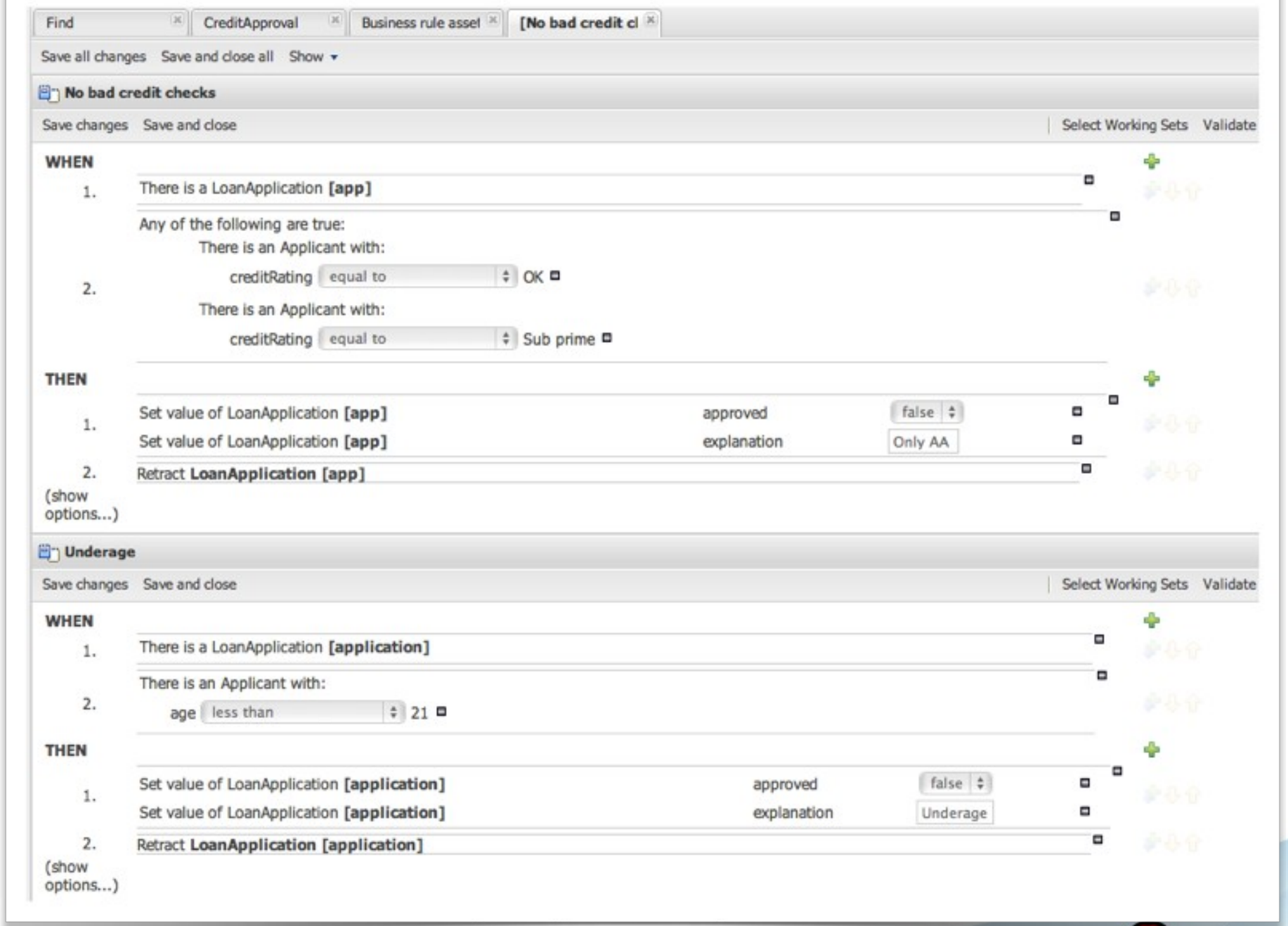
Meta Data
Creator :mic
Created date :29-Nov-2010
Last contributor :mic
Last modified :30-Sep-2008
Description :

