

Derrick Kittler

Solutions Architect



Drools and the BRMS

The Business Logic Integration Platform

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

What is JBoss BRMS?

JBoss Enterprise BRMS

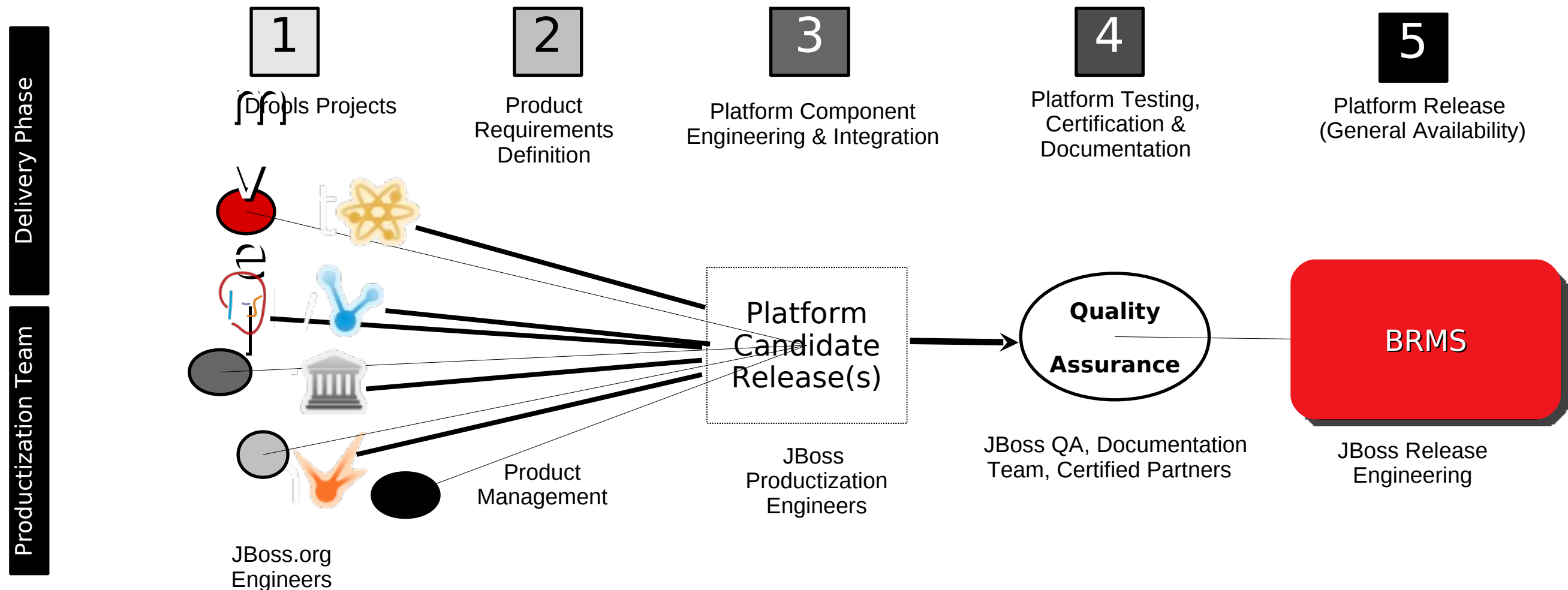
- Single, integrated, certified distributions
- Extensive Q/A Process
- Industry-leading Support
- Documentation
- Secure, Production-level Configurations
- Multi-year Errata Policy



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

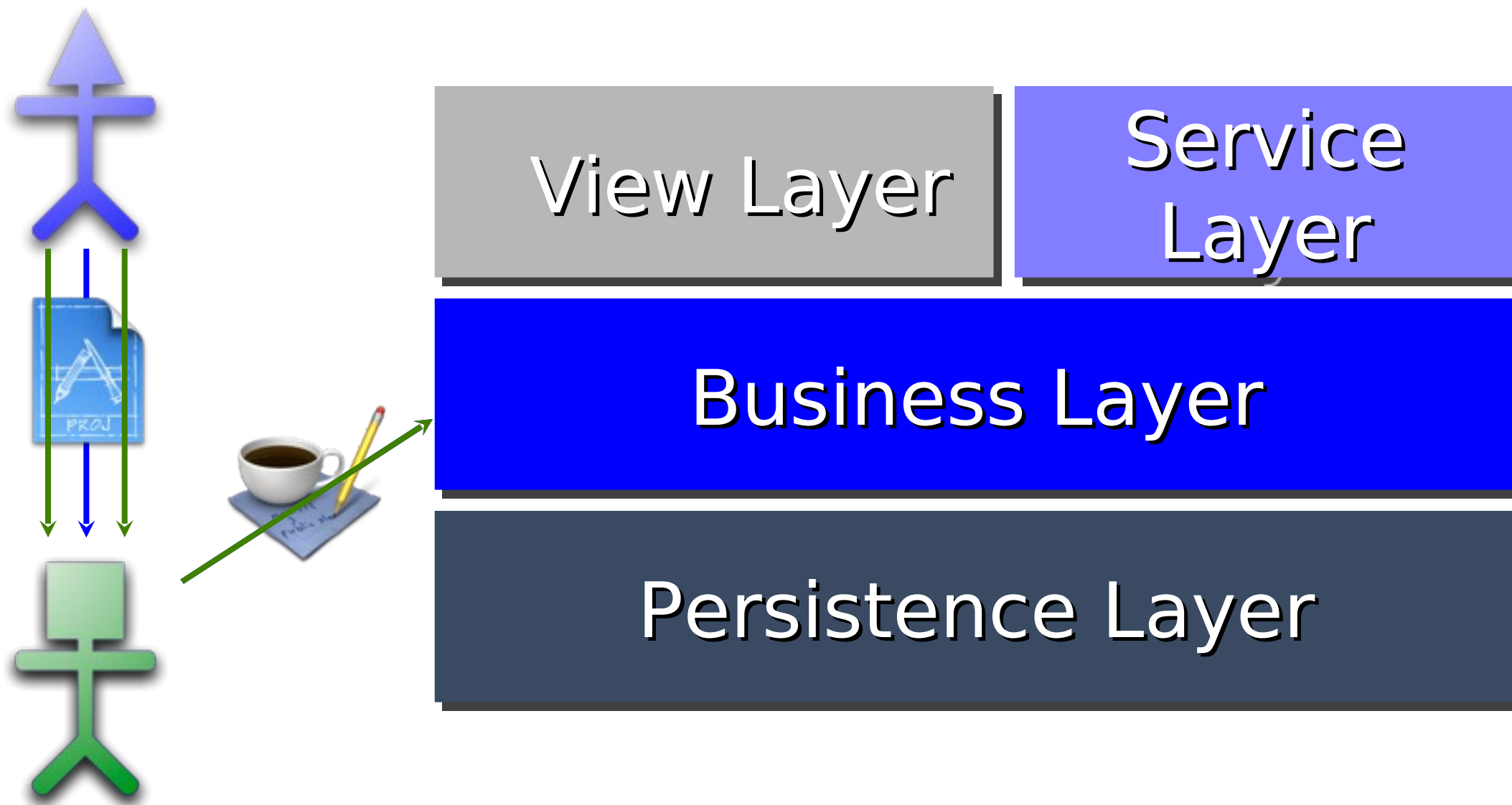
JBoss Enterprise BRMS

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



The Concepts

The Typical SDLC



Extract the Logic

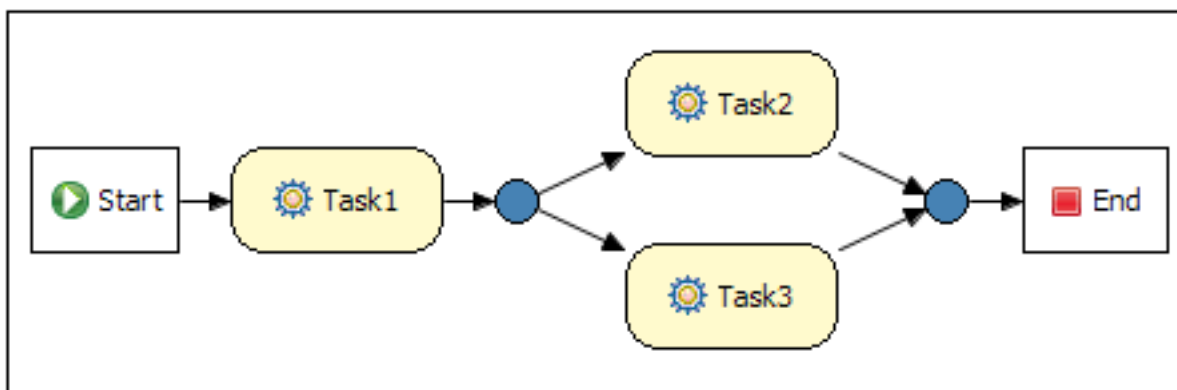
- Separate integration codebase from business logic and into a rules engine
- **Goal:** Remove the requirements churn and complexity
- **Goal:** Empower your Domain Experts to directly author rules

Business Logic

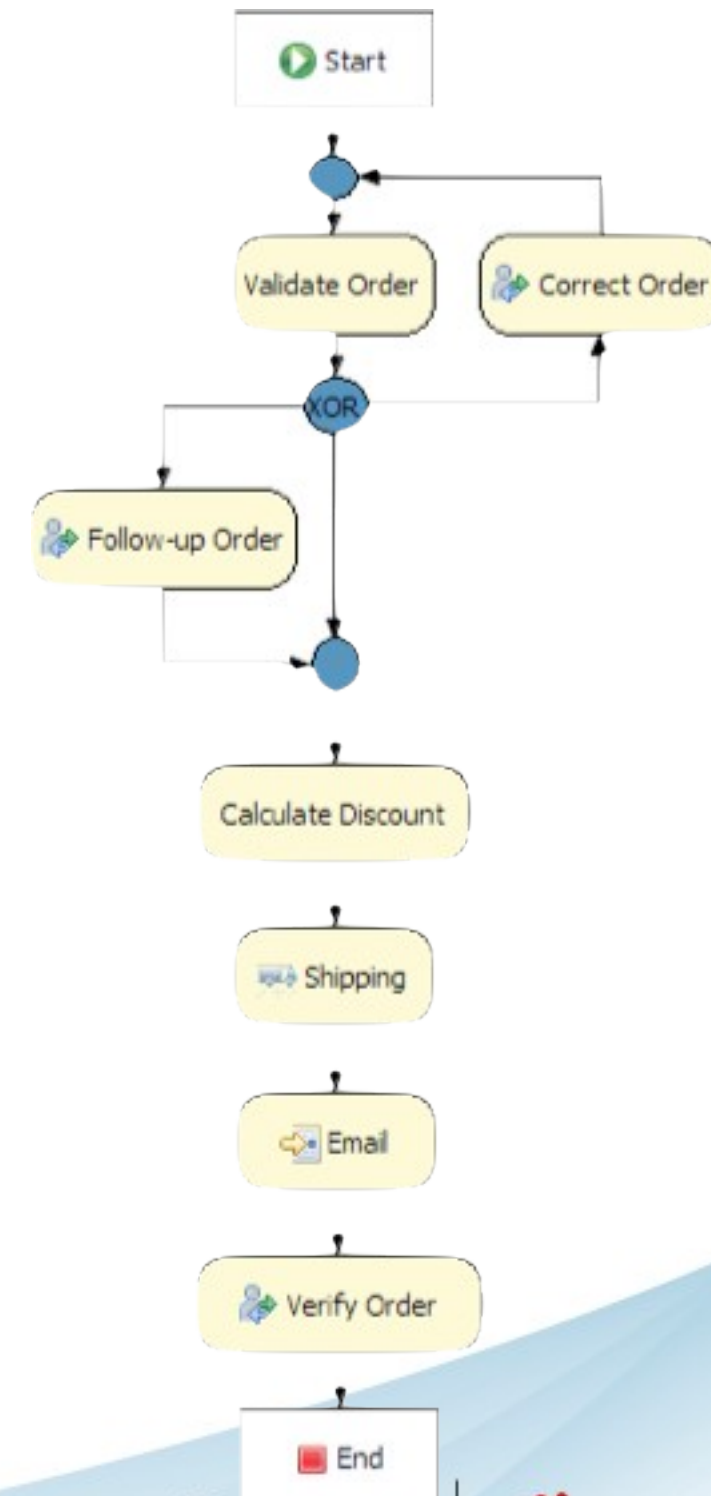
- Rules
- Process
- Response to Events
- Planning Problems

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
```



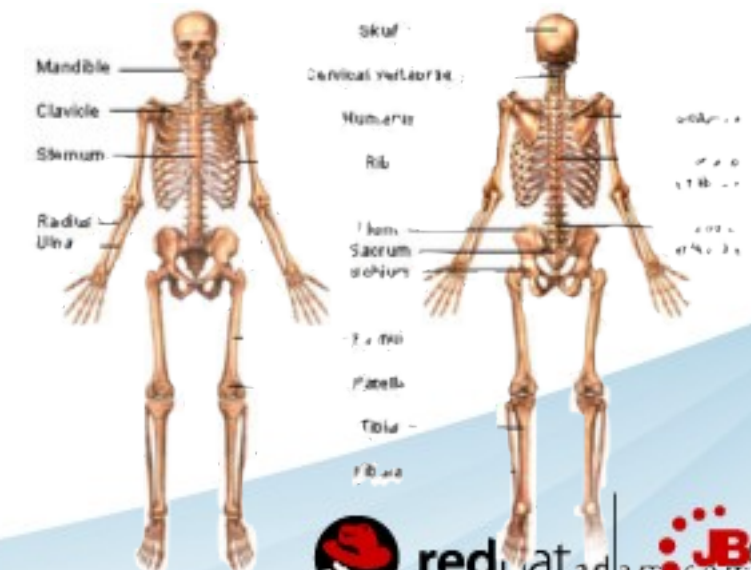
What is a Rules Engine?

- 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


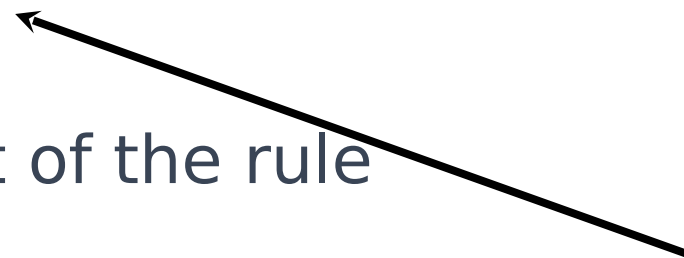
```
public class Room {  
    private String name  
    // getter and setter methods here  
}  
public class Sprinkler {  
    private Room room;  
    private boolean on;  
    // getter and setter methods here  
}  
public class Fire {  
    private Room room;  
    // getter and setter methods here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

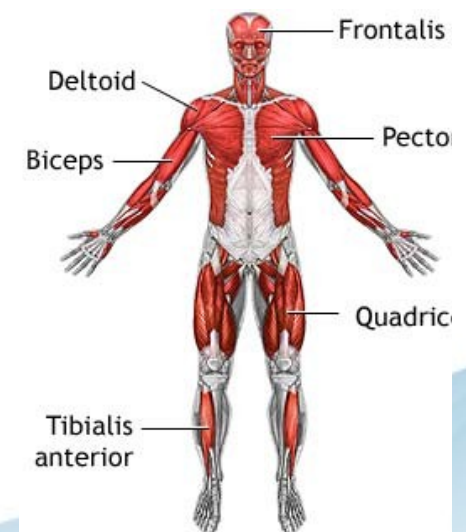
Model is multiple facts and their relationships

- 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!