Categories
Eligibility rules

Single Asset Verification

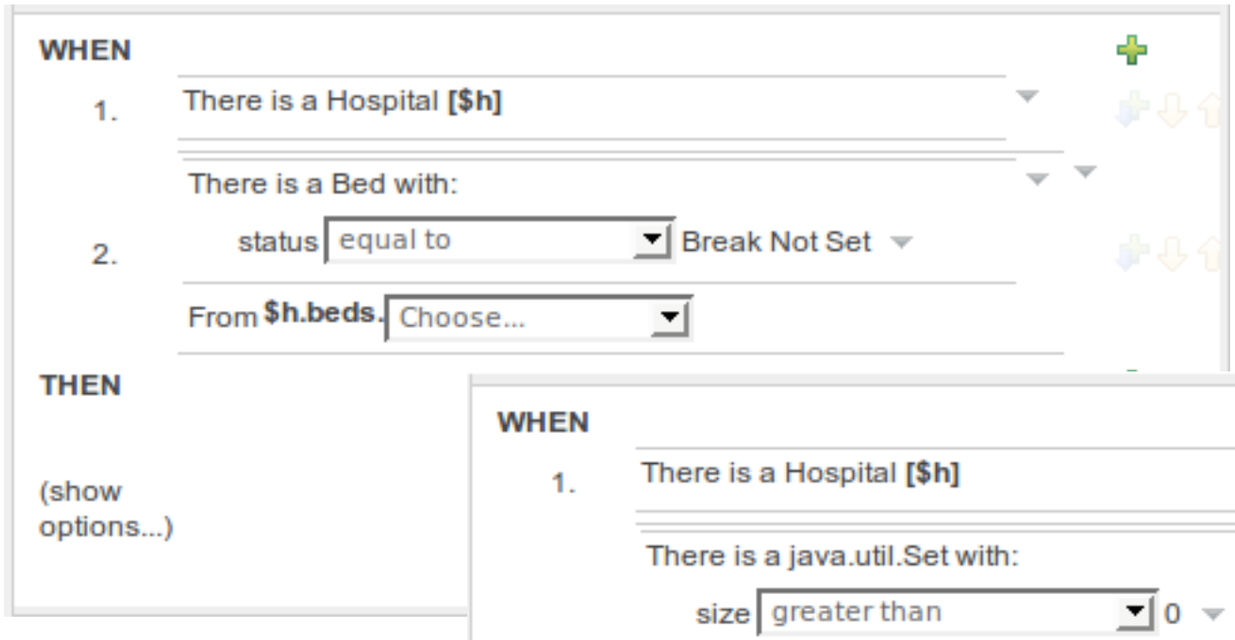


Multiple View / Edits in a Single Tab

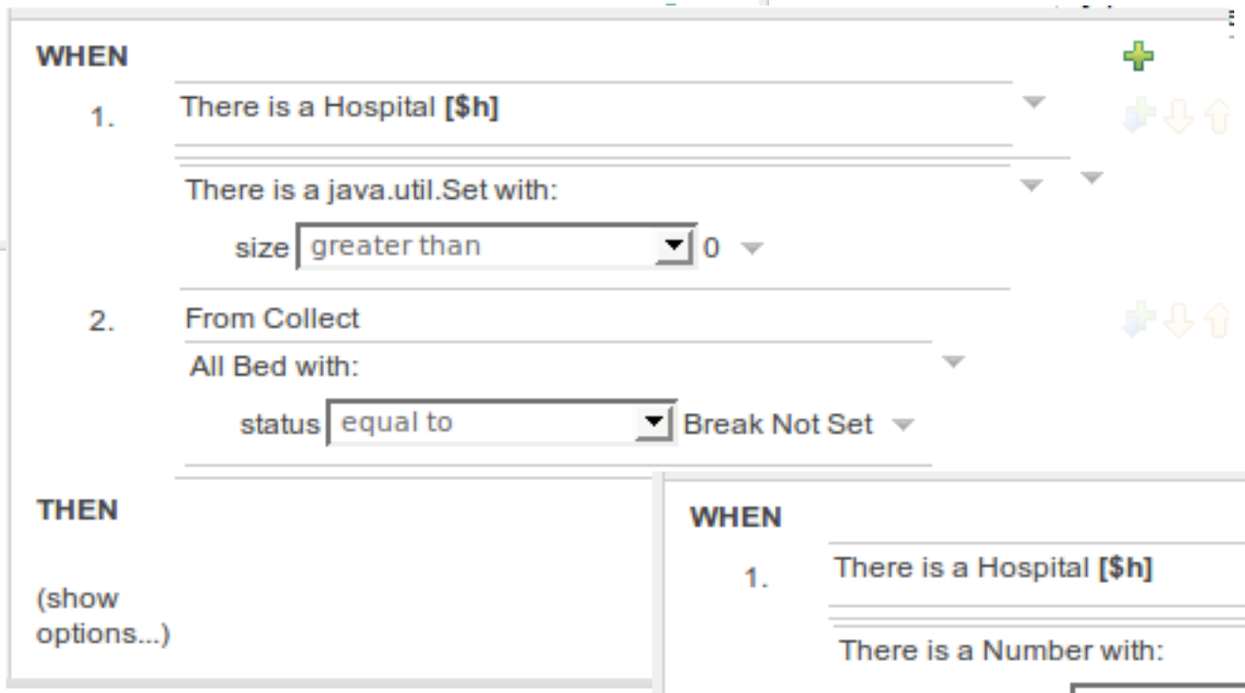


redhat. | JBoss by Red Hat

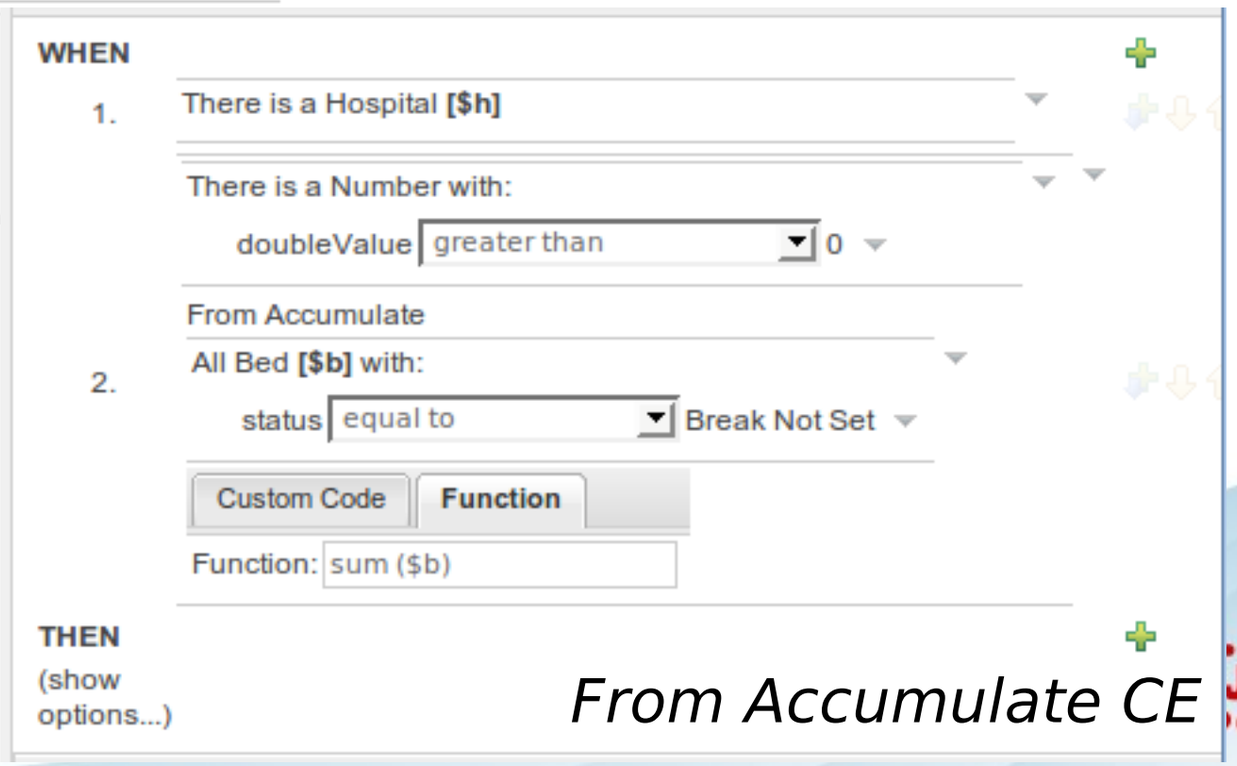
From, Collect & Accumulate Support



From CE

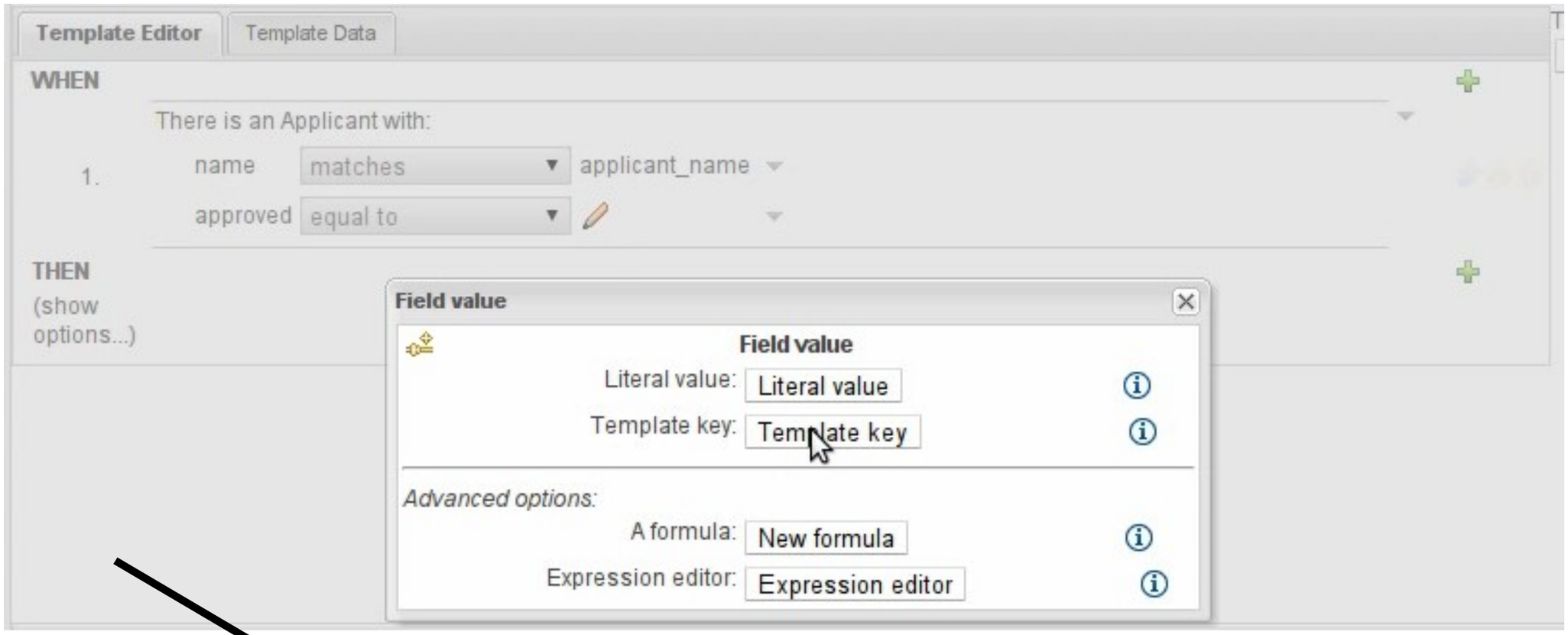


From Collect CE

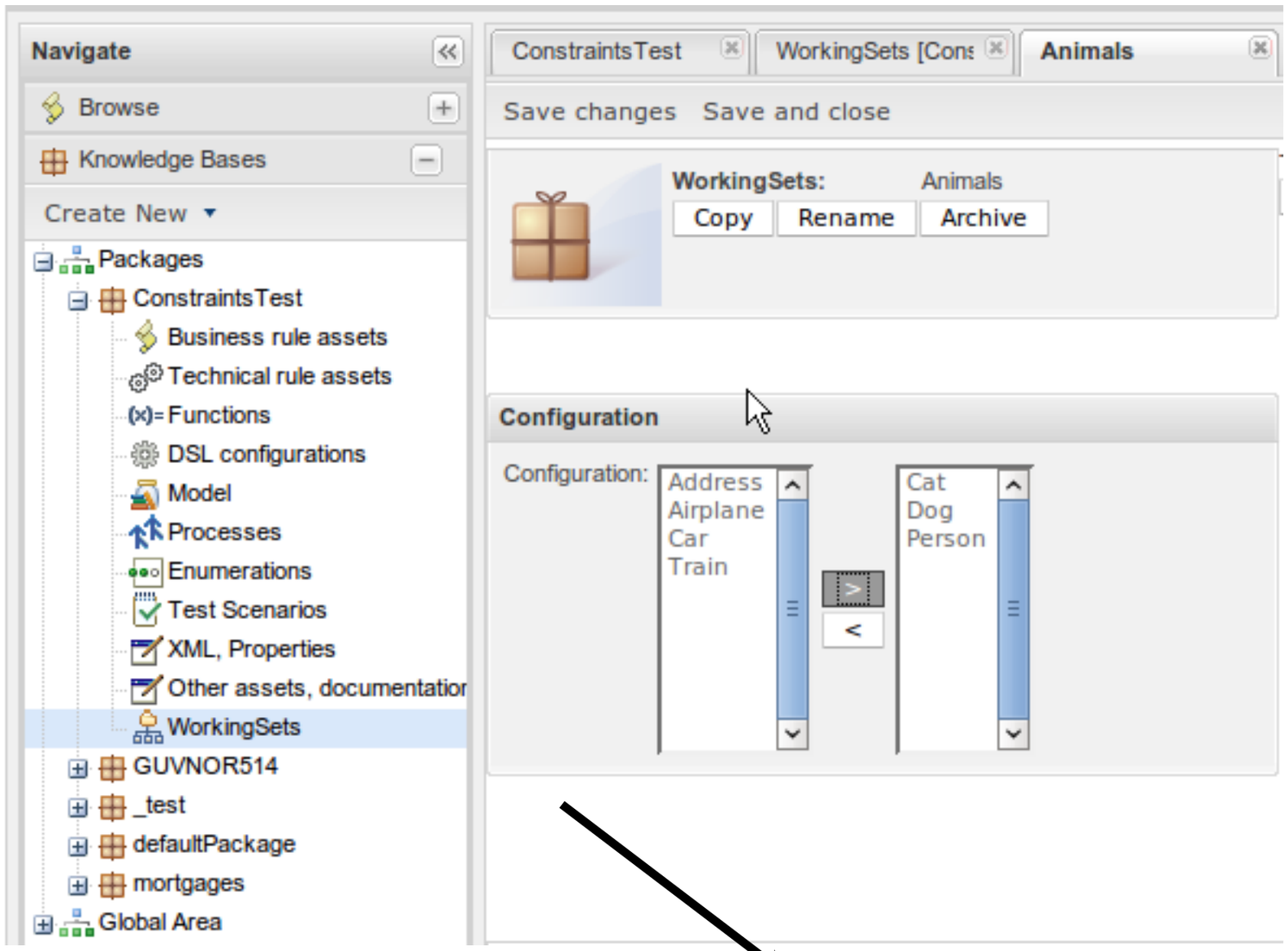


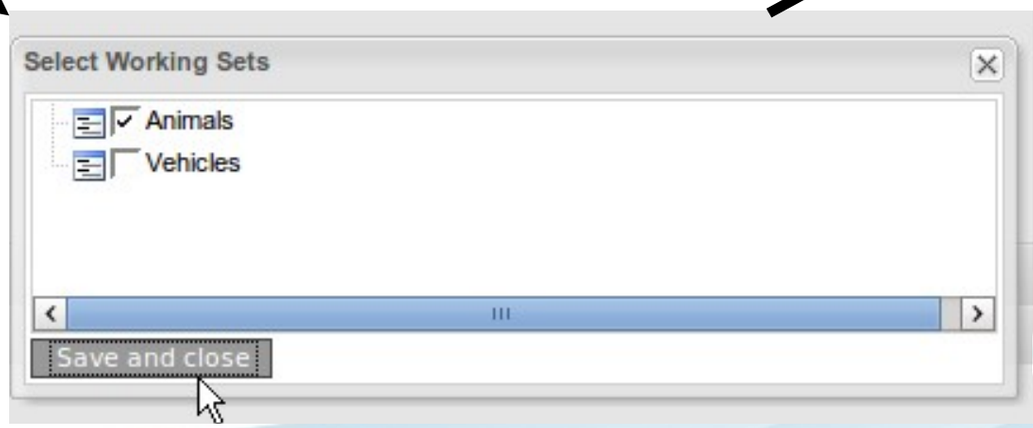
From Accumulate CE

Rule Template Support

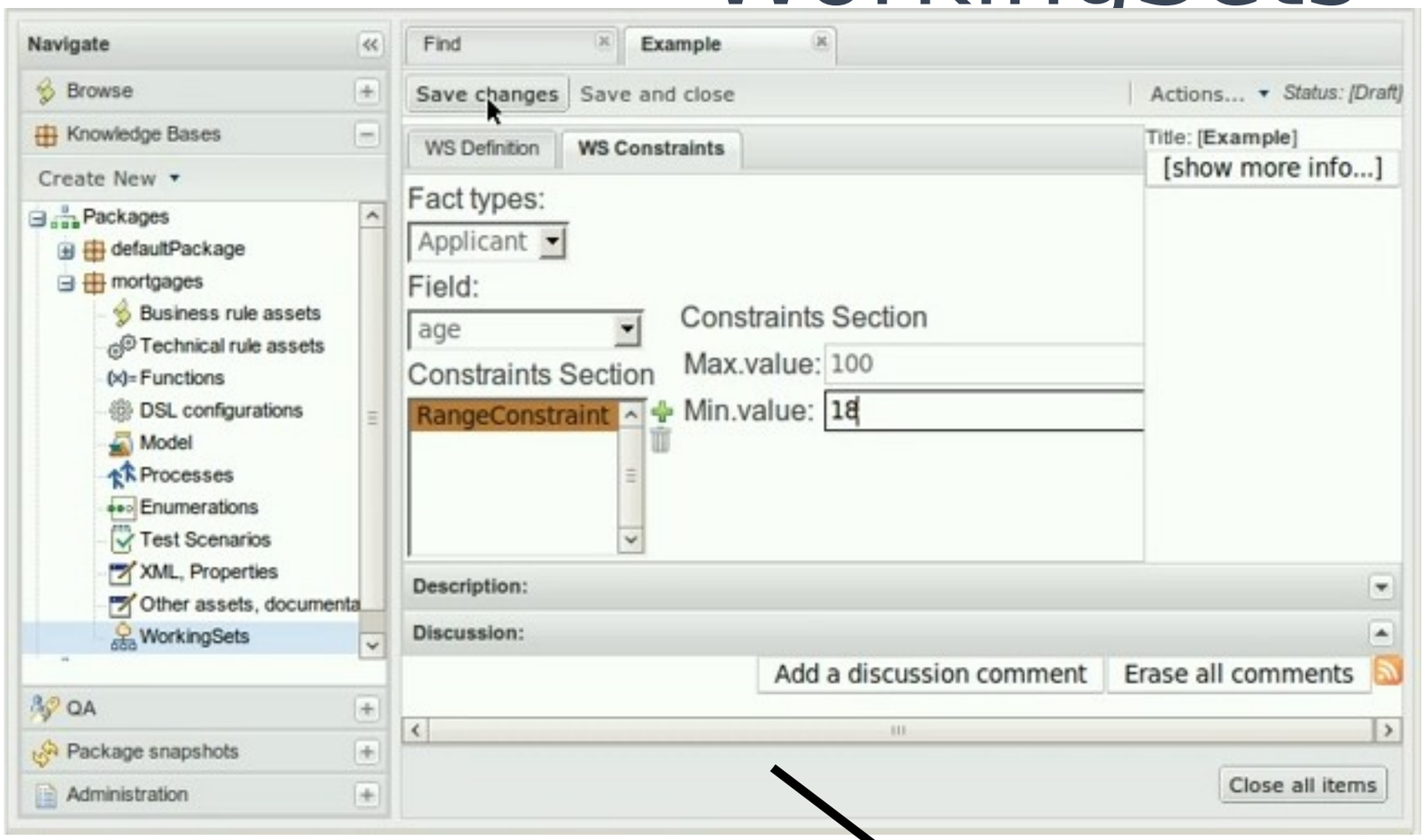


Working Sets

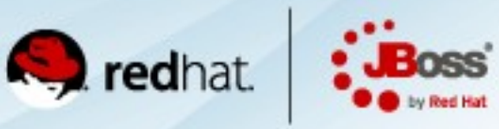




Fact Constraints within WorkingSets

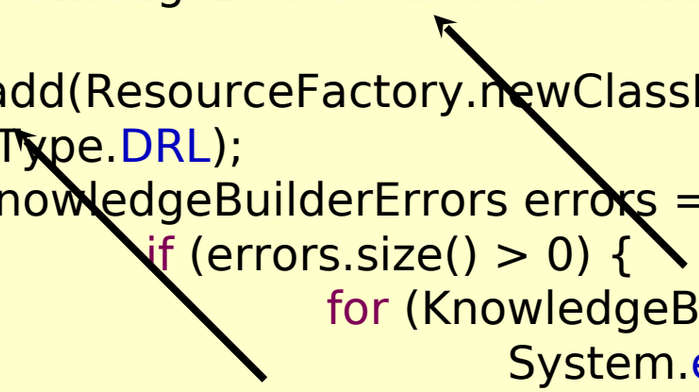


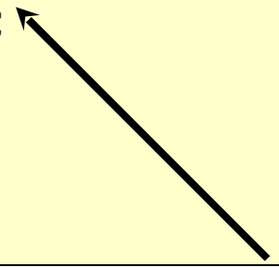
Runtime



Runtime: Building

```
private static KnowledgeBase readKnowledgeBase() throws Exception {
        KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();

kbuilder.add(ResourceFactory.newClassPathResource("fire.drl"),
ResourceType.DRL);
        KnowledgeBuilderErrors errors = kbuilder.getErrors();
                if (errors.size() > 0) {
                        for (KnowledgeBuilderError error: errors) {
                                System.err.println(error);
                        }
                        throw new IllegalArgumentException("Could not parse knowledge.");
                }

    KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
            kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
            return kbase;
    }
```





Camel & Spring Configuration

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:drools="http://drools.org/schema/drools-spring"
      http://drools.org/schema/drools-spring http://verylongurl/drools-spring-1.0.0.xsd>

  <drools:resource id="resource1" type="DRL" source="classpath:orgth:org/drools/container/spring/testSpring.drl"/>

  <drools:kbase id="kbase1">
    <drools:resources>
      <drools:resource type="DRL" source="classpath:org/drools/container/spring/testSpring.drl"/>
      <drools:resource ref="resource1"/>
      <drools:resource source="classpath:org/drools/container/spring/IntegrationExampleTest.xls" type="DTABLE">
        <drools:decisiontable-conf input-type="XLS" worksheet-name="Tables_2" />
      </drools:resource>
    </drools:resources>

    <drools:configuration>
      <drools:mbeans enabled="true" />
      <drools:event-processing-mode mode="STREAM" />
    </drool
  </drools:
</beans>
```

THE BRE now has extensive Spring support, the XSD can be found in the the drools-spring jar. The namespace is "http://drools.org/schema/drools-spring"

Runtime - Execution

```
                try {
                        // load up the knowledge base
                        KnowledgeBase kbase = readKnowledgeBase();
                        StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
                        KnowledgeRuntimeLogger logger = KnowledgeRuntimeLoggerFactory
.newFileLogger(ksession, "test");


                        for( String name: ROOM_NAMES ){
                            Room room = new Room( name );
                            name2room.put( name, room );
                            ksession.insert( room );
                            Sprinkler sprinkler = new Sprinkler( room );
                            ksession.insert( sprinkler );
                        }

                        // go !
                        ksession.fireAllRules();
```

> Everything is OK

Runtime - Execution

```
Fire kitchenFire = new Fire( name2room.get( "kitchen" ) );
Fire officeFire = new Fire( name2room.get( "office" ) );

FactHandle kitchenFireHandle = ksession.insert( kitchenFire );
FactHandle officeFireHandle = ksession.insert( officeFire );

ksession.fireAllRules();
```

```
> Raise the alarm
> Turn on the sprinkler for room kitchen
> Turn on the sprinkler for room office
```

Runtime - Execution

```
ksession.retract( kitchenFireHandle );
ksession.retract( officeFireHandle );
ksession.fireAllRules();
```