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

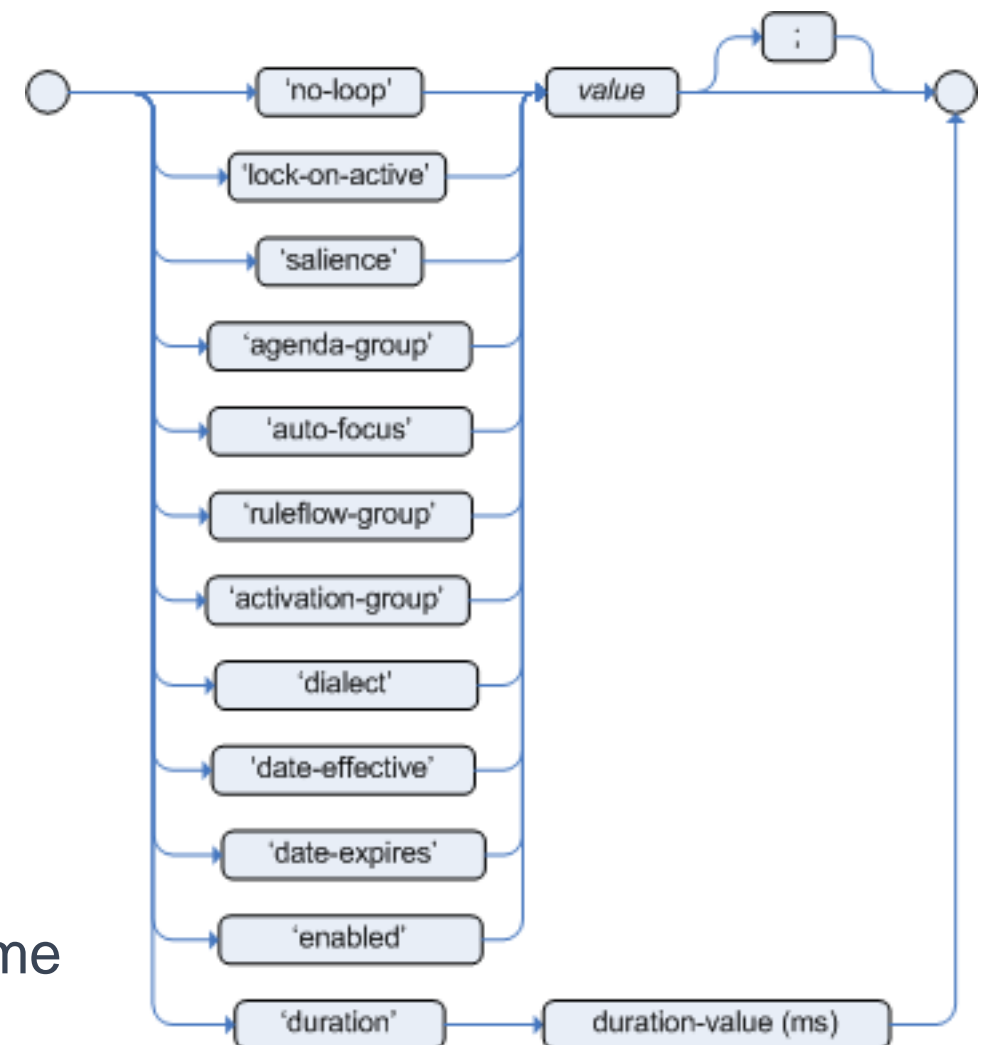
rule "When the fire is gone turn off the sprinkler"
dialect "mvel"
when
    $room : Room()
    $sprinkler : Sprinkler( room == $room, on == true )
    not Fire( room == $room )
then
    modify( $sprinkler ) { on = false };
    println( "Turn off the sprinkler for room " + $room.name );
end
```


Conditional Elements

- **and**
 - ✓ all attributes match (default of comma separated list)
- **or**
 - ✓ either attribute matches
- **eval**
 - ✓ catch all element
 - ✓ wraps any primitive 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 retrieve data from external sources like DBs, WebServices, etc.
- **matches, soundlike,**
 - ✓ 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 Browser

- 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

BRMS – Rich UI for Business Rules

The screenshot displays the JBoss BRMS (Business Rules Management System) interface, which is a rich UI for managing business rules. The interface is divided into several sections:

- Left Panel (Navigation):** Contains a tree view of the project structure, including Assets, Packages, and various rule assets like Business rule assets, Technical rule assets, Functions, DSL configurations, Model, Rule Flows, Enumerations, Test Scenarios, XML, Properties, and Other assets, documentation. It also includes QA, Package snapshots, and Administration links.
- Top Panel (Model Editor):** Shows the 'Simple Insurance Model' with a 'Driver' and 'Policy' section. The 'Driver' section includes fields like locationRiskProfile, eyeColor, age, priorClaims, and name. The 'Policy' section includes fields like basePrice, approved, type, and discountPercent. Below the model editor, there are dropdown menus for 'age' (greater than or equal to 18 and less than (or equal to) 24), 'locationRiskProfile' (is equal to Low), and 'priorClaims' (is equal to 0). The 'Policy' section has a dropdown for 'type' (is equal to COMPREHENSIVE). Below these, there is a 'THEN' section with a text area containing the rule logic: 'Driver's age between 18 and 24 inclusive, with a low risk profile, no prior requesting a comprehensive policy'. There is also a 'Set value of policy basePrice 455' field.
- Right Panel (Scenario Runner):** Contains a 'Run scenario' section with a 'Results' bar showing 50% success. Below this is a 'Summary' section with a warning icon and text: '[adv] field [valueReason] was [Loyal] expected [Other]'. There is a '+GIVEN' section with an 'insert [Advertiser]' button and a '+EXPECT' section with a 'Use real date and time' button. Below these are input fields for 'Advertiser [adv]' with values: advertiserType: Agency, ageOfCustomer: 12, postcode: 4064. There are also input fields for 'valueScore' (equals 42) and 'valueReason' (equals Other, with a note '(Actual: Loyal)'). A 'More...' button is also present. Below the input fields is a 'configuration' section with a dropdown for 'All rules may fire' and a '+ (globals)' button.
- Bottom Right Panel (Test Results):** Contains a 'Scenarios for package: Sensis' section with a 'Run all scenarios' button. Below this is a 'Overall result: FAILURE' section with a 'Results' bar showing 90% success (1 failures out of 23 expectations). There is a 'Rules covered' bar showing 90% of the rules were tested. Below these are 'Uncovered rules' and a 'Scenarios' table with a list of scenarios and their results.

Scenario	Results	Failures	Open
Adv scoring sanity test	100%	0 failures out of 2	Open
Sydney advertiser ratings	100%	0 failures out of 3	Open
Basic assignment validation	100%	0 failures out of 4	Open
Overlapping assignments	100%	0 failures out of 5	Open
Allow email overlap assignments	100%	0 failures out of 5	Open
Consultant self assign authority	100%	0 failures out of 1	Open
QLD self assign	100%	0 failures out of 1	Open
No available consultants	100%	0 failures out of 2	Open

Fact Model Editor

The screenshot shows the JBoss BRMS Fact Model Editor interface within a Mozilla Firefox browser window. The title bar reads "JBoss BRMS - Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". A top bar displays "Welcome: admin" and a "[Sign Out]" link.

Left Panel (Navigate):

- Assets (+)
- Packages (-)
- Create New ▾
- Packages
 - insurance
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model**
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - order_processing
- QA (+)
- Package snapshots (+)
- Administration (+)

Main Panel:

Find: Model [insurance] Simple Insuranc

Save changes | Copy | Archive | Change status | Status: [Draft]

Driver

Add field

- locationRiskProfile: Text
- eyeColor: Text
- age: Whole number (integer)
- priorClaims: Whole number (integer)
- name: Text

Policy

Add field

- basePrice: Whole number (integer)
- approved: True or False
- type: Text
- discountPercent: Whole number (integer)

Add new fact type

View source | Validate

The basic Insurance model

Title: [Simple Insurance Model]
[show more info...]

Close all items

Guided Rule Editor (Web)

The screenshot shows the JBoss BRMS Web Guided Rule Editor interface. The browser window title is "JBoss BRMS - Mozilla Firefox". The top navigation bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". A user greeting "Welcome: admin [Sign Out]" is visible in the top right.

Navigate Panel (Left):

- Assets (+)
- Packages (-)
- Create New ▾
- Packages
 - insurance
 - Business rule assets (selected)
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - order_processing
- QA (+)
- Package snapshots (+)
- Administration (+)

Main Editor Area:

Find: Business rule asset Safe Youth

Save changes | Copy | Archive | Change status | Status: [Draft]

WHEN

Driver

- age: greater than or equal to 18 and less than (or equal to) 24
- locationRiskProfile: is equal to Low