```
> Turn on the sprinkler for room office
> Turn on the sprinkler for room kitchen
> Cancel the alarm
> Everything is ok
```

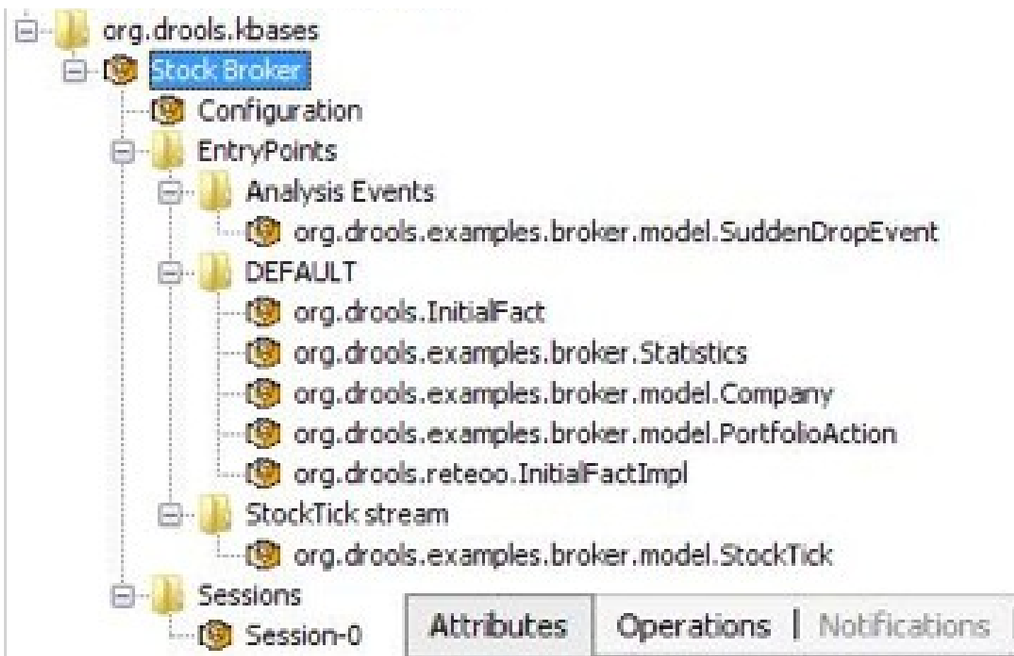
Use FROM for external data