- priorClaims: is equal to 0

Policy [policy]

- type: is equal to COMPREHENSIVE

THEN

- Set value of policy basePrice 455

(options)

View source | Validate

Driver's age between 18 and 24 inclusive, with a low risk profile, no prior claims and requesting a comprehensive policy

Close all items

Web Decision Table Editor

JBoss BRMS - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Welcome: admin [Sign Out]

Navigate

- Assets
- Packages
- Create New
- Packages
 - insurance
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
 - mortgages
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Rule Flows
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentation
- QA
- Package snapshots
- Administration

Find Pricing loans

Save changes

Decision table

Modify...

	Desc	amount min	amount max	period	income	deposit max	Loan approved	LMI
income: Asset (1 Item)								
3		131000	200000	30	Asset	20000	true	0
income: Job (2 Items)								
1		10000	100000	20	Job	2000	true	0
2		100001	130000	20	Job	3000	true	10

Close all items

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

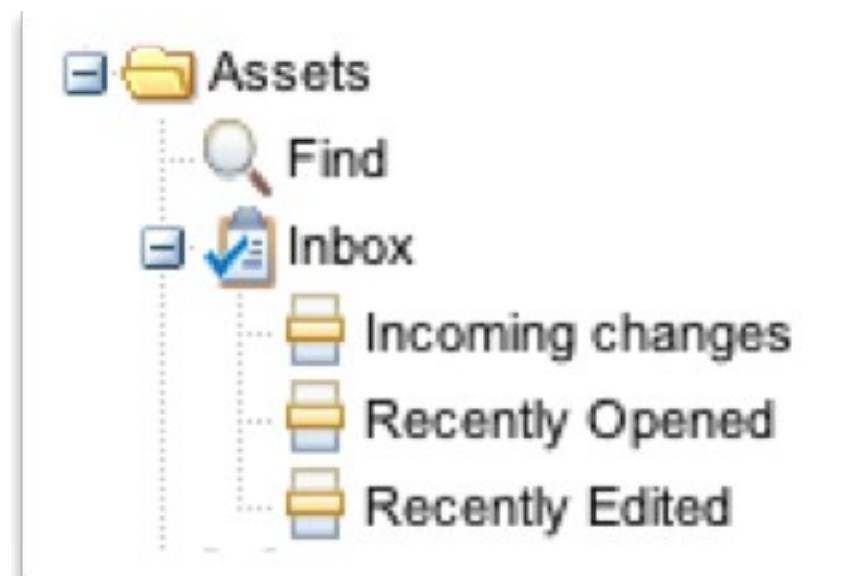
Description:

Discussion:

Comment by alan_parsons on Fri Aug 07 13:50:18 GMT+1000 2009:
Hey all !

Comment by alan_parsons on Fri Aug 07 13:51:42 GMT+1000 2009:
This is more discussion - are we in agreement?

[Add a discussion comment](#) [Erase all comments](#) 



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-brms/org.drools.guvnor.Guvnor/package/mortgages/LATEST/SCENARIOS> ⓘ

Change Set: <http://127.0.0.1:8080/jboss-brms/org.drools.guvnor.Guvnor/package/mortgages/LATEST/SCENARIOS>

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

The screenshot shows a business rule editor interface. At the top, there are tabs for 'Find', 'CreditApproval', 'mortgages', 'Business rule asset', and 'Pricing loans'. Below the tabs, there are buttons for 'Save changes' and 'Save and close'. On the right, there are buttons for 'Validate', 'Verify', 'View source', 'Actions...', and 'Status:'. A red arrow points from the title 'Single Asset Verification' to the 'Verify' button.

The main area displays a 'Decision table' with a 'Modify' dropdown. The table has columns: 'Row N...', 'amount min', 'amount max', 'period', 'income', 'deposit max', 'Loan approved', 'LMI', and 'rate'. There are three rows of data:

Row N...	amount min	amount max	period	income	deposit max	Loan approved	LMI	rate
1	131000	200000	30	Asset	2000			
2	10000	100000	20	Job	2000			
3	100001	130000	20	Job	3000			

Below the table, there are sections for 'income: Asset (1 Item)' and 'income: Job (2 Items)'. A 'Verification report' dialog is open in the foreground, showing the following details:

Verification report

- Errors (0 items).**
- Warnings (11 items).**
 - Rule 'RegexDslRule' has no RHS.
 - Rule 'Dummy rule' has no RHS.
 - Rule base covers == 30, but it is missing != 30
 - Rule base covers == Asset, but it is missing != Asset
 - Reason:** LiteralRestriction from rule [Row 3 Pricing loans] value '== Asset'
 - Impacted rules:**
 - Rule base covers == Job, but it is missing != Job
 - Rule base covers == 20, but it is missing != 20
 - Rule base covers == Job, but it is missing != Job
 - Rule base covers == 20, but it is missing != 20
 - Rule base covers == Sub prime, but it is missing != Sub prime
 - Rule base covers == OK, but it is missing != OK
 - Rule base covers == OK, but it is missing != OK
- Notes (0 items).**

Multiple View / Edits in a Single Tab

Find

CreditApproval

Business rule asset

[No bad credit cl

Save all changes Save and close all Show ▾

No bad credit checks

Save changes Save and close

Select Working Sets Validate

WHEN

1. There is a LoanApplication [app]

Any of the following are true:

There is an Applicant with:

creditRating equal to OK

2. There is an Applicant with:

creditRating equal to Sub prime

THEN

1. Set value of LoanApplication [app] approved false

Set value of LoanApplication [app] explanation Only AA

2. Retract LoanApplication [app]

(show options...)

Underage

Save changes Save and close

Select Working Sets Validate

WHEN

1. There is a LoanApplication [application]

There is an Applicant with:

2. age less than 21

THEN

1. Set value of LoanApplication [application] approved false

Set value of LoanApplication [application] explanation Underage

2. Retract LoanApplication [application]

(show options...)

From, Collect & Accumulate Support

WHEN

1. There is a Hospital [\$h]
2. There is a Bed with:
status equal to Break Not Set
From \$h.beds. Choose...

THEN

(show options...)

From CE

WHEN

1. There is a Hospital [\$h]
2. From Collect
All Bed with:
status equal to Break Not Set

THEN

(show options...)

From Collect CE

WHEN

1. There is a Hospital [\$h]
2. There is a Number with:
doubleValue greater than 0
From Accumulate
All Bed [\$b] with:
status equal to Break Not Set

Custom Code Function

Function: sum (\$b)

THEN

(show options...)

From Accumulate CE

Rule Template Support

Template Editor Template Data

WHEN

There is an Applicant with:

1. name matches applicant_name

approved equal to

THEN
(show options...)

Field value

Literal value: Literal value

Template key: Template key

Advanced options:

A formula: New formula

Expression editor: Expression editor

Template Editor **Template Data**

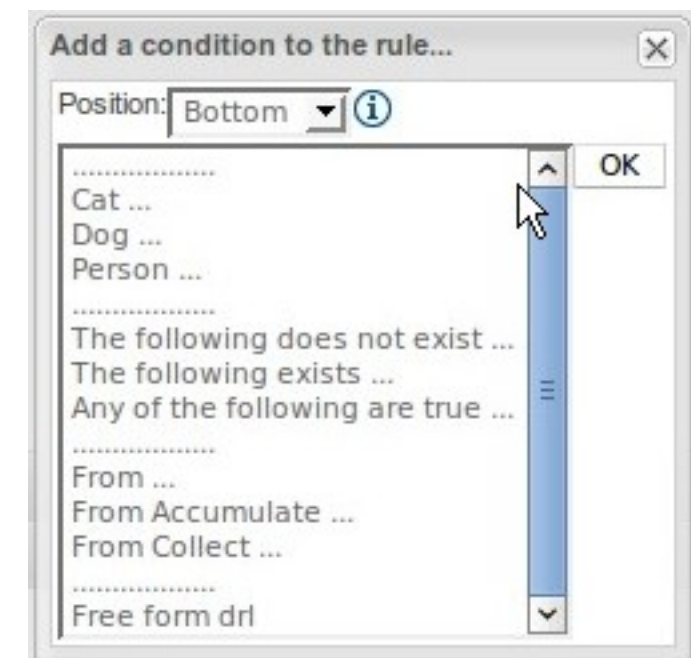
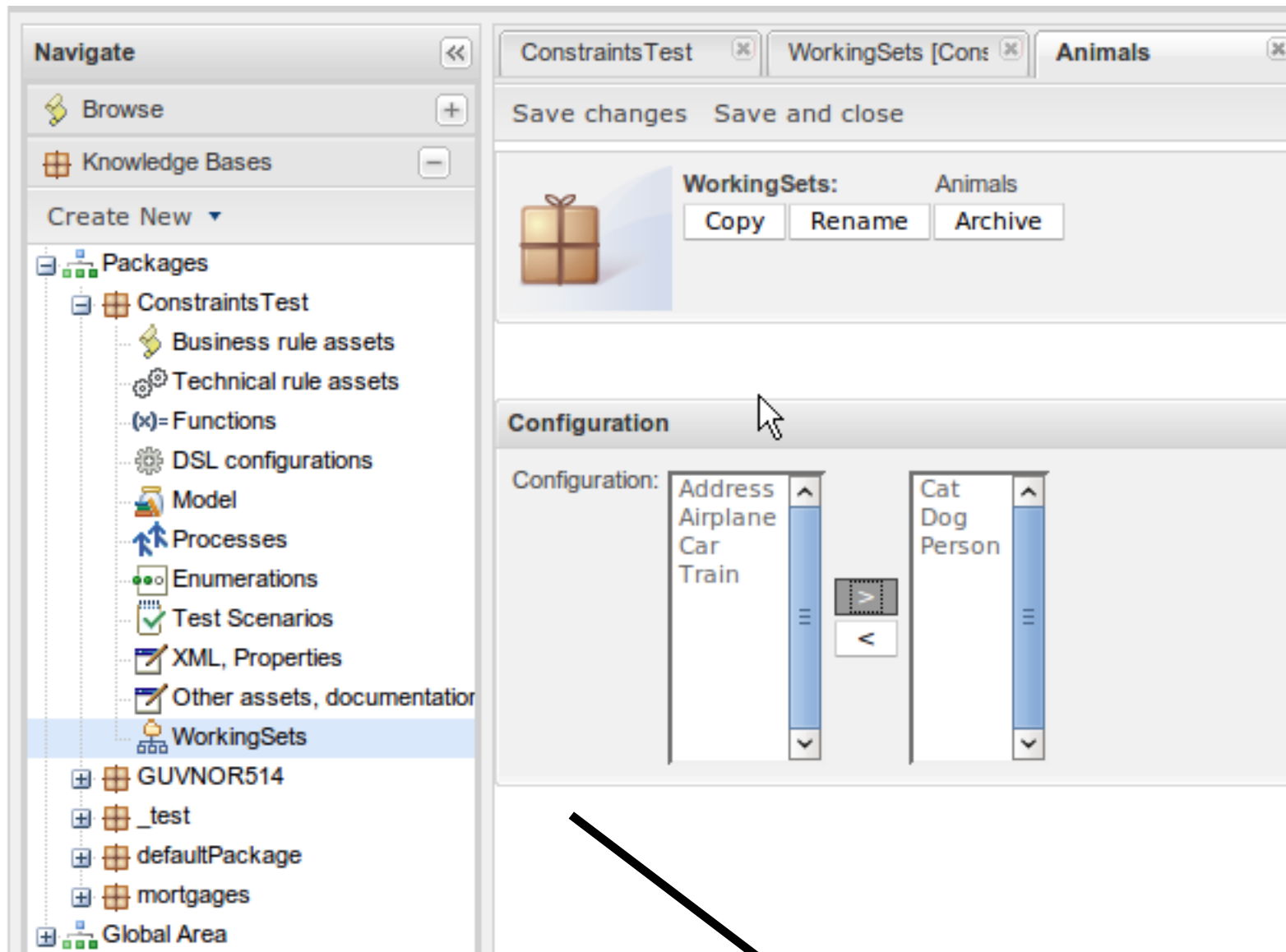
Modify

Add row...

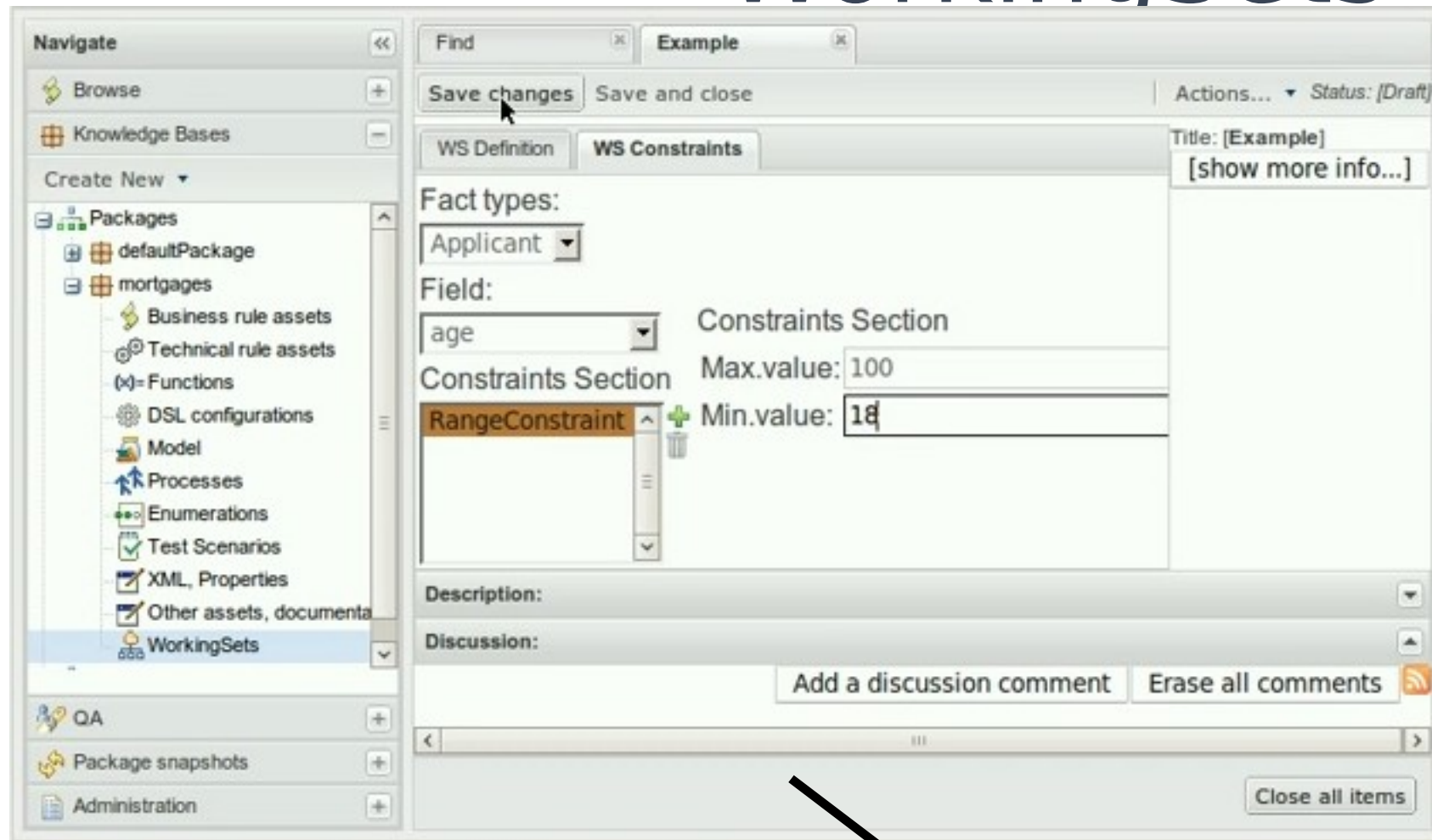
Remove selected row(s)...

	is_approved	amount
baunax	true	400
	false	450

Working Sets



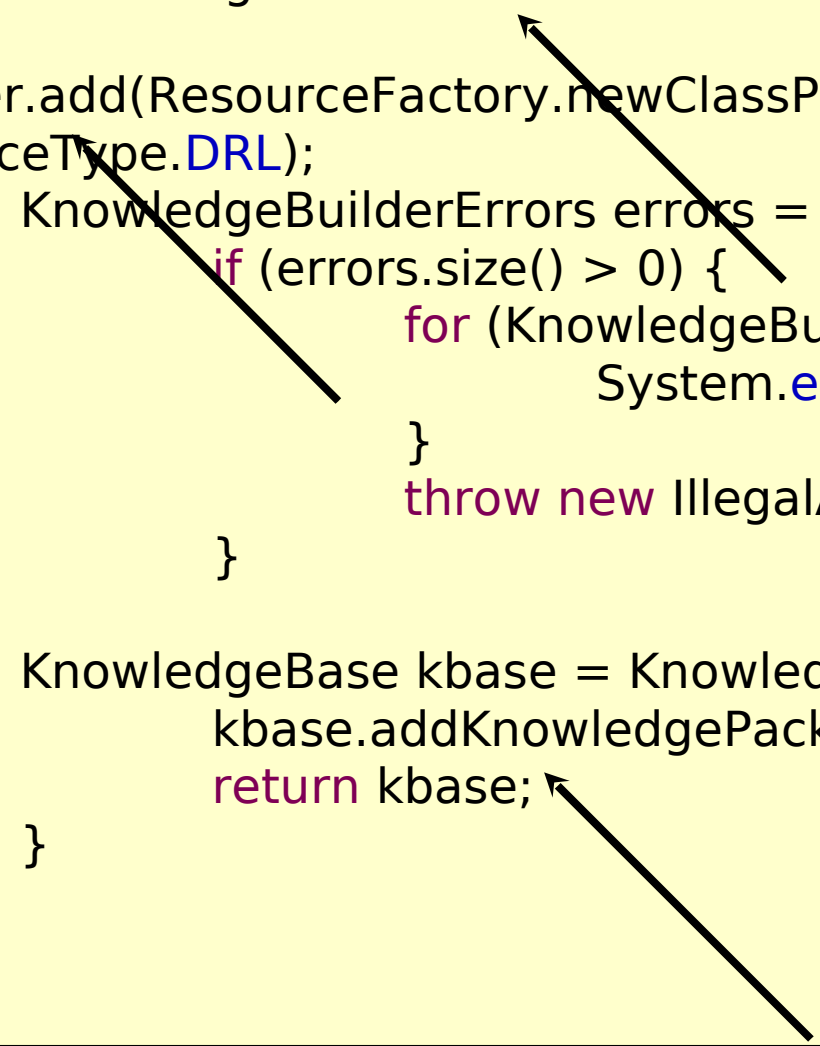
Fact Constraints within WorkingSets



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"
       xsi:schemaLocation="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" />
        </drools:configuration>
    </drools:kbase>
</beans>
```