```
rule “Find Vehicles for a given zip code"
when
  $zipCode : ZipCode() Vehicle() from   $hibernate.getNamedQuery( “FindVehicles” )
.setParameters( [ “zipCode” : $zipCode ]) .list()
then
...
end
```

Can be a Web Service, Hibernate or any external system

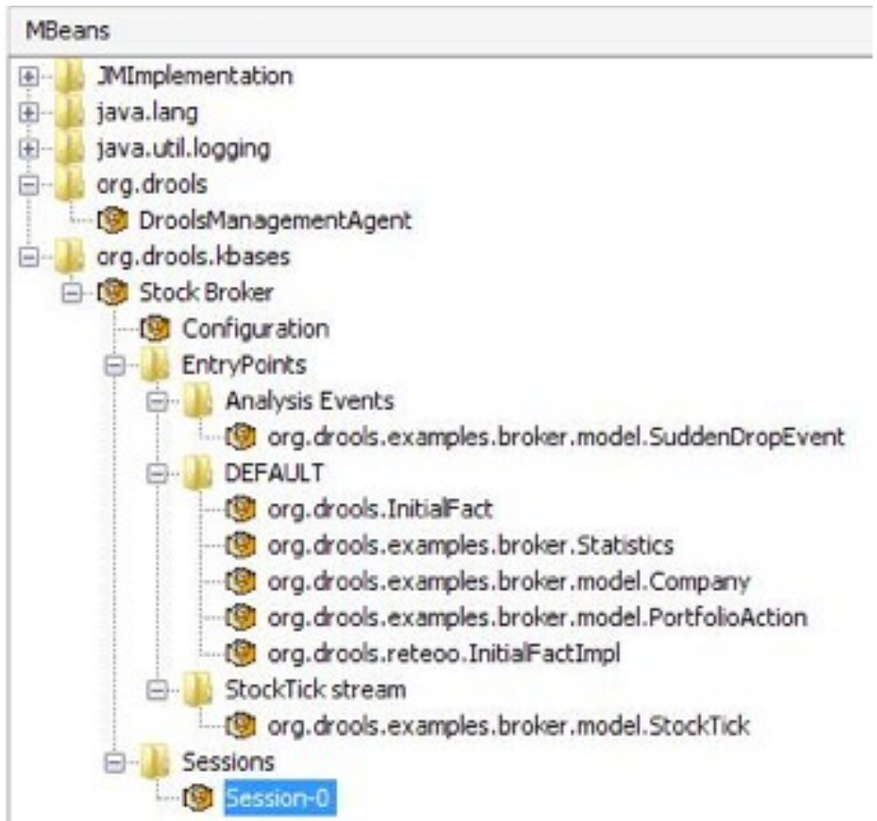
API Enhancements - JMX

- Drools 5.1 brings support to the JMX standard and enables knowledge base and knowledge session monitoring and inspection using any JMX console.

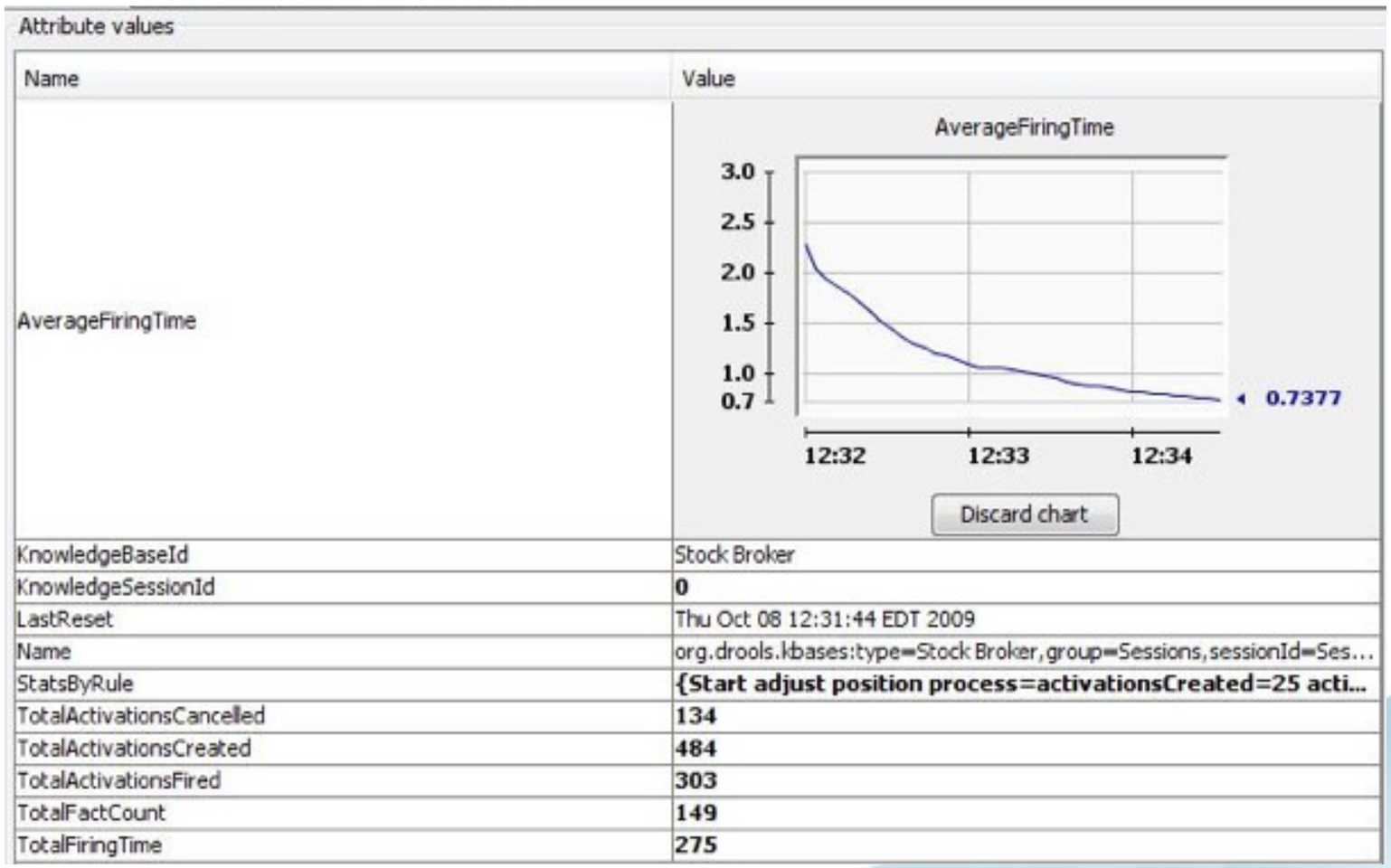


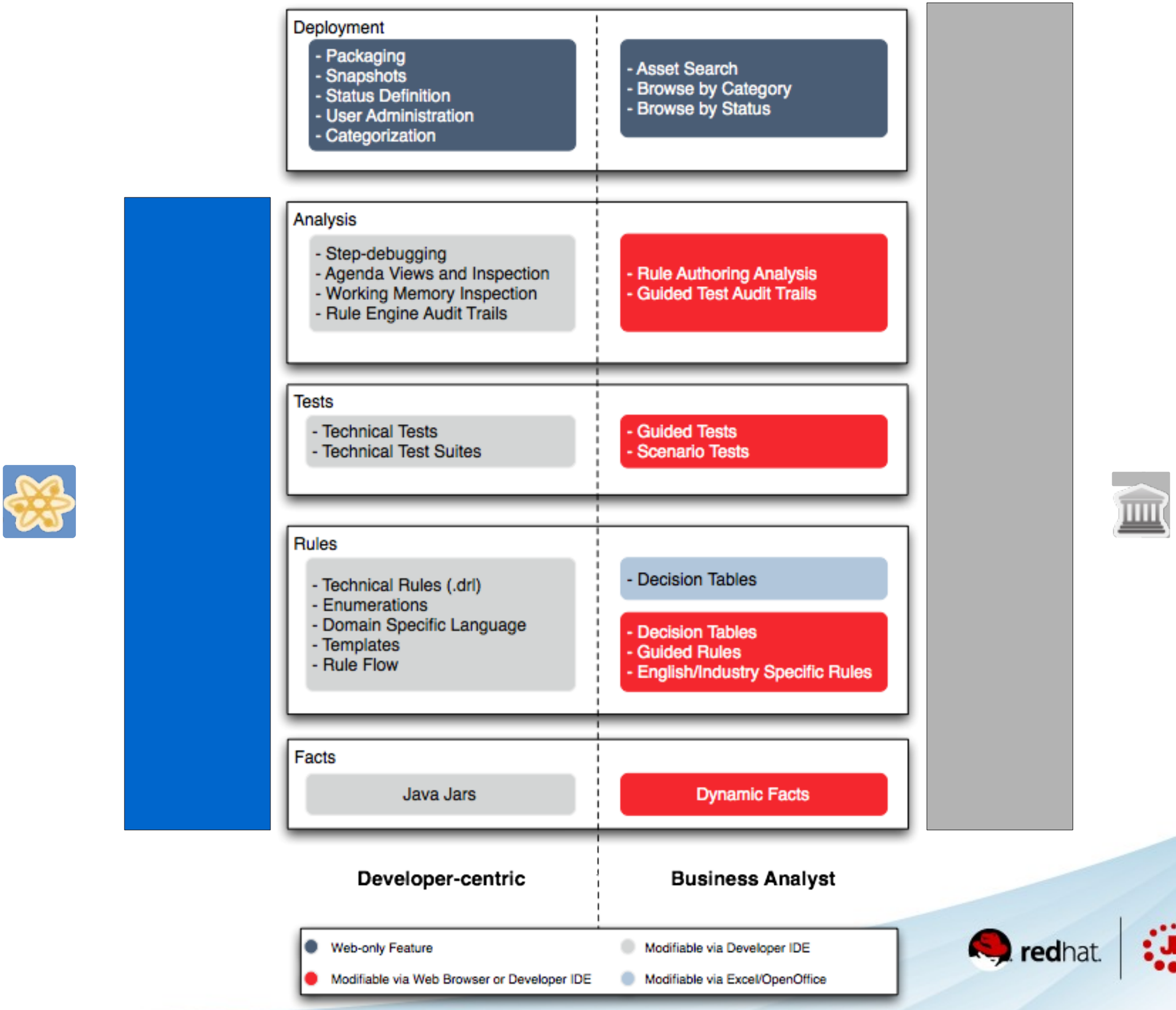
KnowledgeBase Stats

API Enhancements - JMX

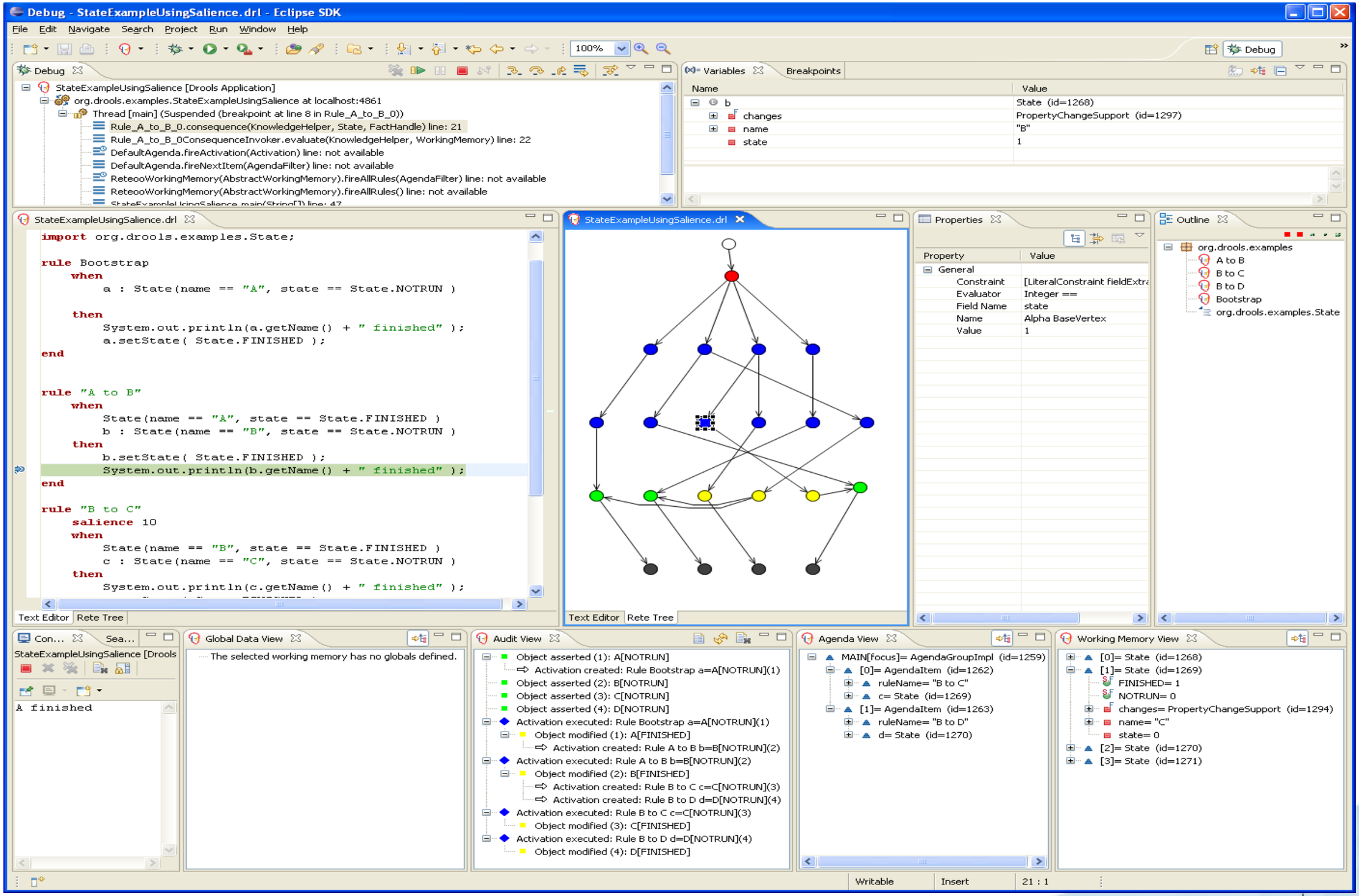


KnowledgeSession Stats



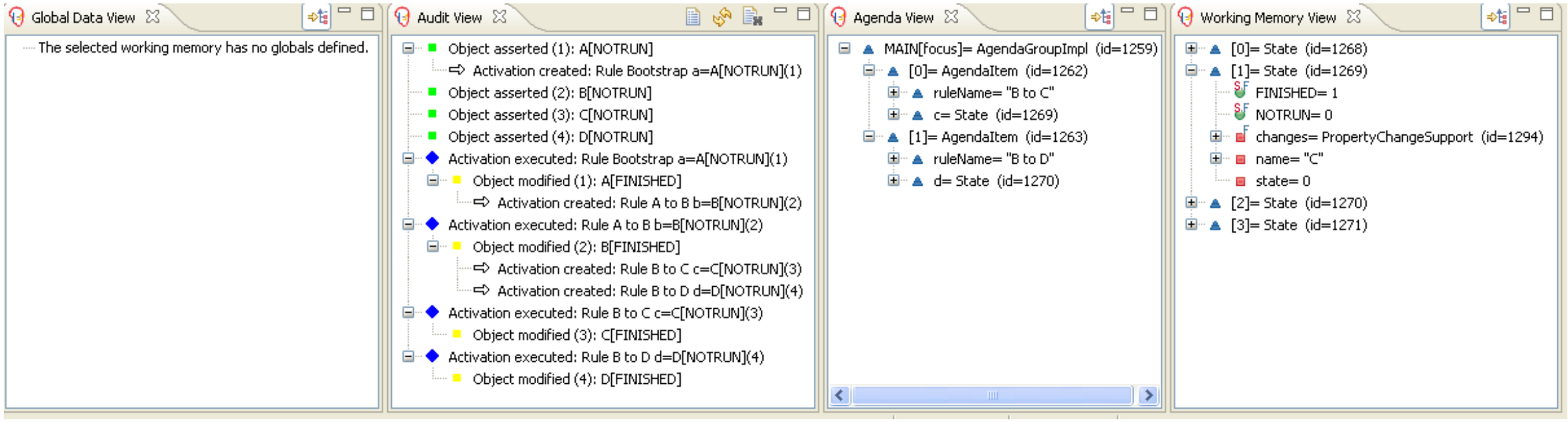

Deployment
- Packaging
- Snapshots
- Status Definition
- User Administration
- Categorization
- Asset Search
- Browse by Category
- Browse by Status
Analysis
- Step-debugging
- Agenda Views and Inspection
- Working Memory Inspection
- Rule Engine Audit Trails
- Rule Authoring Analysis
- Guided Test Audit Trails
Tests
- Technical Tests
- Technical Test Suites
- Guided Tests
- Scenario Tests
Rules
- Technical Rules (.drl)
- Enumerations
- Domain Specific Language
- Templates
- Rule Flow
- Decision Tables
- Decision Tables
- Guided Rules
- English/Industry Specific Rules
Facts
Java Jars
Dynamic Facts
Developer-centric
Business Analyst
Web-only Feature
Modifiable via Web Browser or Developer IDE
Modifiable via Developer IDE
Modifiable via Excel/OpenOffice

Eclipse – Developer Perspective



Debugging