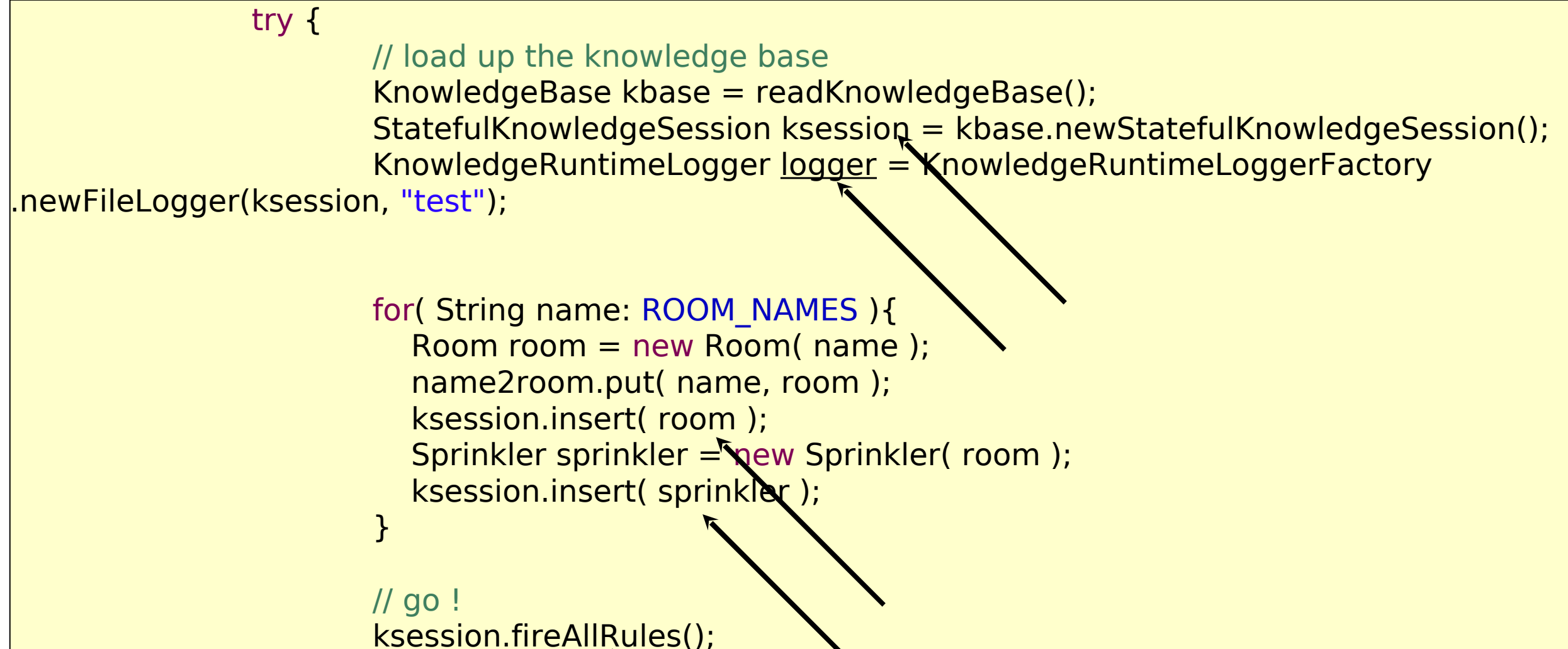
THE BRE now has extensive Spring support, the XSD can be found in the the drools-spring jar.

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();  
}
```

A diagram illustrating the execution flow of the provided Java code. Arrows point from the following elements to the `ksession.fireAllRules();` line: the `logger` variable, the `new Sprinkler(room);` line, the `new Room(name);` line, and the `ROOM_NAMES` constant.

> 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

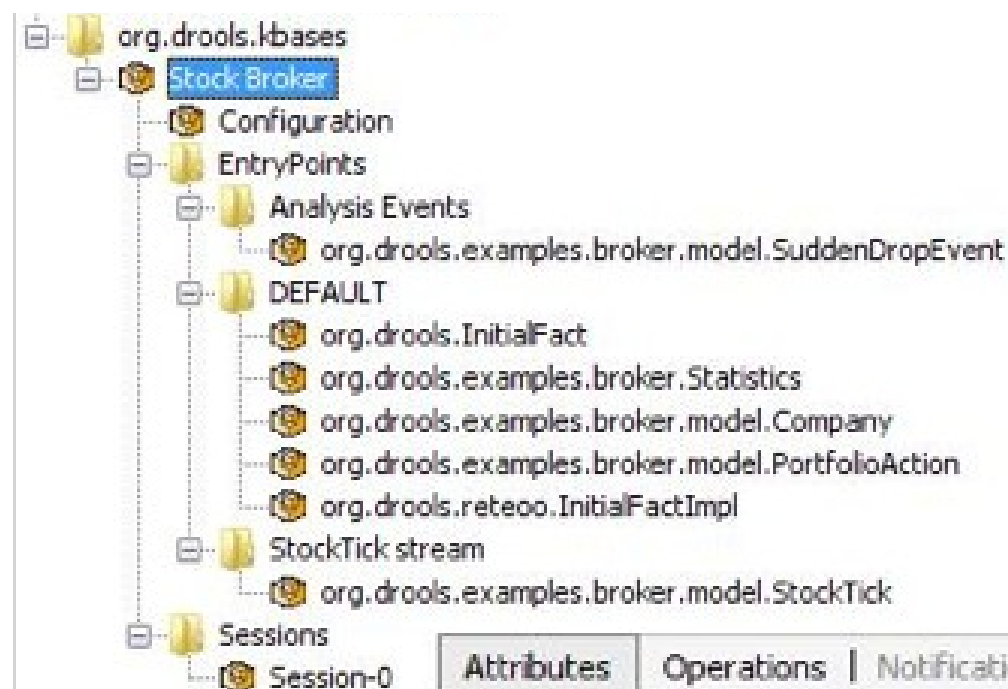
Use FROM for external data

```
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.

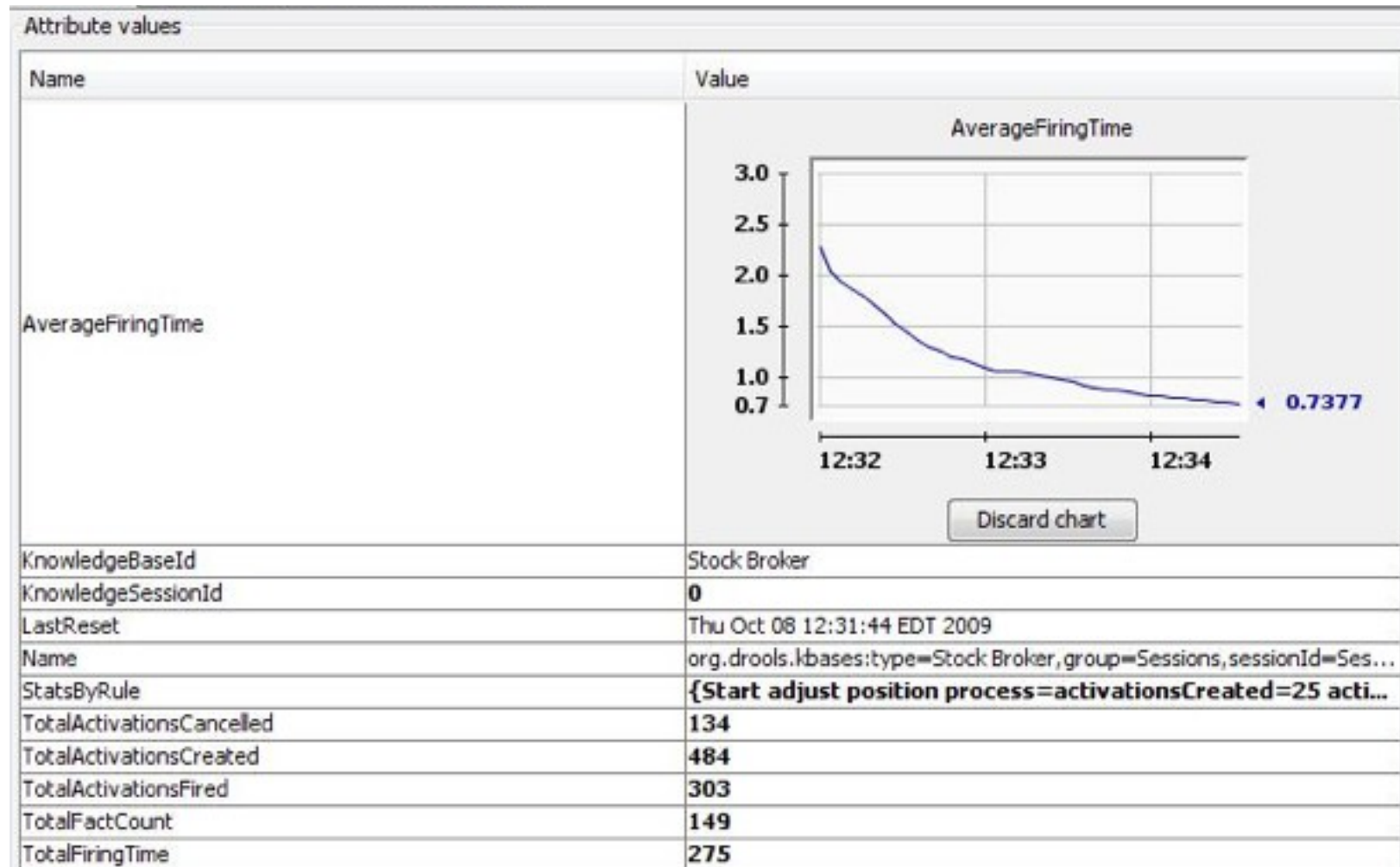
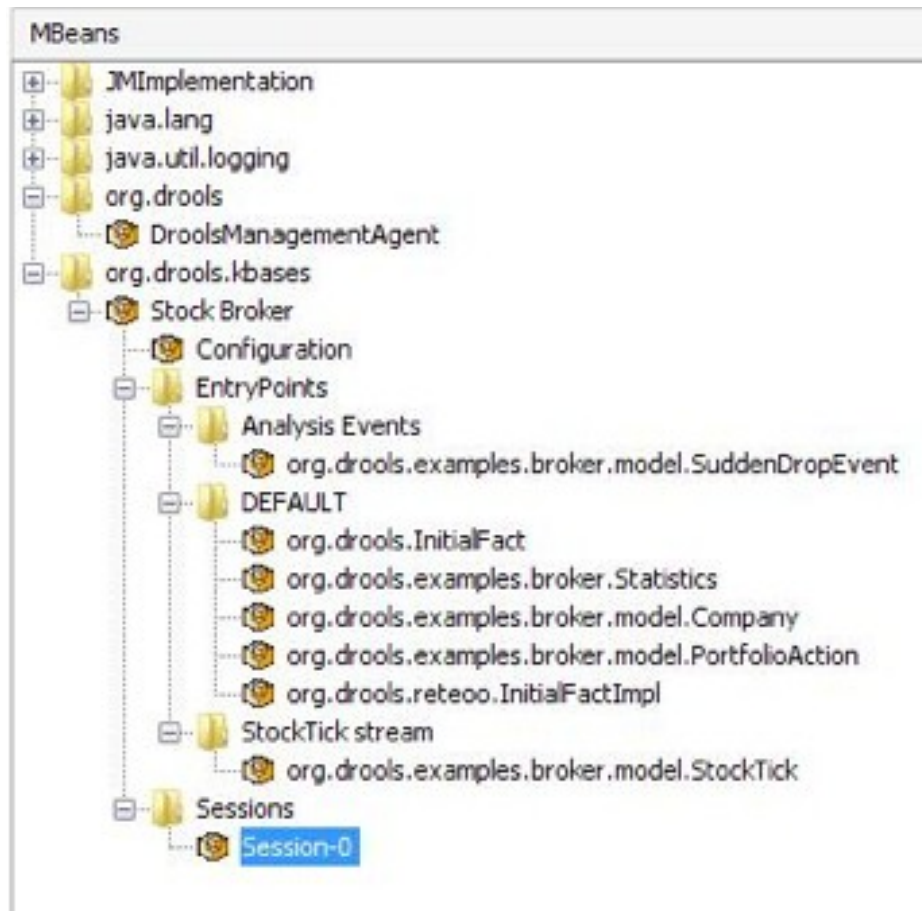