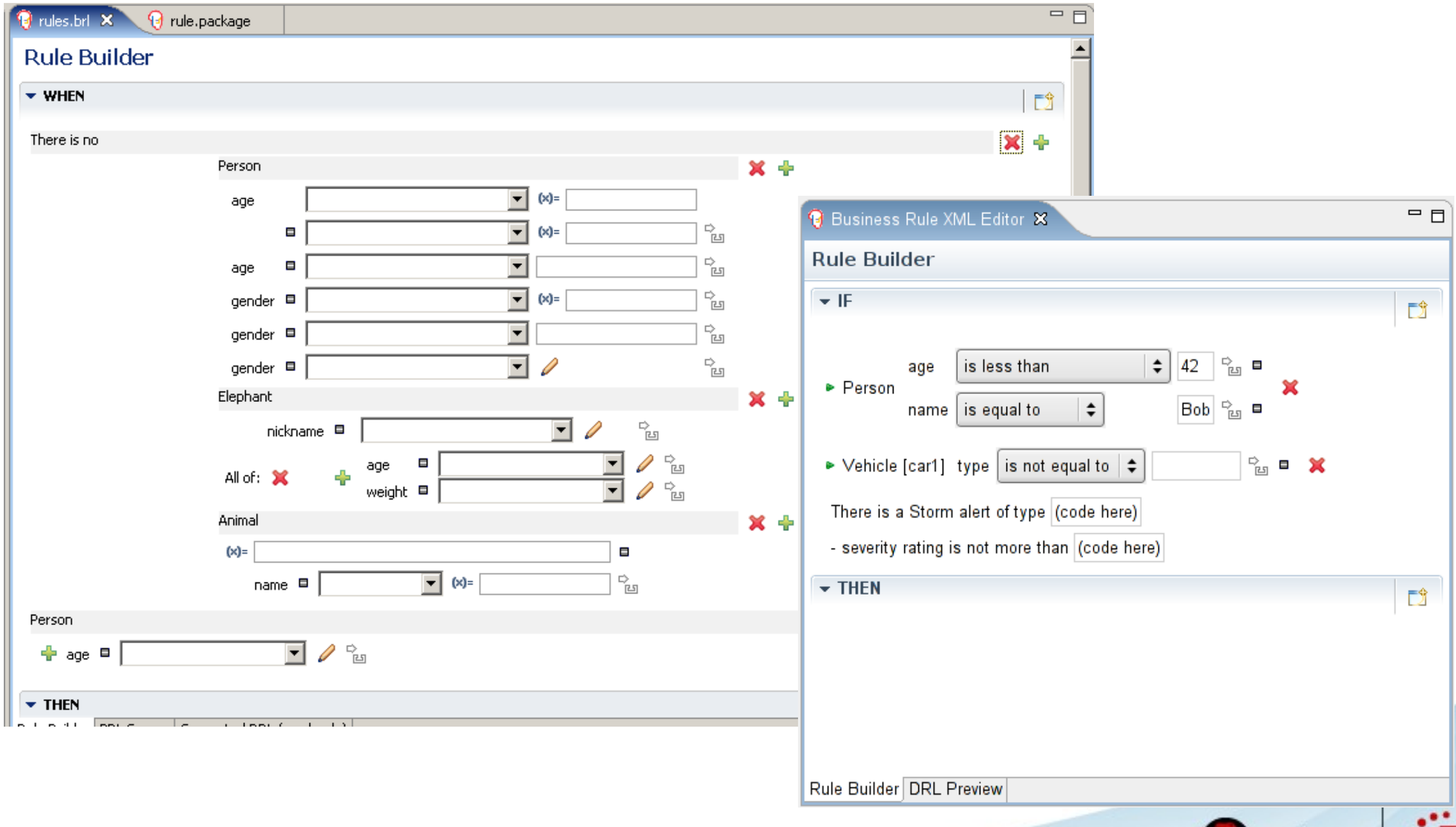
Debug Views



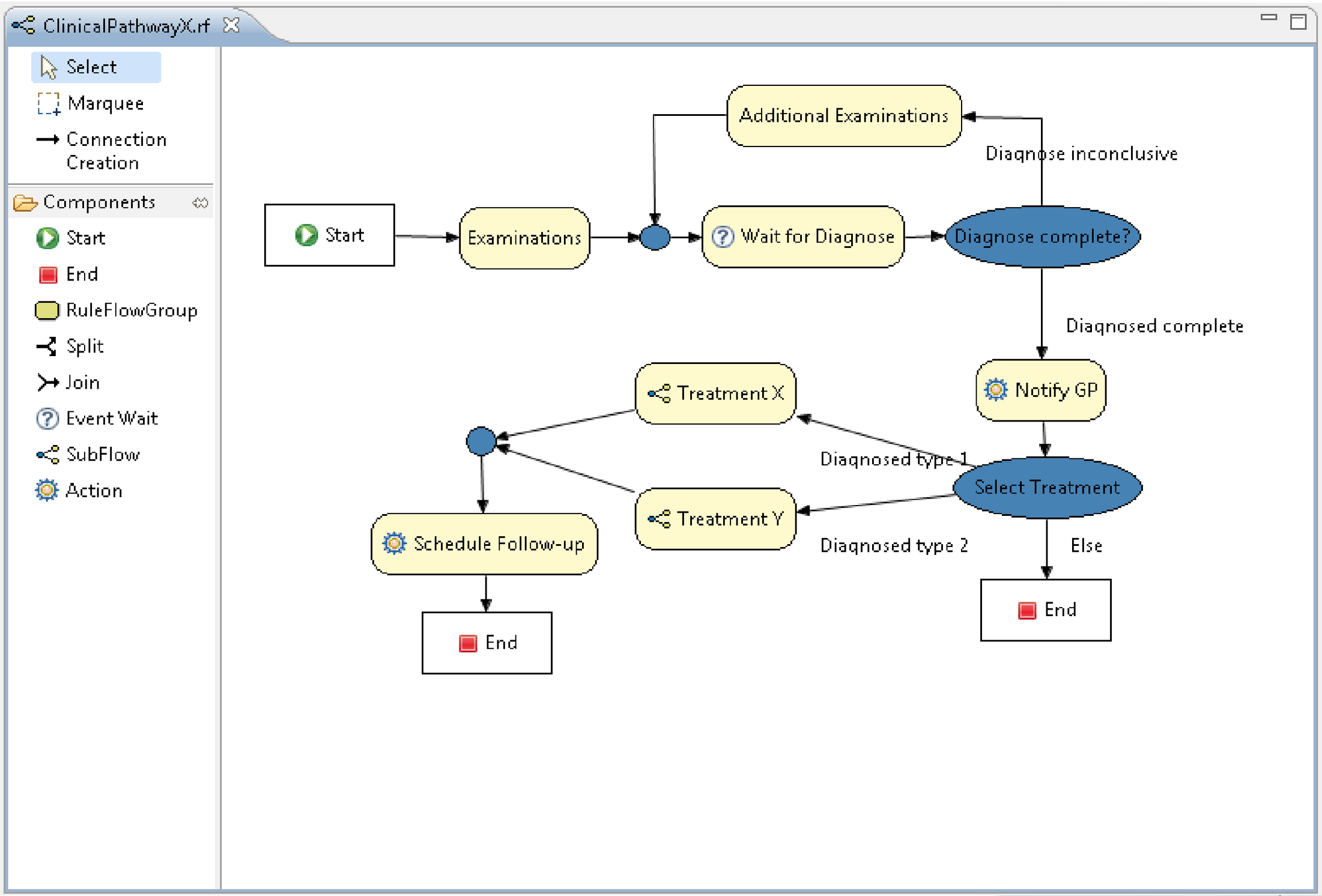
- New rule “perspective” to configure IDE for rules as needed

Guided Rule Editor (Eclipse)

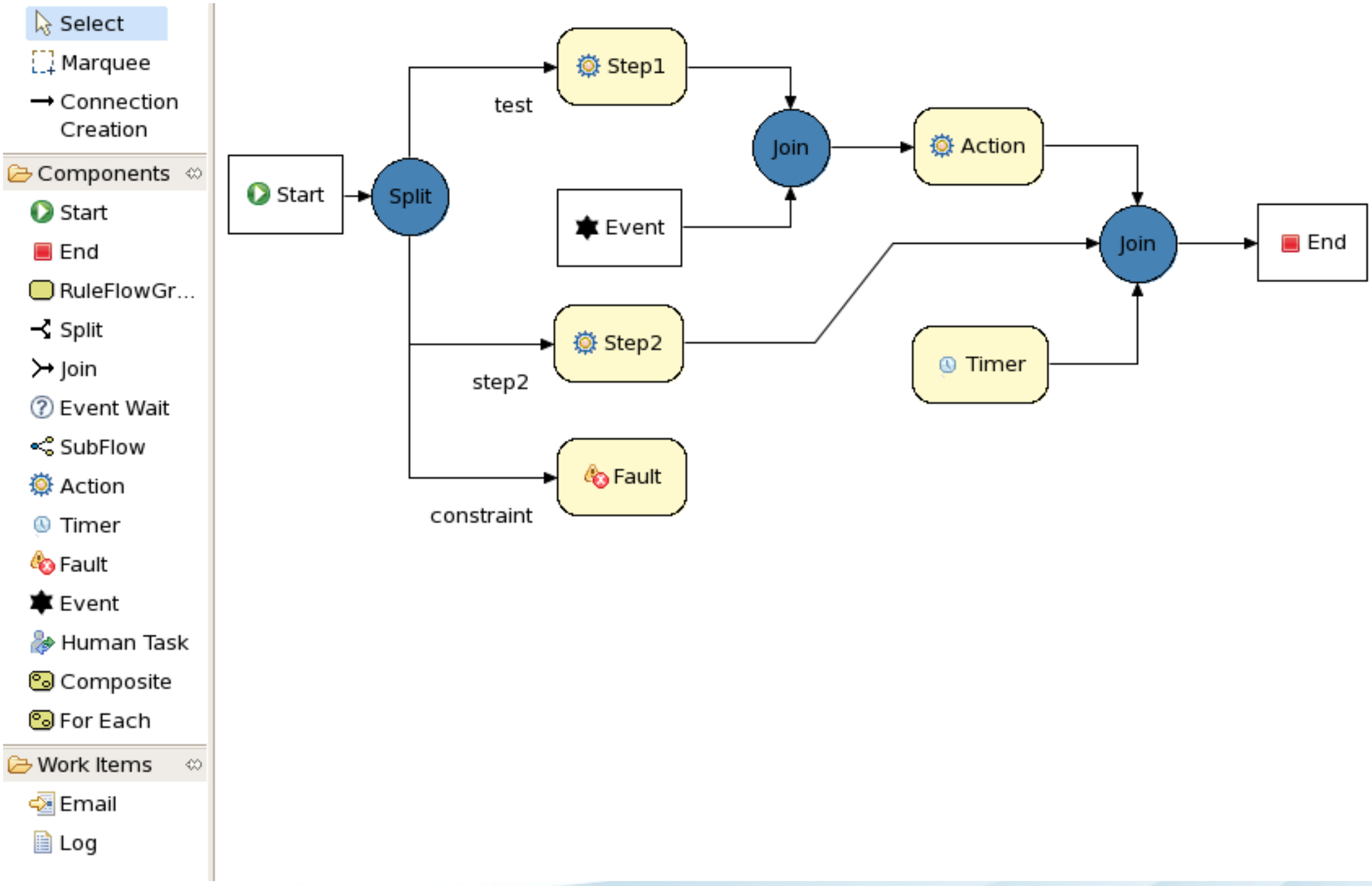
Eclipse Guided Editor



Rule Flow

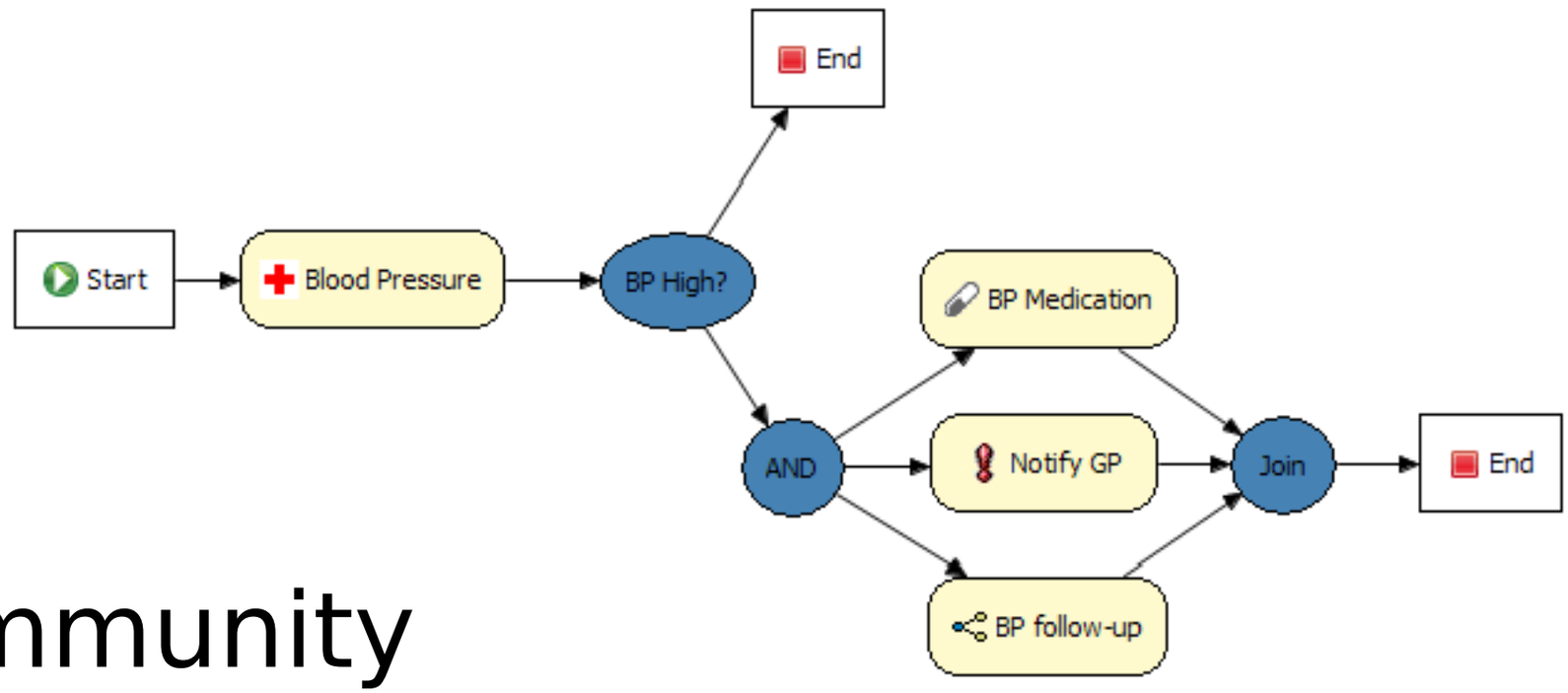

ClinicalPathwayX.rf
Select
Marquee
Connection Creation
Components
Start
End
RuleFlowGroup
Split
Join
Event Wait
SubFlow
Action
Start
Examinations
Additional Examinations
Wait for Diagnose
Diagnose complete?
Diagnose inconclusive
Diagnosed complete
Notify GP
Select Treatment
Diagnosed type 1
Diagnosed type 2
Else
Treatment X
Treatment Y
Schedule Follow-up
End
End

Node Types


Select
Marquee
Connection Creation
Components
Start
End
RuleFlowGr...
Split
Join
Event Wait
SubFlow
Action
Timer
Fault
Event
Human Task
Composite
For Each
Work Items
Email
Log
Start
Split
test
Step1
step2
Step2
constraint
Fault
Event
Join
Action
Timer
Join
End

Domain Specific Process Steps

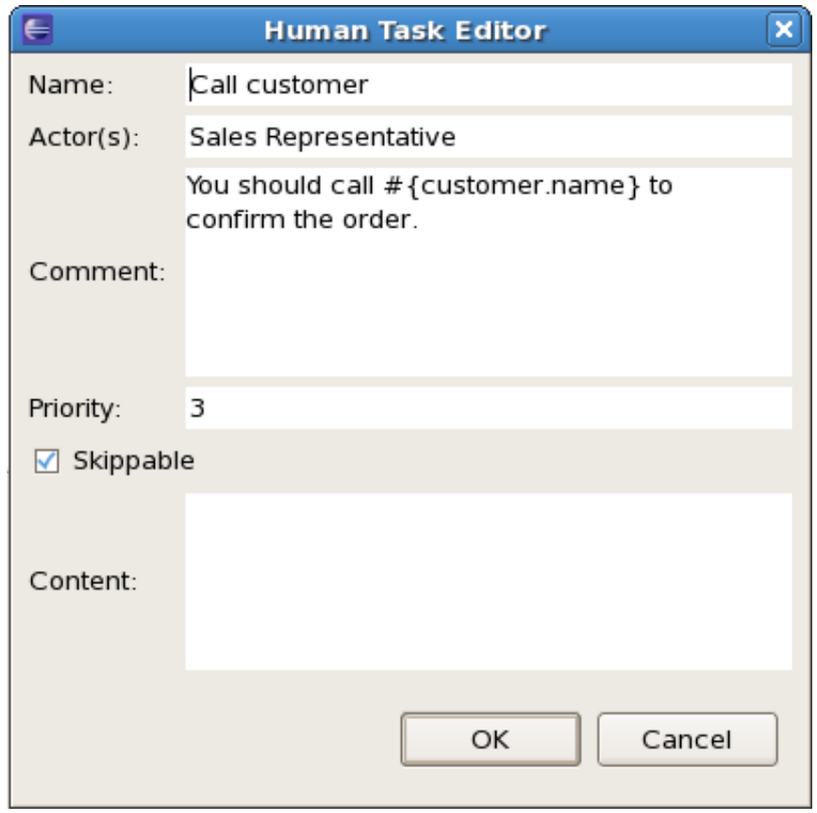
- Extend Drools Flow to incorporate your application's needs

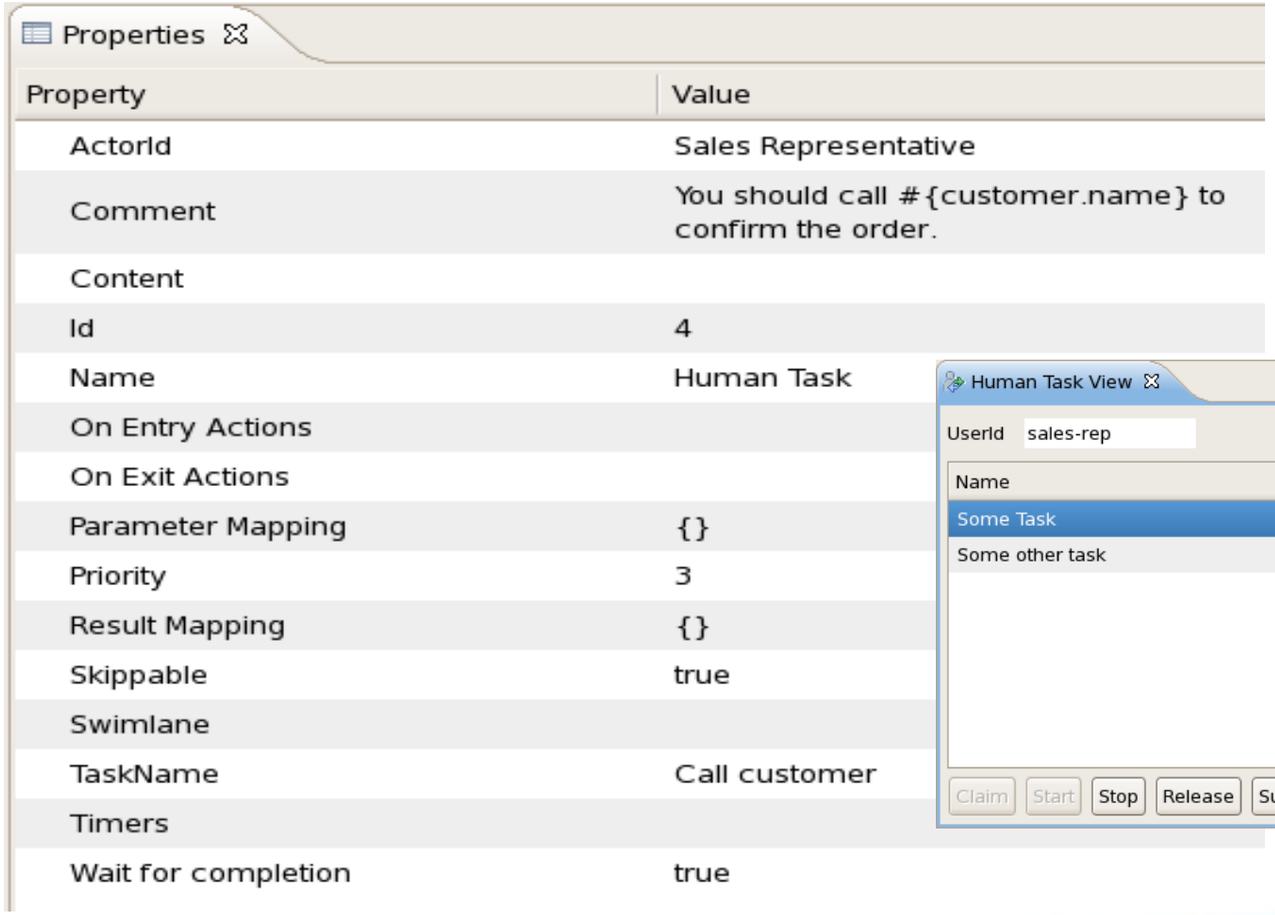


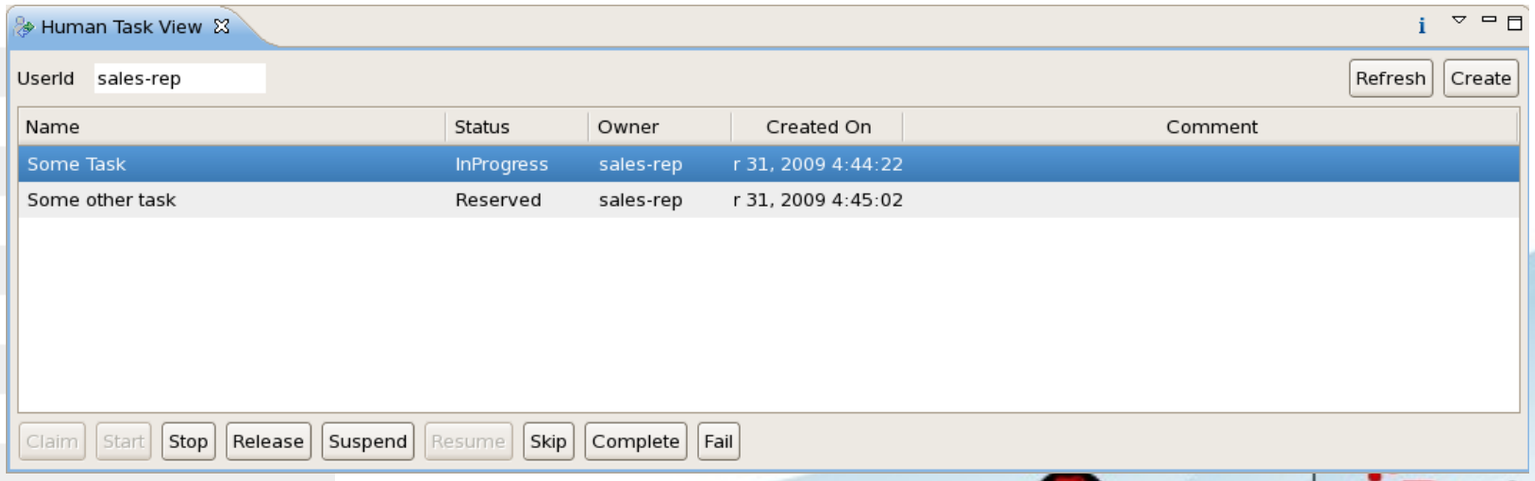
Note: Community Only

Human Based Tasks

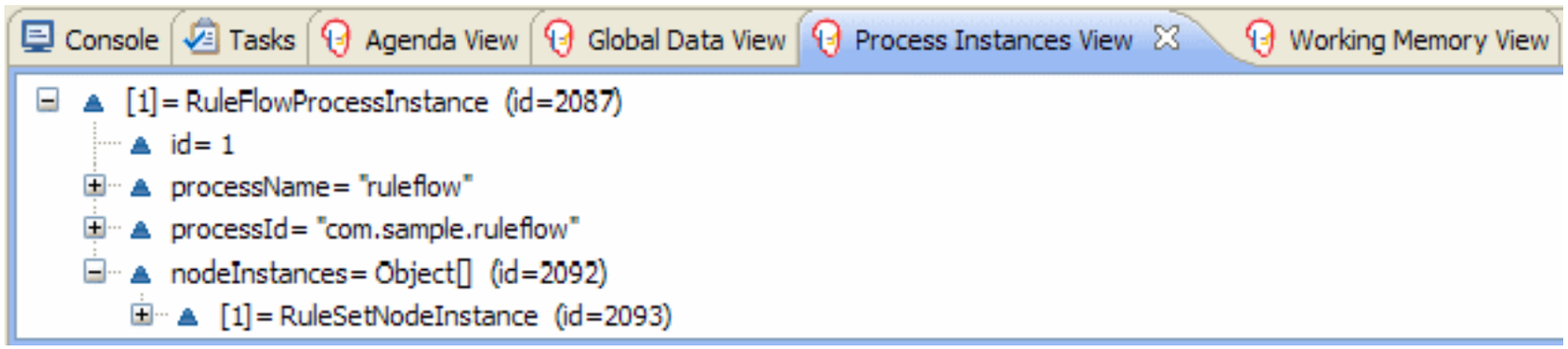
- Swim-lanes
- Modular Implementation
- Supports WS-HumanTask

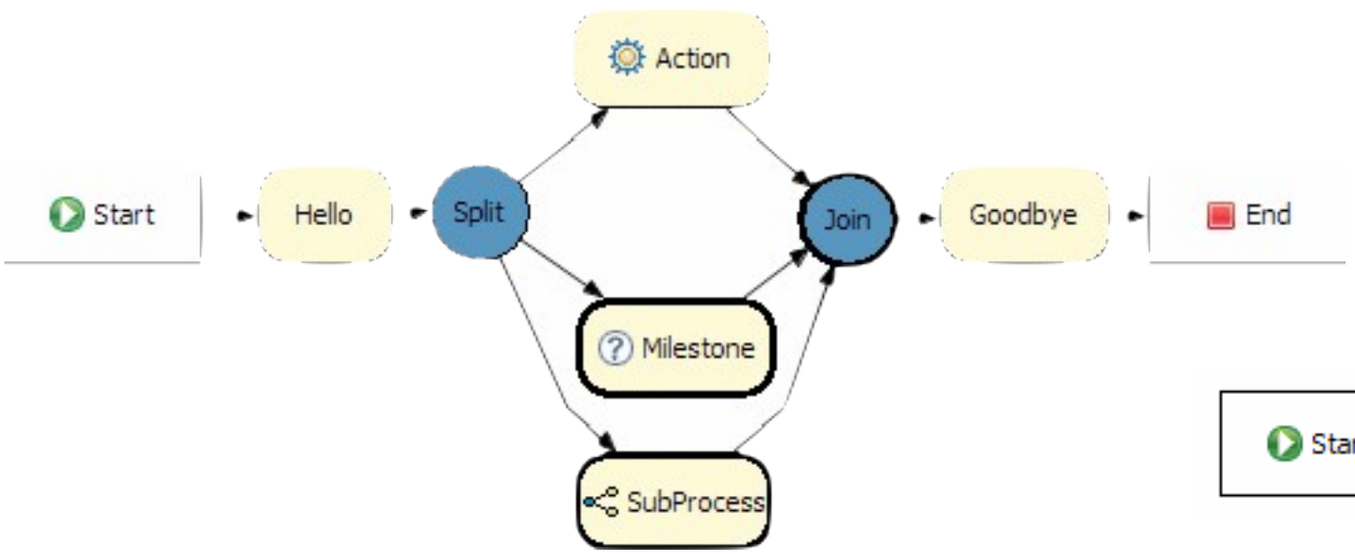


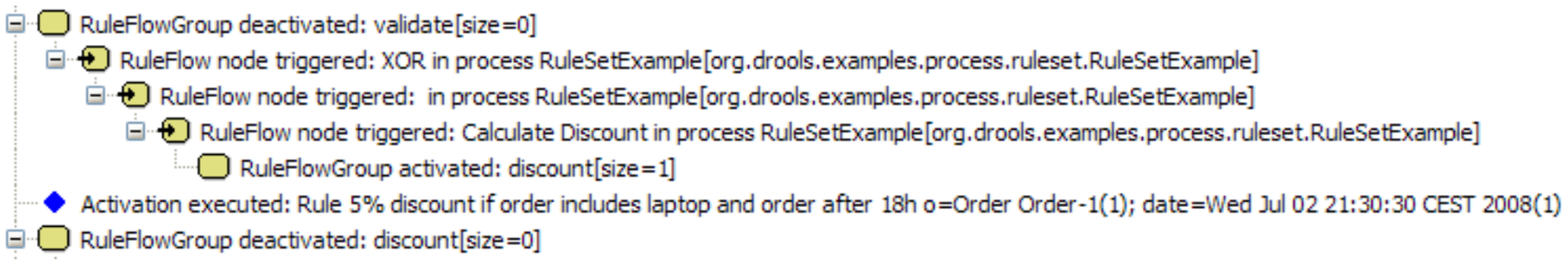




Integrated Tooling







CEP

Complex event processing

Real time events (concurrent events – thread safe):

sliding windows:

```
Pattern(...) over window:time(3000)
```

Drools Fusion

redhat. | JBoss by Red Hat

Fusion Enables:

- Event Detection:
  - From an event cloud or set of streams, select all the meaningful events, and only them.
- [Temporal] Event Correlation:
  - Ability to correlate events and facts declaring both temporal and non-temporal constraints between them.
  - Ability to reason over event aggregation
- Event Abstraction:
  - Ability to compose complex events from atomic events AND reason over them

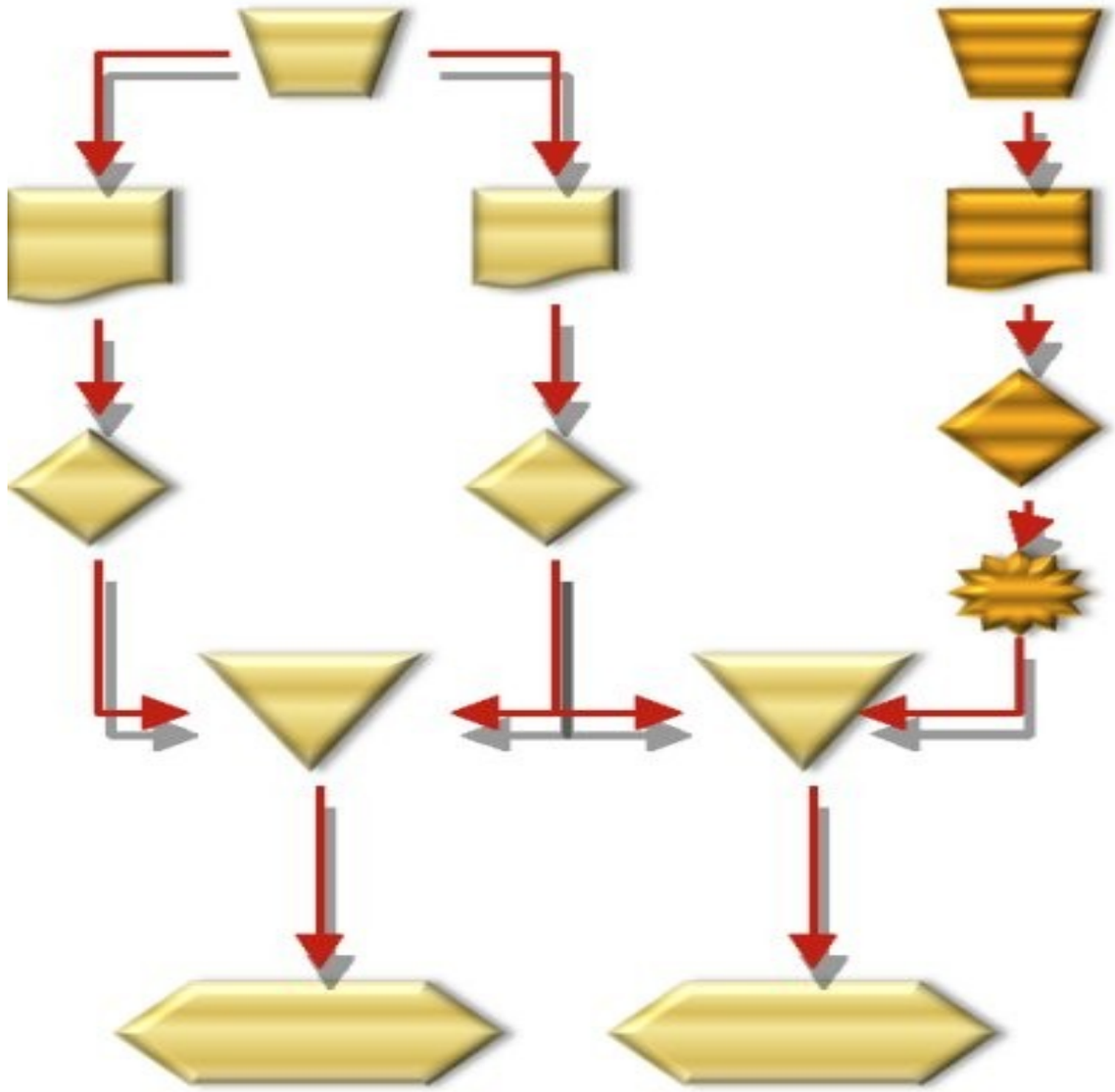
Events