KnowledgeBase Stats

Attributes	Operations	Notifications	Metadata
Attribute values			
Name	Value		
EntryPoints	[EntryPoint::DEFAULT, EntryPoint::Analysis Events, EntryPoint::Stoc...		
Globals	{services=interface org.drools.examples.broker.BrokerServices}		
Id	Stock Broker		
Packages	[org.drools.examples.broker, org.drools.examples.broker.model]		
SessionCount	1		

API Enhancements - JMX

KnowledgeSession Stats





Deployment <ul style="list-style-type: none">- Packaging- Snapshots- Status Definition- User Administration- Categorization	<ul style="list-style-type: none">- Asset Search- Browse by Category- Browse by Status
Analysis <ul style="list-style-type: none">- Step-debugging- Agenda Views and Inspection- Working Memory Inspection- Rule Engine Audit Trails	<ul style="list-style-type: none">- Rule Authoring Analysis- Guided Test Audit Trails
Tests <ul style="list-style-type: none">- Technical Tests- Technical Test Suites	<ul style="list-style-type: none">- Guided Tests- Scenario Tests
Rules <ul style="list-style-type: none">- Technical Rules (.drl)- Enumerations- Domain Specific Language- Templates- Rule Flow	<ul style="list-style-type: none">- Decision Tables- Decision Tables- Guided Rules- English/Industry Specific Rules
Facts <ul style="list-style-type: none">Java Jars	<ul style="list-style-type: none">Dynamic Facts

Developer-centric

Business Analyst

Web-only Feature	Modifiable via Developer IDE
Modifiable via Web Browser or Developer IDE	Modifiable via Excel/OpenOffice



Eclipse – Developer Perspective

Debug - StateExampleUsingSaliency.drl - Eclipse SDK

File Edit Navigate Search Project Run Window Help

100%

Debug

StateExampleUsingSaliency [Drools Application]

org.drools.examples.StateExampleUsingSaliency at localhost:4861

Thread [main] (Suspended (breakpoint at line 8 in Rule_A_to_B_0))

Rule_A_to_B_0.consequence(KnowledgeHelper, State, FactHandle) line: 21

Rule_A_to_B_0ConsequenceInvoker.evaluate(KnowledgeHelper, WorkingMemory) line: 22

DefaultAgenda.fireActivation(Activation) line: not available

DefaultAgenda.fireNextItem(AgendaFilter) line: not available

ReteooWorkingMemory(AbstractWorkingMemory).fireAllRules(AgendaFilter) line: not available

ReteooWorkingMemory(AbstractWorkingMemory).fireAllRules() line: not available

StateExampleUsingSaliency.main(String[]) line: 47

Variables

Breakpoints

Name Value

b State (id=1268)

changes PropertyChangeSupport (id=1297)

name "B"

state 1

StateExampleUsingSaliency.drl

```
import org.drools.examples.State;

rule Bootstrap
when
    a : State(name == "A", state == State.NOTRUN )
then
    System.out.println(a.getName() + " finished" );
    a.setState( State.FINISHED );
end

rule "A to B"
when
    State(name == "A", state == State.FINISHED )
    b : State(name == "B", state == State.NOTRUN )
then
    b.setState( State.FINISHED );
    System.out.println(b.getName() + " finished" );
end

rule "B to C"
    salience 10
when
    State(name == "B", state == State.FINISHED )
    c : State(name == "C", state == State.NOTRUN )
then
    System.out.println(c.getName() + " finished" );
end
```

Text Editor Rete Tree

Text Editor Rete Tree

Properties

Outline

org.drools.examples

A to B

B to C

B to D

Bootstrap

org.drools.examples.State

Global Data View

The selected working memory has no globals defined.

Audit View

Object asserted (1): A[NOTRUN]

Activation created: Rule Bootstrap a=A[NOTRUN](1)

Object asserted (2): B[NOTRUN]

Object asserted (3): C[NOTRUN]

Object asserted (4): D[NOTRUN]

Activation executed: Rule Bootstrap a=A[NOTRUN](1)

Object modified (1): A[FINISHED]

Activation created: Rule A to B b=B[NOTRUN](2)

Activation executed: Rule A to B b=B[NOTRUN](2)

Object modified (2): B[FINISHED]

Activation created: Rule B to C c=C[NOTRUN](3)

Activation created: Rule B to D d=D[NOTRUN](4)

Activation executed: Rule B to C c=C[NOTRUN](3)

Object modified (3): C[FINISHED]

Activation executed: Rule B to D d=D[NOTRUN](4)

Object modified (4): D[FINISHED]

Agenda View

MAIN[focus]= AgendaGroupImpl (id=1259)

[0]= AgendaItem (id=1262)

ruleName= "B to C"

c= State (id=1269)

[1]= AgendaItem (id=1263)

ruleName= "B to D"

d= State (id=1270)

Working Memory View

[0]= State (id=1268)

[1]= State (id=1269)

FINISHED= 1

NOTRUN= 0

changes= PropertyChangeSupport (id=1294)

name= "C"

state= 0

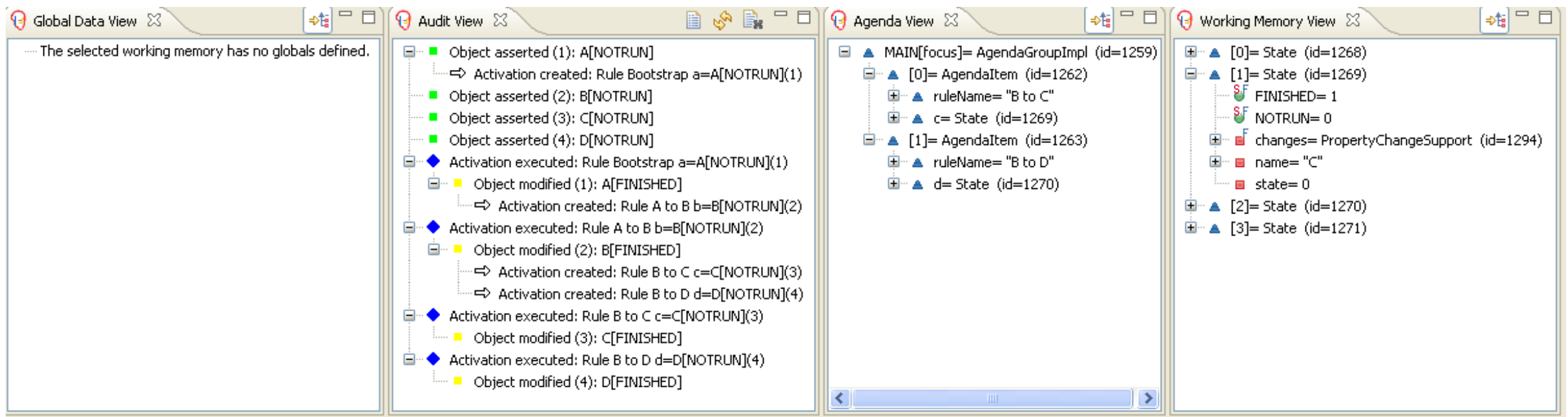
[2]= State (id=1270)

[3]= State (id=1271)

Writable Insert 21 : 1

Debugging

Debug Views



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

Guided Rule Editor (Eclipse)

Eclipse Guided Editor

The image displays two overlapping windows from the Eclipse IDE's Guided Rule Editor.

The main window, titled "Rule Builder", shows a rule configuration interface. It has tabs for "rules.brl" and "rule.package". The "WHEN" section is active, showing a list of conditions. The first condition is "There is no" followed by a red 'X' and a green '+' icon. Below this, there are three groups of conditions, each with a red 'X' and a green '+' icon:

- Person**:
 - age [dropdown] (x)= [text box]
 - [checkbox] [dropdown] (x)= [text box]
 - age [checkbox] [dropdown] [text box]
 - gender [checkbox] [dropdown] (x)= [text box]
 - gender [checkbox] [dropdown] [text box]
 - gender [checkbox] [dropdown] [text box]
- Elephant**:
 - nickname [checkbox] [dropdown] [text box]
 - All of: [checkbox] [text box]
 - age [checkbox] [dropdown] [text box]
 - weight [checkbox] [dropdown] [text box]
- Animal**:
 - (x)= [text box]
 - name [checkbox] [dropdown] (x)= [text box]

The "THEN" section is visible at the bottom, showing a "Person" condition with a green '+' icon and a text box.

The second window, titled "Business Rule XML Editor", also shows a "Rule Builder" interface. It has a tab for "Business Rule XML Editor". The "IF" section is active, showing a list of conditions:

- Person**:
 - age [dropdown] is less than [dropdown] 42 [text box]
 - name [dropdown] is equal to [dropdown] Bob [text box]
- Vehicle [car1]**:
 - type [dropdown] is not equal to [dropdown] [text box]

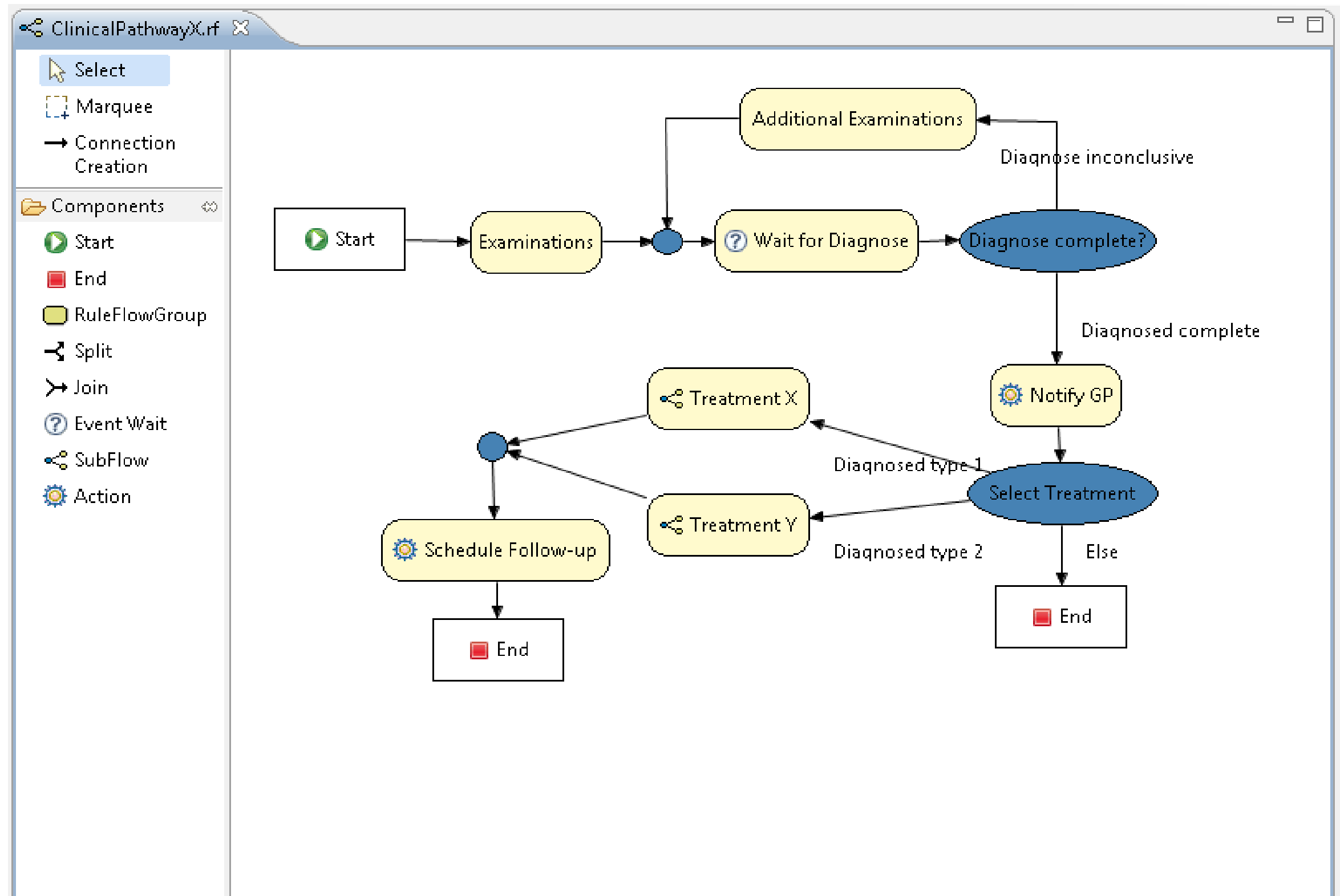
Below the conditions, there are two lines of text:

- There is a Storm alert of type (code here)
- severity rating is not more than (code here)

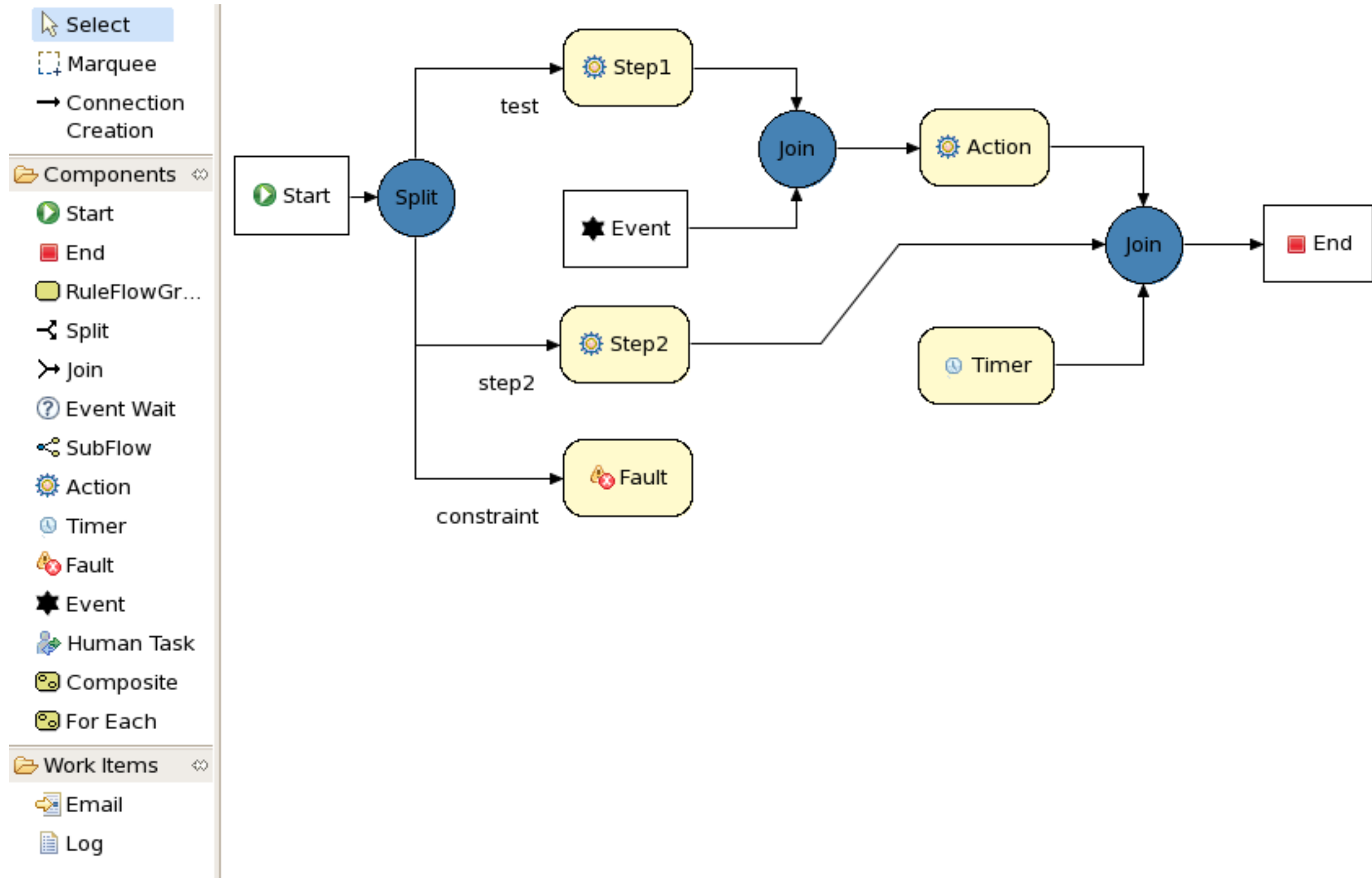
The "THEN" section is visible at the bottom, showing a "Person" condition with a green '+' icon and a text box.

At the bottom of the "Business Rule XML Editor" window, there are tabs for "Rule Builder" and "DRL Preview".

Rule Flow

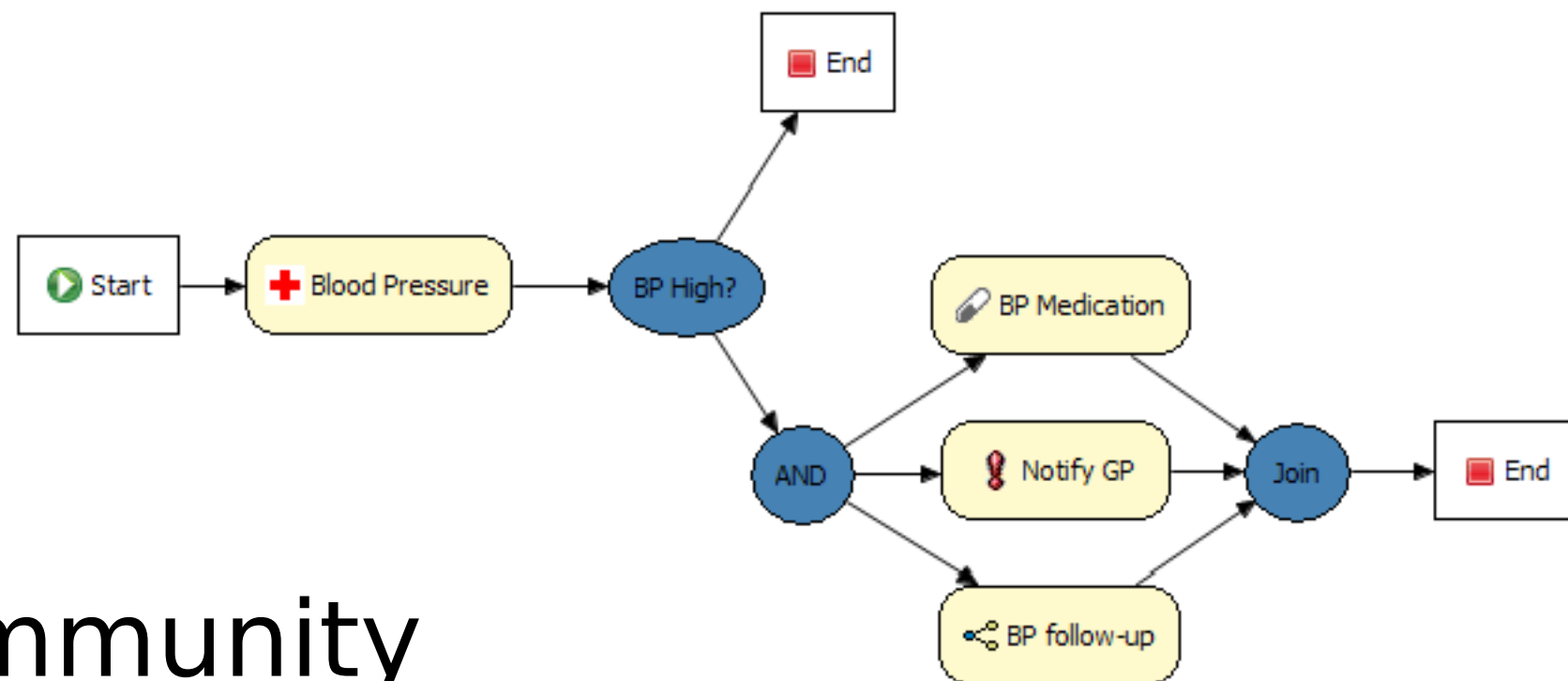


Node Types



Domain Specific Process Steps

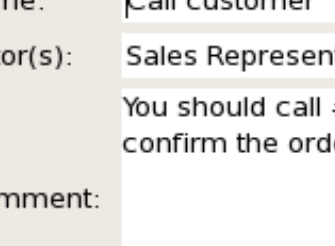
- Extend Drools Flow to incorporate your application's needs



Note: Community
Only

Human Based Tasks

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



The screenshot shows a dialog box titled "Human Task Editor" with a close button (X) in the top right corner. The dialog contains several fields and a checkbox:

- Name:** A text field containing "Call customer".
- Actor(s):** A text field containing "Sales Representative".
- Comment:** A text area containing the text "You should call #{customer.name} to confirm the order.".
- Priority:** A text field containing the number "3".
- Skippable:** A checkbox that is checked, followed by the text "Skippable".
- Content:** A large empty text area.

At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Properties

Property	Value
ActorId	Sales Representative
Comment	You should call #{customer.name} to confirm the order.
Content	
Id	4
Name	Human Task
On Entry Actions	
On Exit Actions	
Parameter Mapping	{}
Priority	3
Result Mapping	{}
Skippable	true
Swimlane	
TaskName	Call customer
Timers	
Wait for completion	true

Human Task View

UserId sales-rep

Name

Some Task

Some other task

Claim

Start

Stop

Release

Human Task View

Userid

sales-rep

Refresh

Create

Name	Status	Owner	Created On	Comment
Some Task	InProgress	sales-rep	r 31, 2009 4:44:22	
Some other task	Reserved	sales-rep	r 31, 2009 4:45:02	

Claim

Start

Stop

Release

Suspend

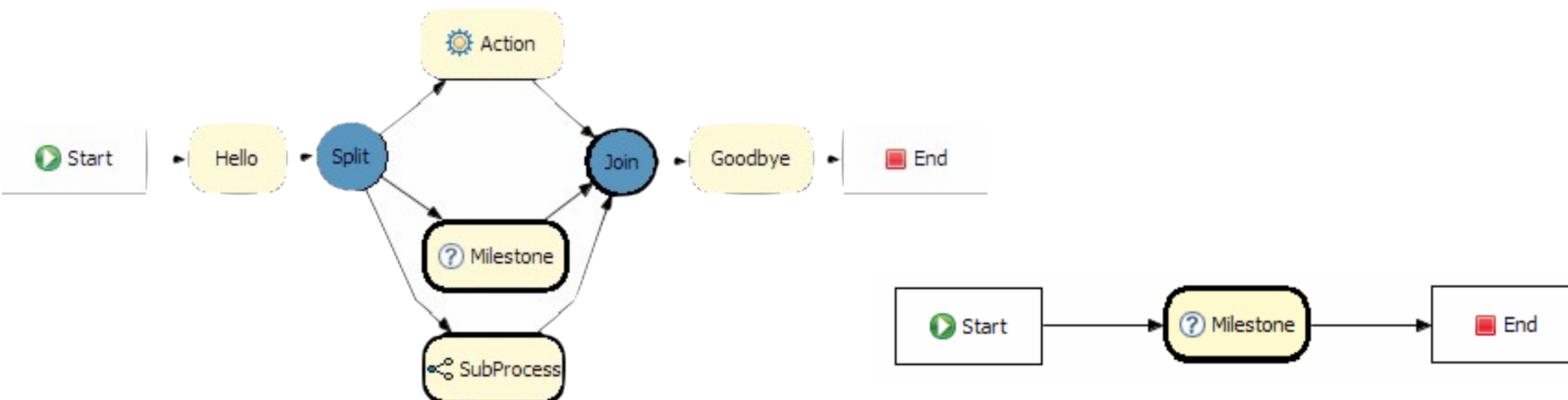
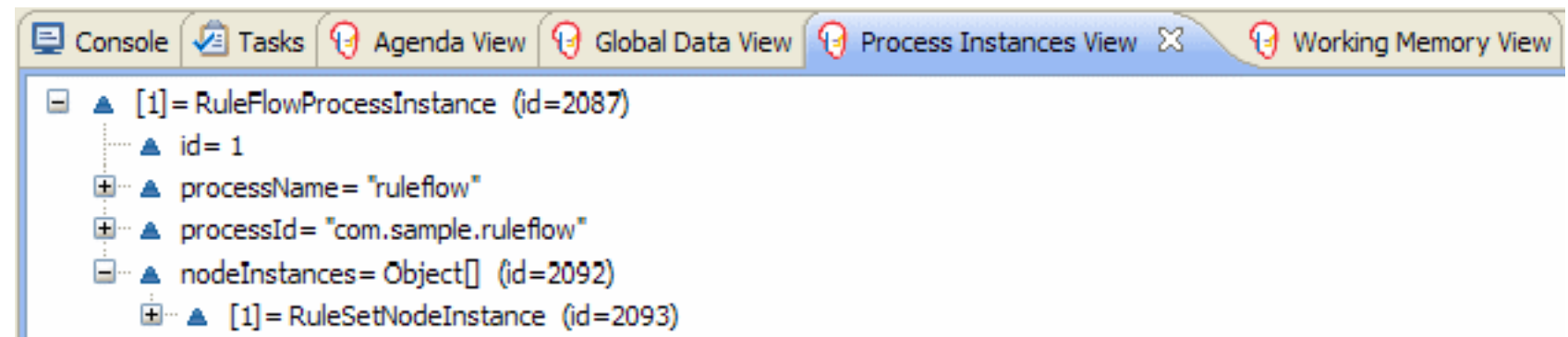
Resume

Skip

Complete

Fail

Integrated Tooling



```
RuleFlowGroup deactivated: validate[size=0]
  RuleFlow node triggered: XOR in process RuleSetExample[org.drools.examples.process.ruleset.RuleSetExample]
    RuleFlow node triggered: in process RuleSetExample[org.drools.examples.process.ruleset.RuleSetExample]
      RuleFlow node triggered: Calculate Discount in process RuleSetExample[org.drools.examples.process.ruleset.RuleSetExample]
        RuleFlowGroup activated: discount[size=1]
      Activation executed: Rule 5% discount if order includes laptop and order after 18h o=Order Order-1(1); date=Wed Jul 02 21:30:30 CEST 2008(1)
    RuleFlowGroup deactivated: discount[size=0]
```

CEP

Complex event processing

Real time events (concurrent events – thread safe):

sliding windows:

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

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

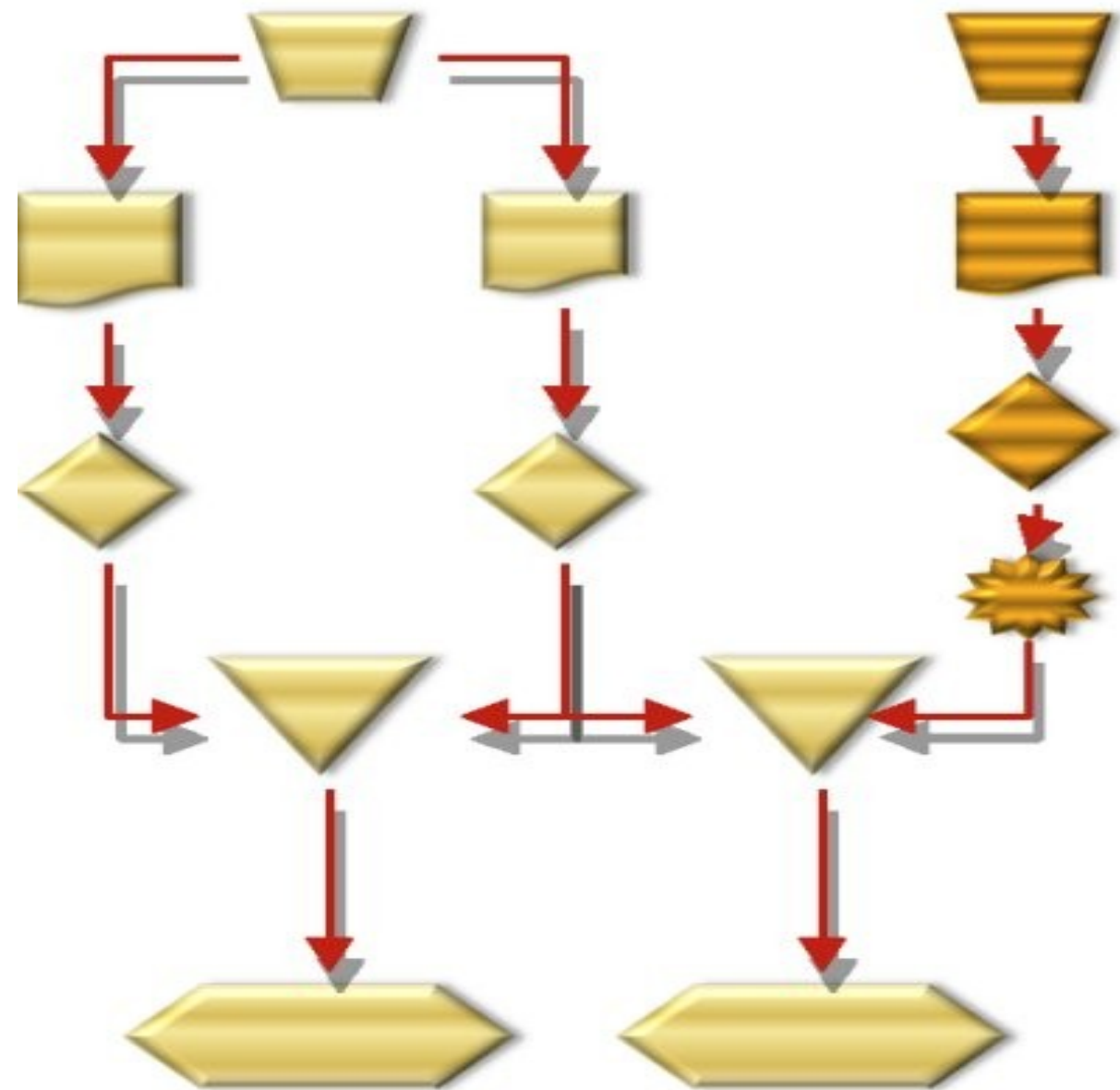
// generating an event class
declare StockTick
  @role( event )

  symbol : String
  price : double
end
```

- (Usually) Immutable Facts
- Strong Temporal Relationship
- Managed Lifecycle
- 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
```

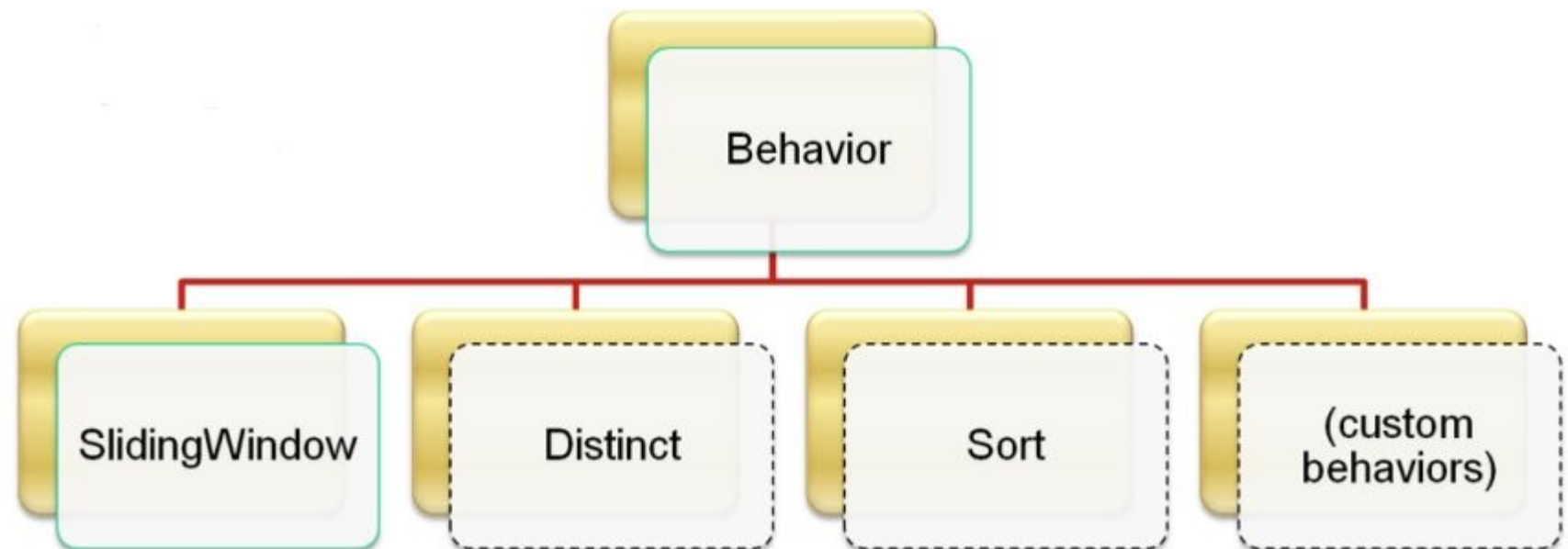
- Event and Time semantics:
 - Point in Time
 - Over an Interval
- Unified semantics for event correlation over time
- Temporal Constraints:
 - Set of 13 operators to express temporal relationship between events

Expressive Event Constraints

	Point-Point	Point-Interval	Interval-Interval
<i>A before B</i>			
<i>A meets B</i>			
<i>A overlaps B</i>			
<i>A includes B</i>			
<i>A finishes B</i>			
<i>A starts B</i>			
<i>A coincides B</i>			

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

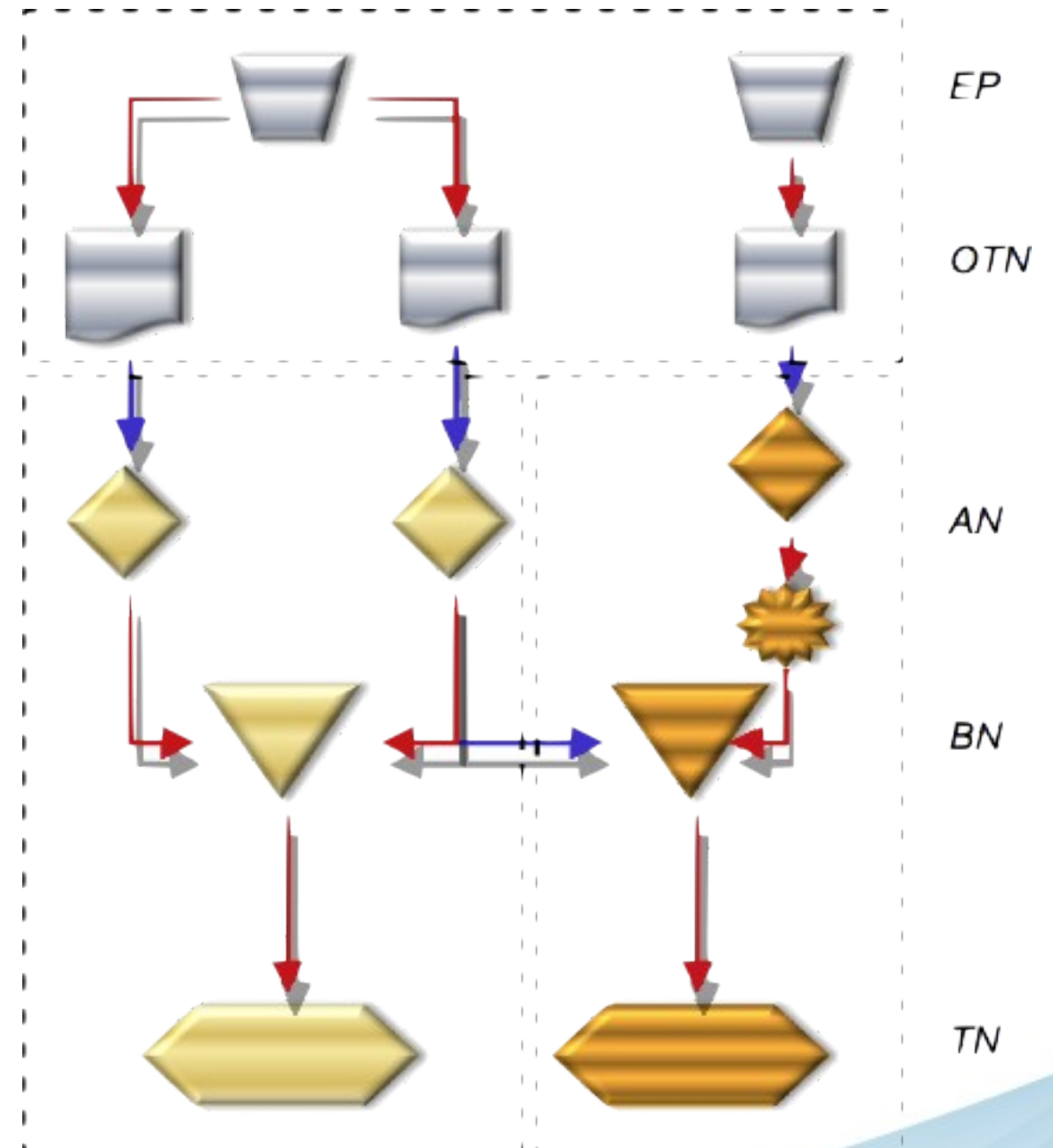


Memory Management

- 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



Planner

Note:
Not Yet
Supported!

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.

$$3 \times 3 = 9$$

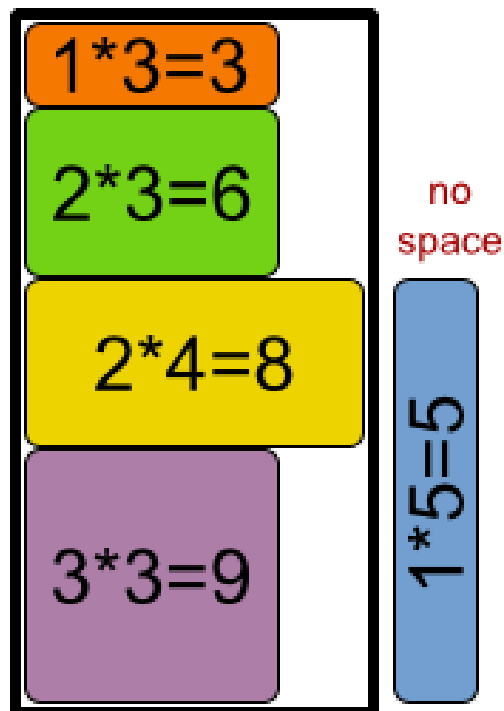
$$2 \times 4 = 8$$

$$2 \times 3 = 6$$

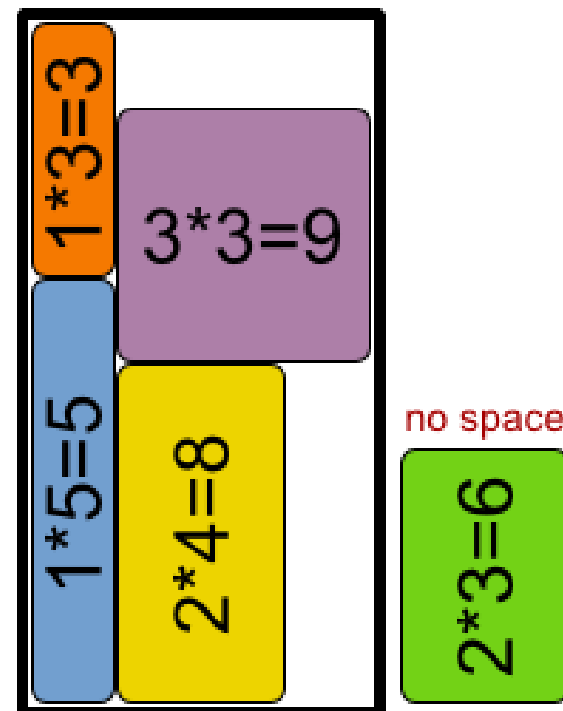
$$1 \times 5 = 5$$

$$1 \times 3 = 3$$

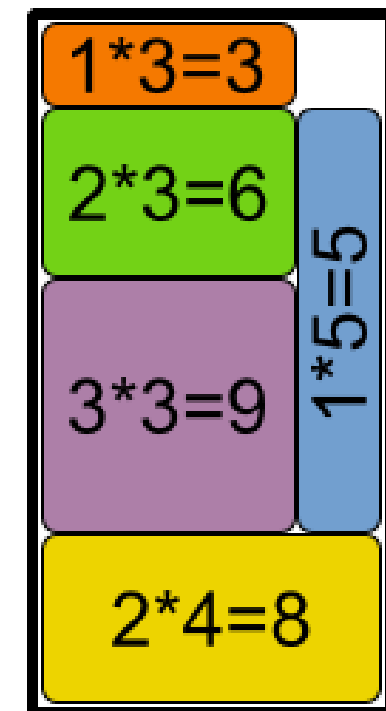
Largest size
first



Largest side
first



Drools
Planner



Employee shift rostering

Populate each work shift with a nurse.