```
// declaring existing class
import some.package.VoiceCall
declare VoiceCall
  @role( event )
  @timestamp( calltime )
  @duration( duration )
end

// generating an event class
declare StockTick
  @role( event )

  symbol : String
  price : double
end
```

- (Usually) Immutable Facts
- Strong Temporal Relationship
- Managed Lifecylce
- Use of sliding windows
- Can provide metadata on:
  - @timestamp, @duration or @expires

Streams

```
rule "Buying Cheese"
when
    Customer( name == "Bob",
              $likes : likes )
    OrderEvent( product == "Cheese",
                type == $likes ) from
                entry-point "online stream"
then
    // do something
end
```



Temporal Reasoning

```
rule "Cancel Transaction"
when
    Customer( $cid : id)
    $o : OrderEvent( customer == $cid,
                     $tid : transactionId )
    not OrderAck( transactionId == $tid,
                  this after[0,10] $o )
then
    // do something
end
```

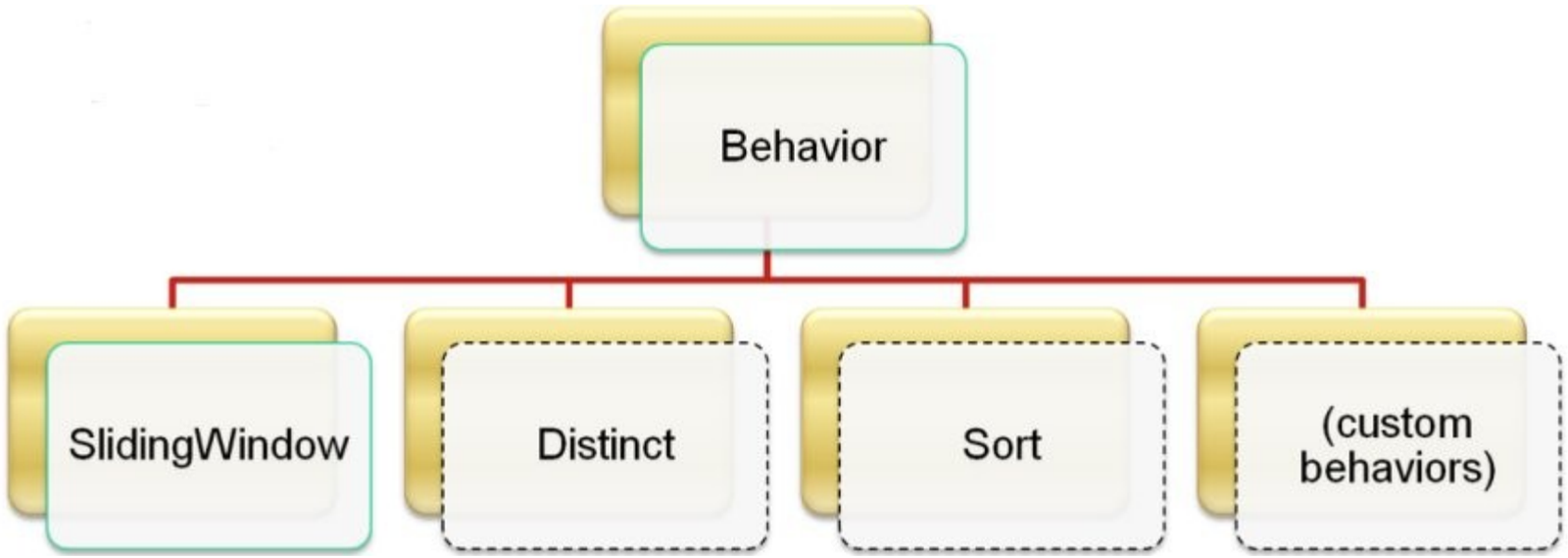
- Event and Time semantics:
  - Point in Time
  - Over an Interval
- Unified semantics for event correlation over time
- Temporal Constraints:
  - Set of 13 operators to express temporal relationship between events

Expressive Event Constraints

	Point-Point	Point-Interval	Interval-Interval
A ***before*** B			
A ***meets*** B			
A ***overlaps*** B			
A ***includes*** B			
A ***finishes*** B			
A ***starts*** B			
A ***coincides*** B			

Behaviors & Sliding Windows

- **Behaviors**: special semantics to certain patterns
  - sliding windows, distinct, sort, etc
- **SlidingWindow**: Allows reasoning over a moving window of “interest”
  - Time
  - Length

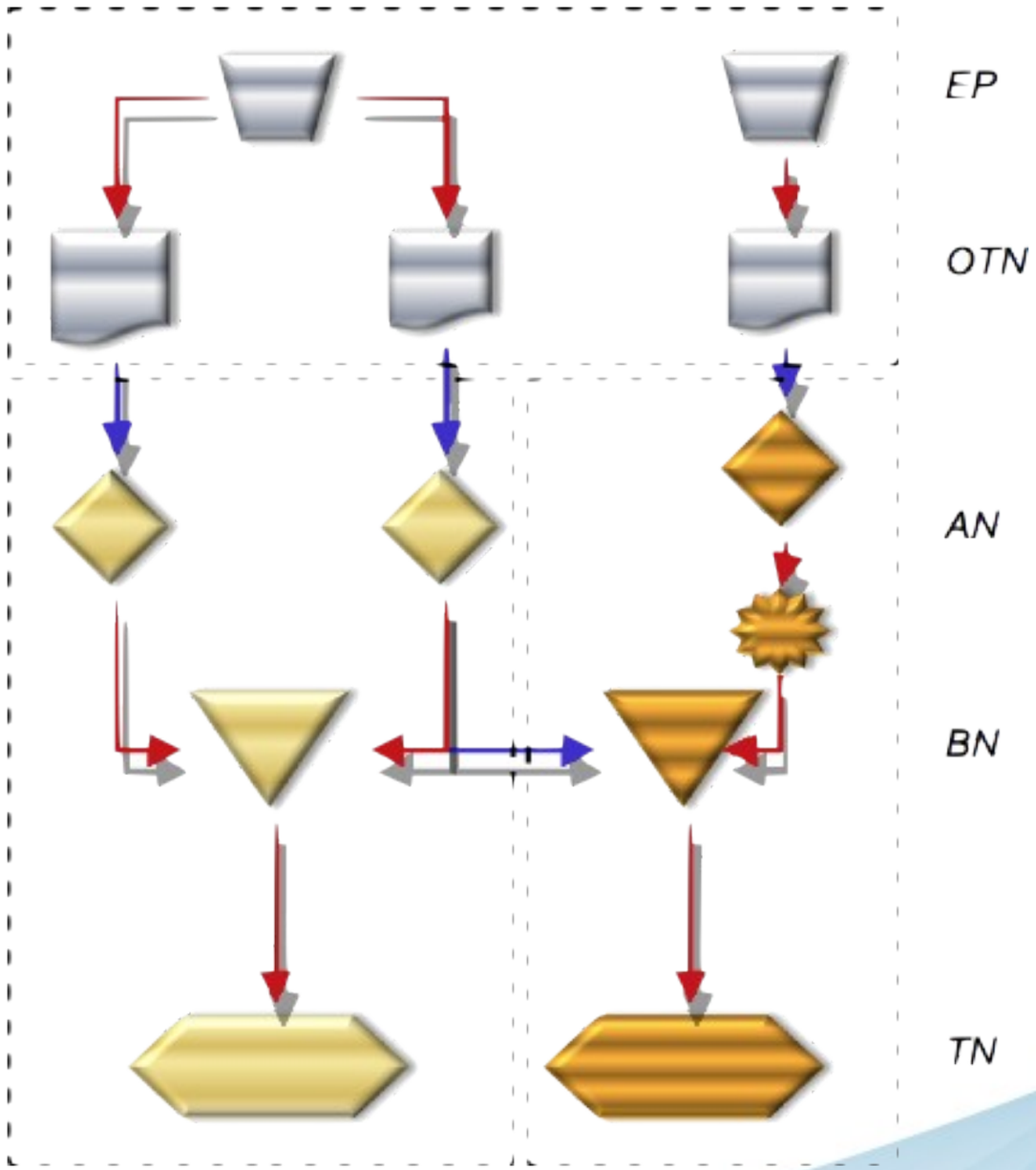


Memory Management

- CEP scenarios are stateful by nature.
- Events usually are only interesting during a short period of time.
- Fusion manages the benefits memory management while events still in window of consideration

Rulebase Partitioning

- Achieves coarse grained parallelization
- No fundamental changes in the matching algorithm (ReteOO)
- Preserves optimizations, especially node sharing



Drools Planner

Planner

Note:
Not Yet Supported!

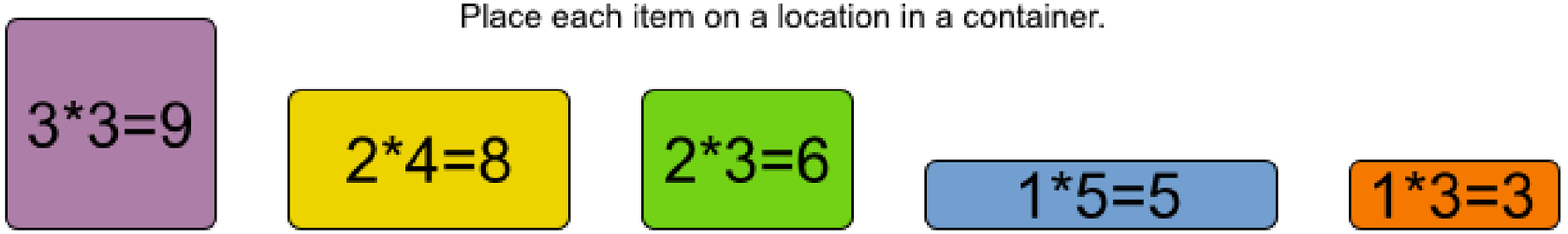
redhat. | JBoss by Red Hat

What is Planner?

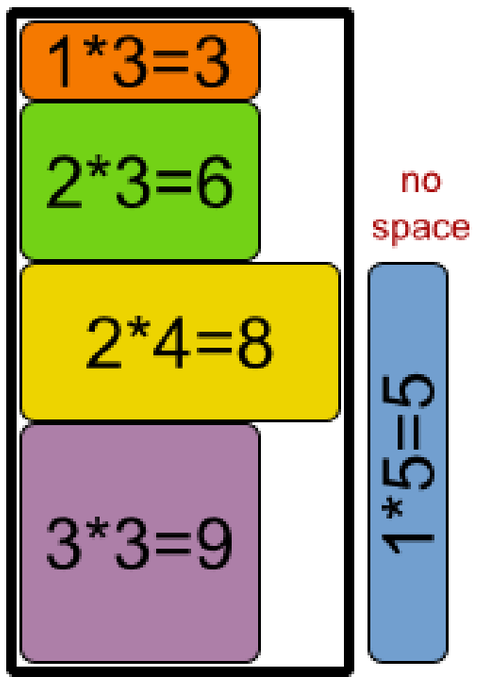
- Optimizes Automated Planning
- Use Cases:
  - Space Planning
  - Employee Shift Rostering
  - Team Scheduling
- Still in development!
  - Not commercially supported.

Bin packing

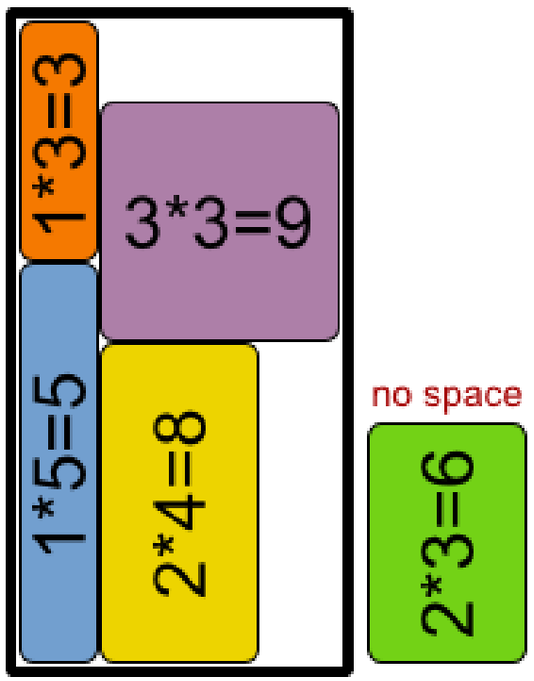
Place each item on a location in a container.



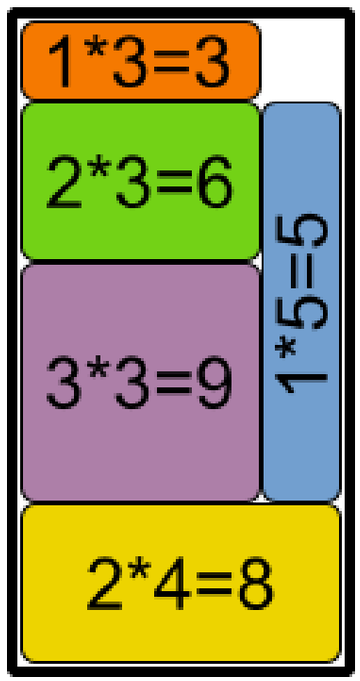
Largest size first



Largest side first



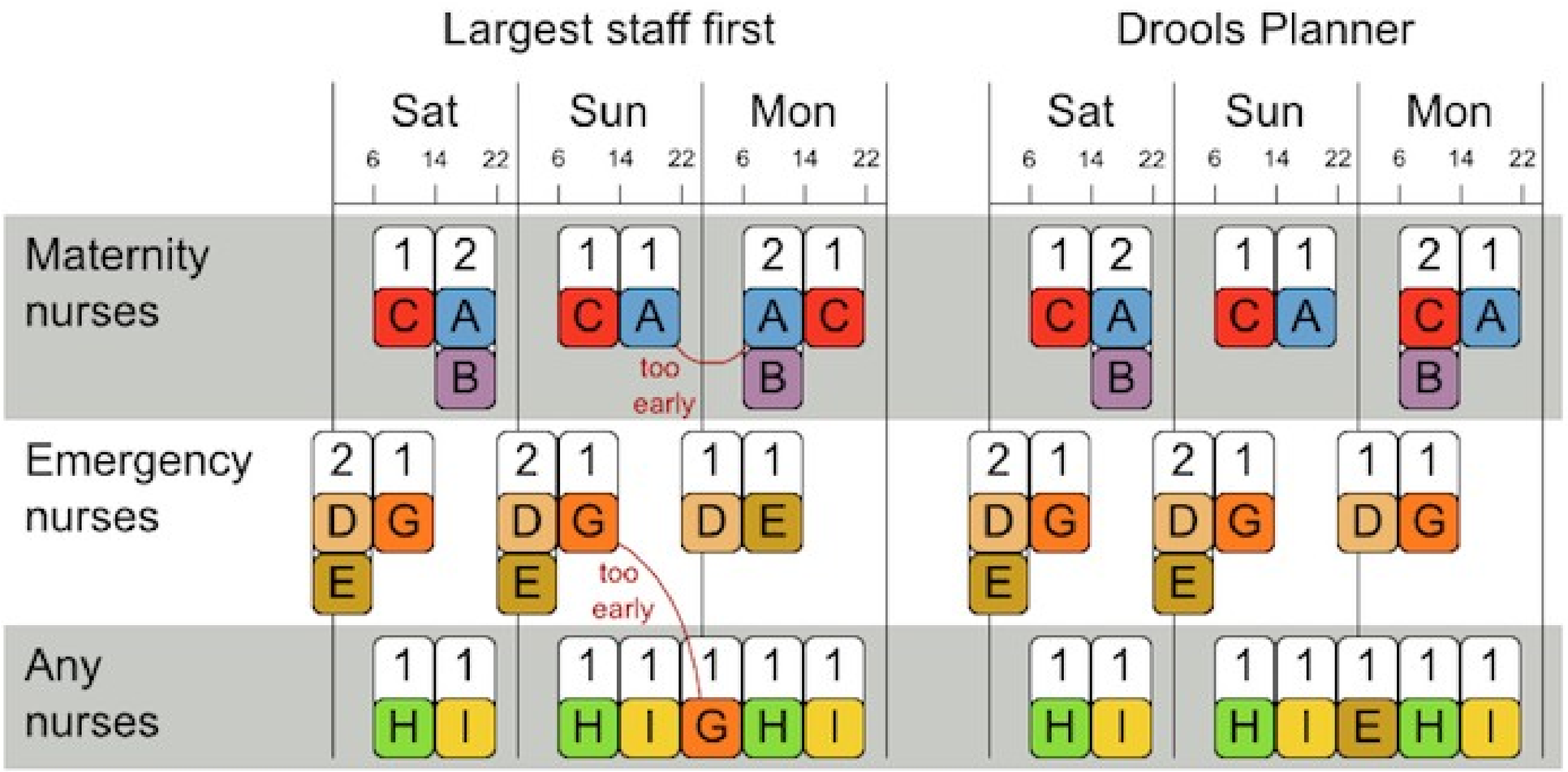
Drools Planner



Employee shift rostering

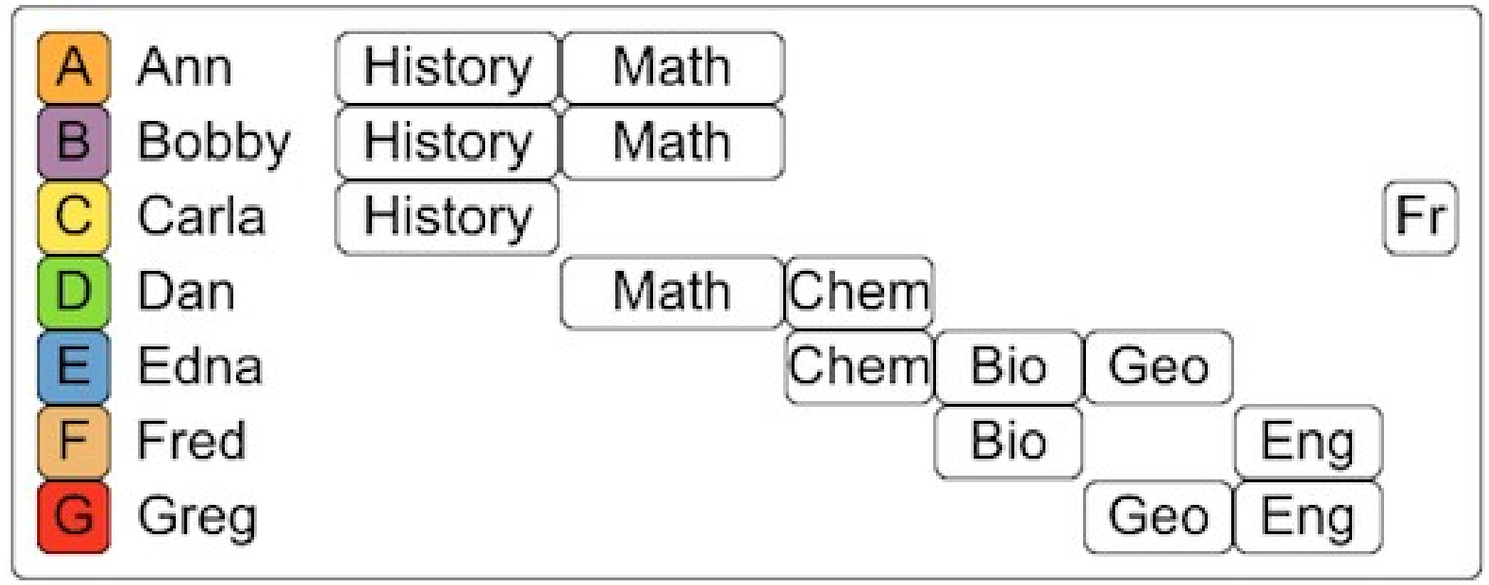
Populate each work shift with a nurse.

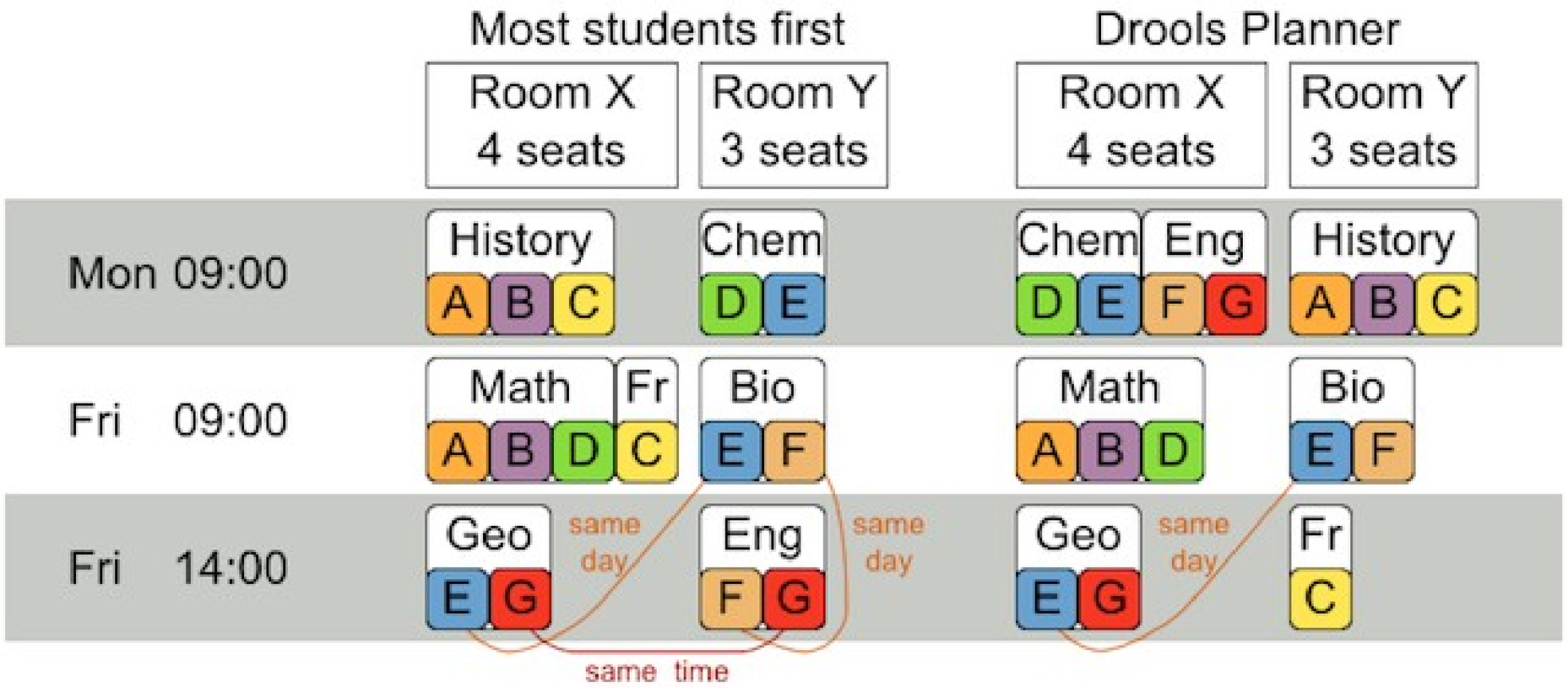
Maternity nurses	Emergency nurses	Basic nurses
A Ann, B Beth, C Cory	D Dan, E Elin, G Greg	H Hue, I Ilse



Examination timetabling

Assign each exam a period and a room.



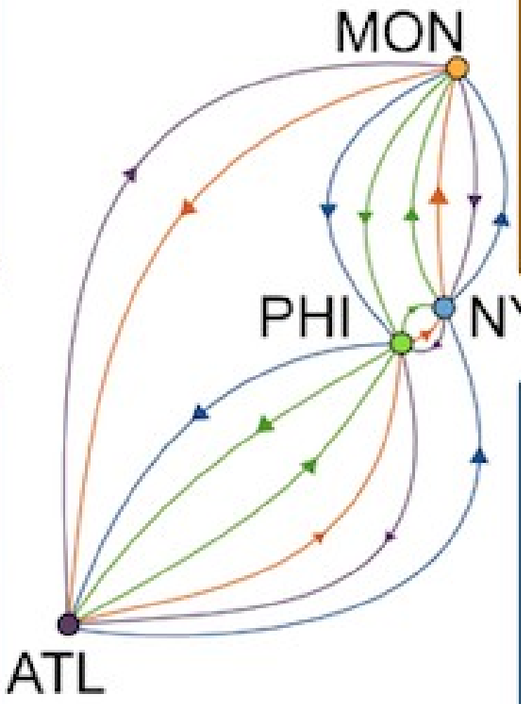


Traveling tournament

Schedule each match in a timeslot.

Philadephia Phillies					
1	away to	NYM			80
2	away to	MON			337
3	PHI	VS	ATL		380
4	PHI	VS	NYM		0
5	PHI	VS	MON		0
6	away to	ATL			665
Team distance: 2.127					665

Montréal Expos				
1	MON	VS	ATL	0
2	MON	VS	PHI	0
3	MON	VS	NYM	0
4	away to	ATL		929
5	away to	PHI		665
6	away to	NYM		80
Team distance: 2.011				337



Atlanta Braves				
1	away to	MON		929
2	away to	NYM		337
3	away to	PHI		80
4	ATL	VS	MON	665
5	ATL	VS	NYM	0
6	ATL	VS	PHI	0
Team distance: 2.011				0

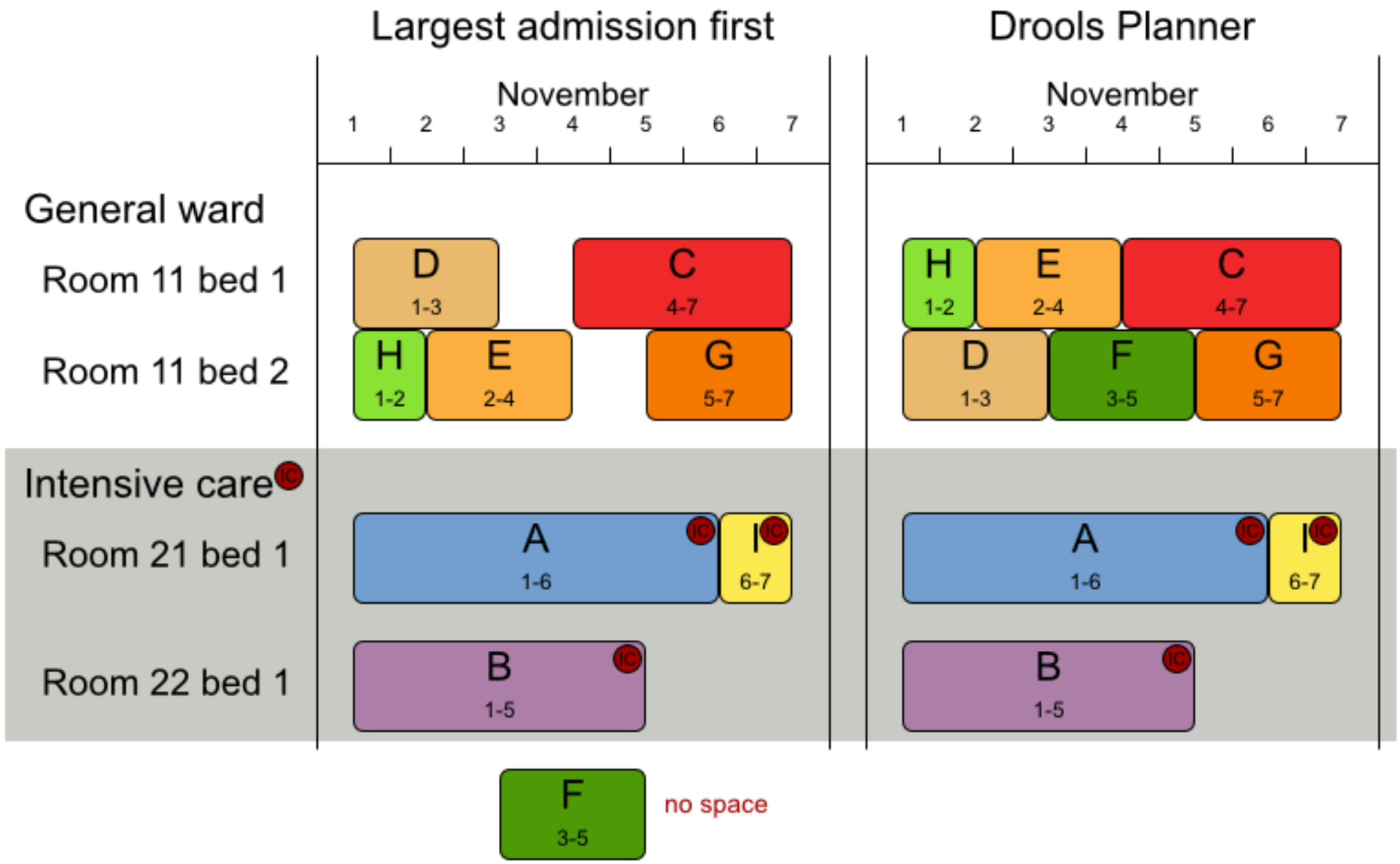
New York Mets				
1	NYM	VS	PHI	0
2	NYM	VS	ATL	0
3	away to	MON		337
4	away to	PHI		380
5	away to	ATL		665
6	NYM	VS	MON	745
Team distance: 2.127				0

Drools Planner

Total distance: 8.276

Patient admission schedule

Assign each patient a hospital bed.



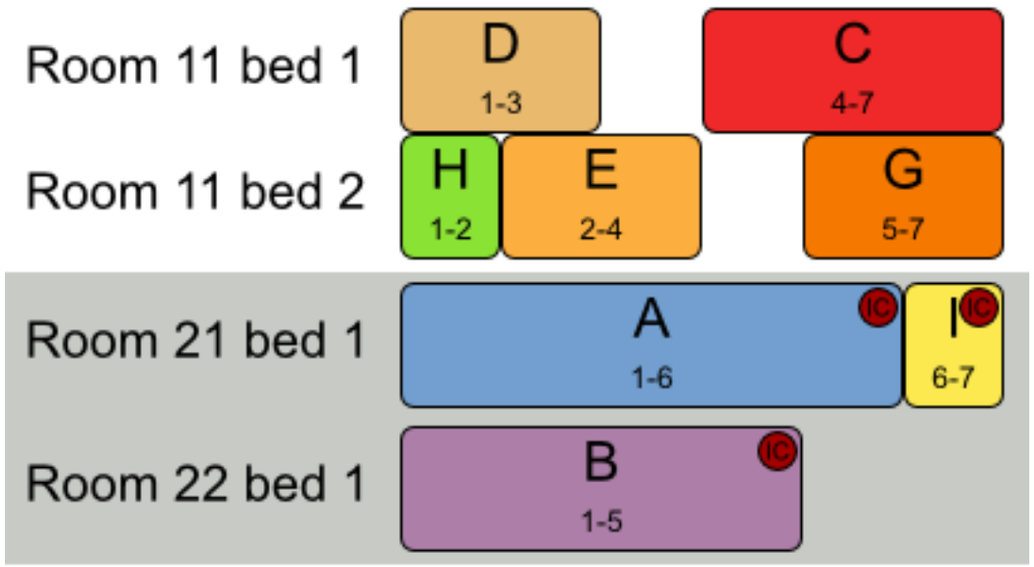
Patient Admission Schedule

- Hard constraints
  - No 2 patients in same bed in same night
  - Room gender limitation
  - Department minimum or maximum age
  - Patient requires specific room equipment(s)
- Soft constraints
  - Patient prefers maximum room size
  - Department specialization
  - Room specialization
  - Patient prefers specific room equipment(s)

Planner Syntax

```
// If a hospital patient prefers specific equipment, try to assign him/her a hospital room with such
equipment.
 rule "preferredPatientEquipment"
     when
         // If a patient prefers specific equipment
         PreferredPatientEquipment($patient : patient, $equipment : equipment);

         // and that patient is assigned a room
         BedDesignation(patient == $patient, $room : room);

         // and that room doesn't have that equipment
         not RoomEquipment(room == $room, equipment == $equipment);
    then
         ... // lower the score with a certain weight
end
```

Needle in a Haystack



- How many possible solutions?
  - 310 beds
    - in 105 rooms
    - in 4 departments
      - 84 nights
- 2750 patients (admissions)
- Numbers from a real dataset

Commercial Support

- Drools ships within several supported platforms:
  - JBoss Enterprise BRMS
  - JBoss Enterprise SOA Platform *
  - JBoss Enterprise Application Platform **

Support Open Source!

- We need You!
  - **Subscribe**
  - Contribute
    - Code
    - Features
    - Documentation
    - Testing

- Let us continue our work that benefits so many.

Materials

- This presentations:
  - https://github.com/derrickhackman/Presentations
- JBoss Developer Studio Download:
  - http://devstudio.jboss.com/earlyaccess/index.html
- JBoss World Registration:
  - http://redhat.com/summit use registration code RHSRPLO
- BRMS Camel Integration
  - http://lucazamador.wordpress.com/2010/05/28/drools-apache-camel-integration/
- BRMS JMX
  - http://blog.athico.com/2009/10/drools-monitoring-with-jmx.html
- From/Collect/Accumulate Support
  - http://ilesteban.wordpress.com/2010/05/28/guvnor-guided-editor-suuport-for-fromcollectaccumulate-elements/
- Templates
  - http://locademiaz.wordpress.com/2010/05/28/new-guvnor-feature-rules-templates/