Maternity nurses

A Ann **B** Beth **C** Cory

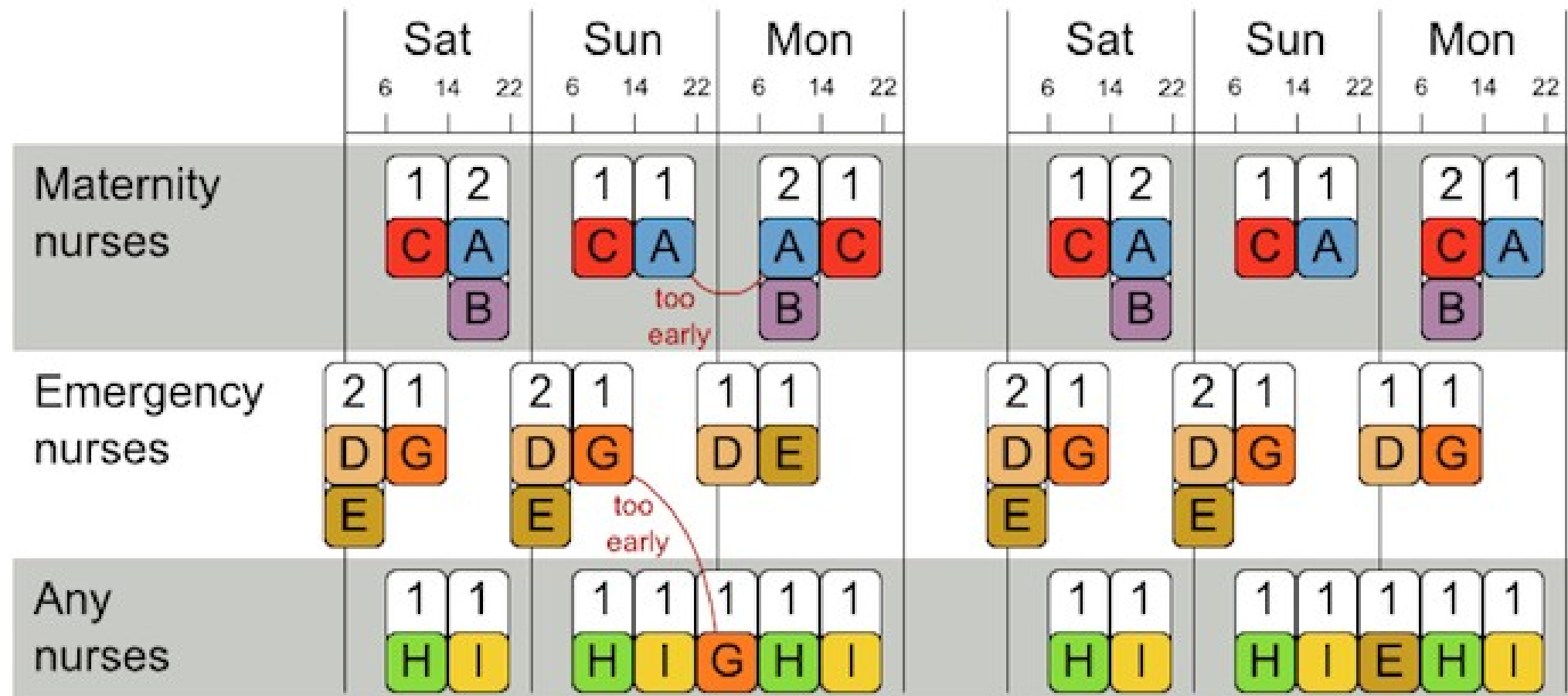
Emergency nurses

D Dan **E** Elin **G** Greg

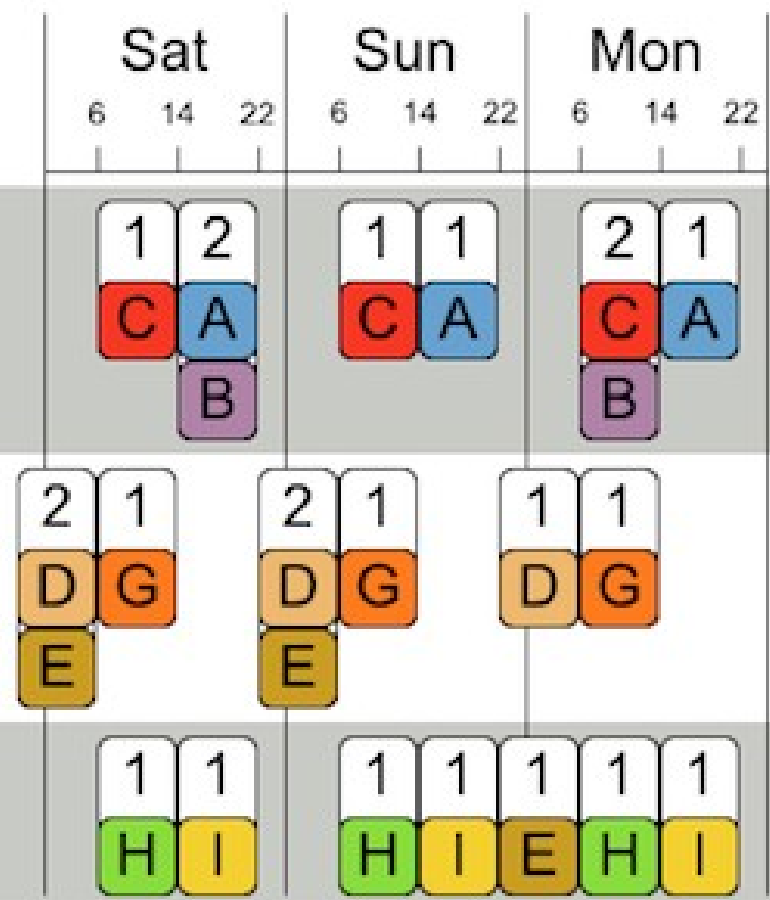
Basic nurses

H Hue **I** Ilse

Largest staff first

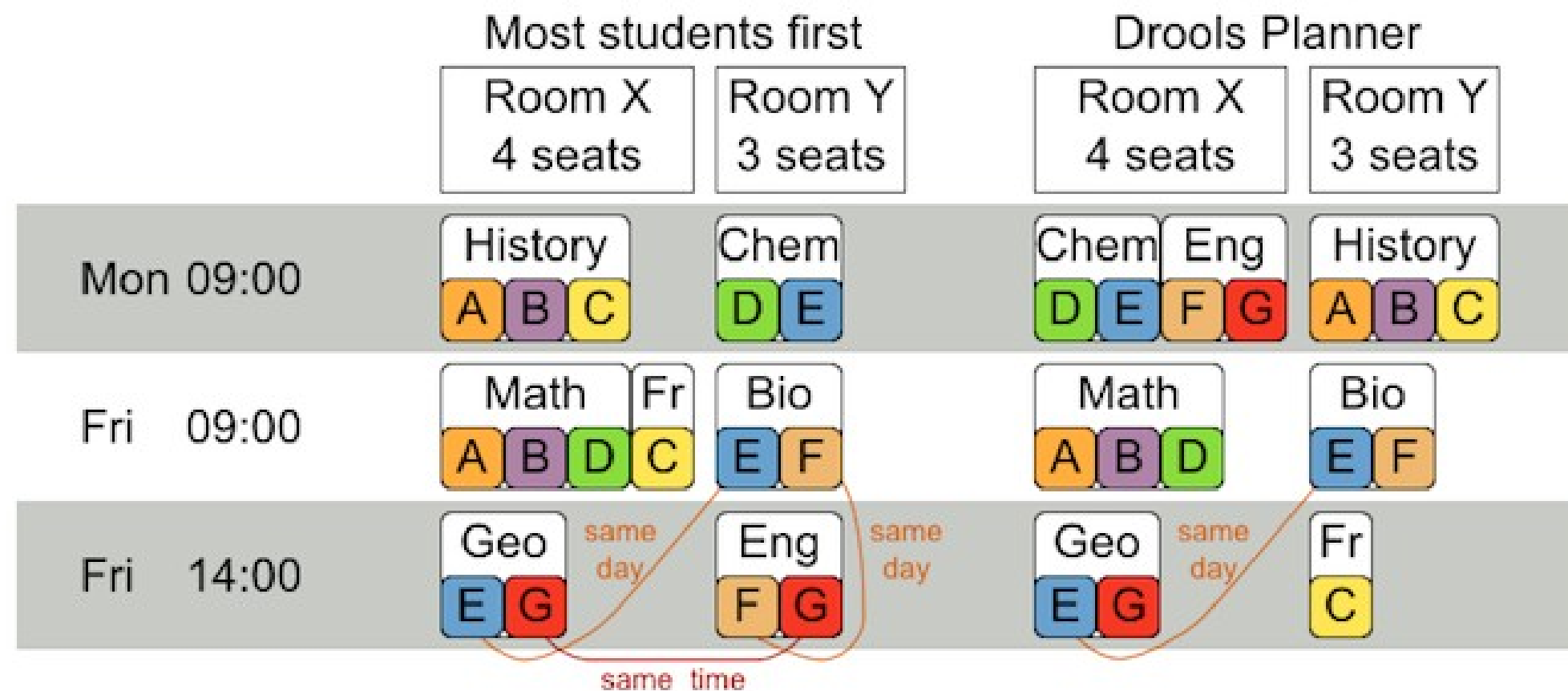
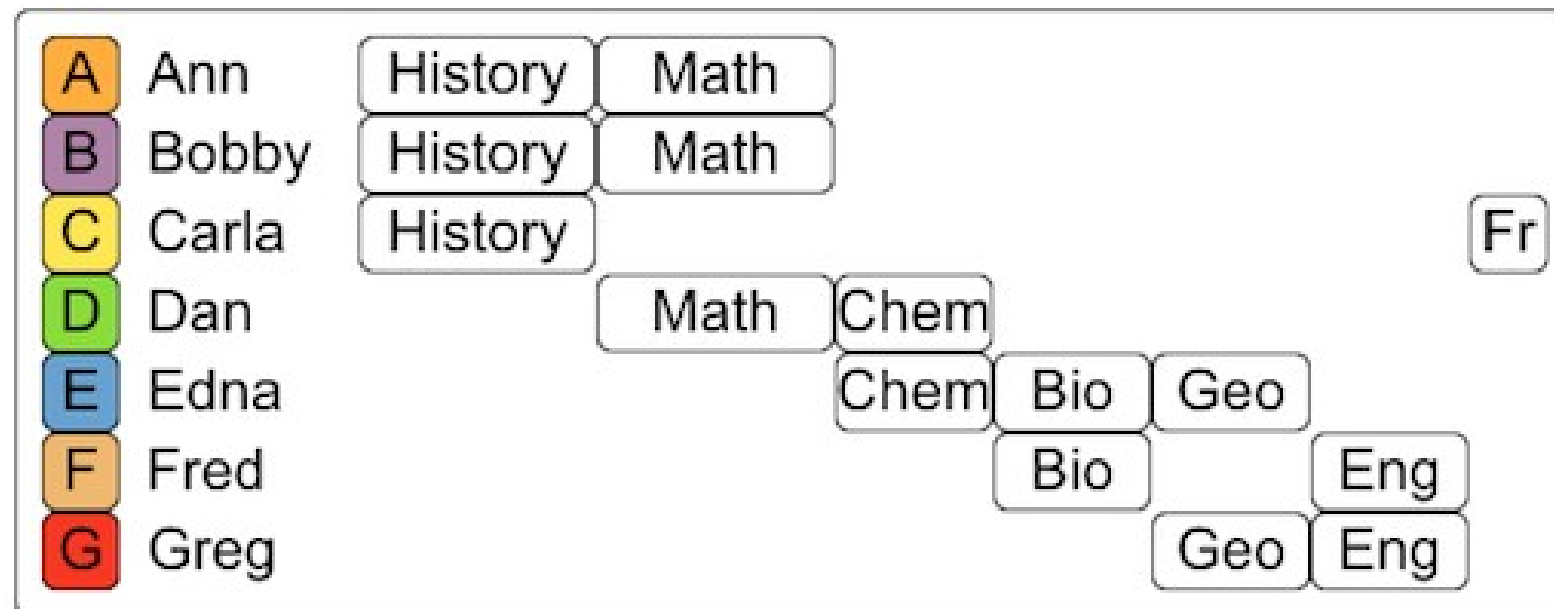


Drools Planner



Examination timetabling

Assign each exam a period and a room.

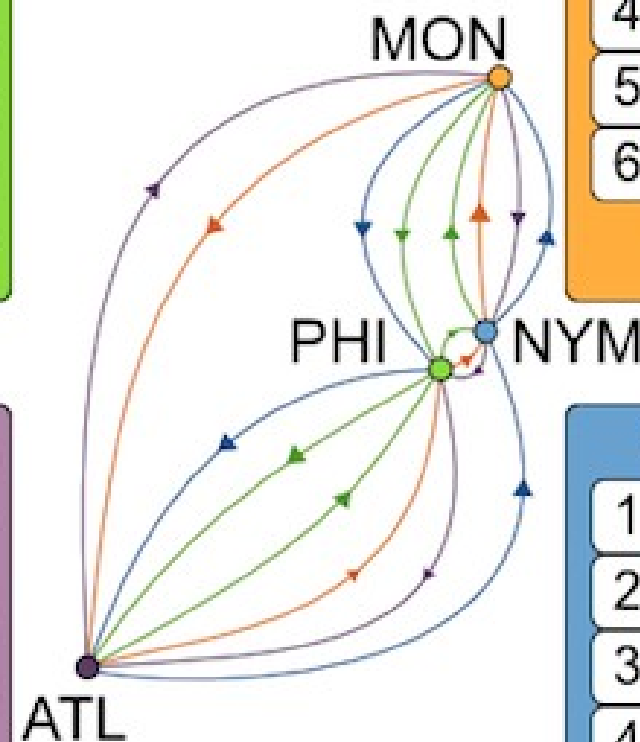


Philadelphia Phillies				80
1	away to	NYM		337
2	away to	MON		380
3	PHI	VS	ATL	0
4	PHI	VS	NYM	0
5	PHI	VS	MON	665
6	away to	ATL		665
Team distance:				2.127

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

Traveling tournament

Schedule each match in a timeslot.



Drools Planner

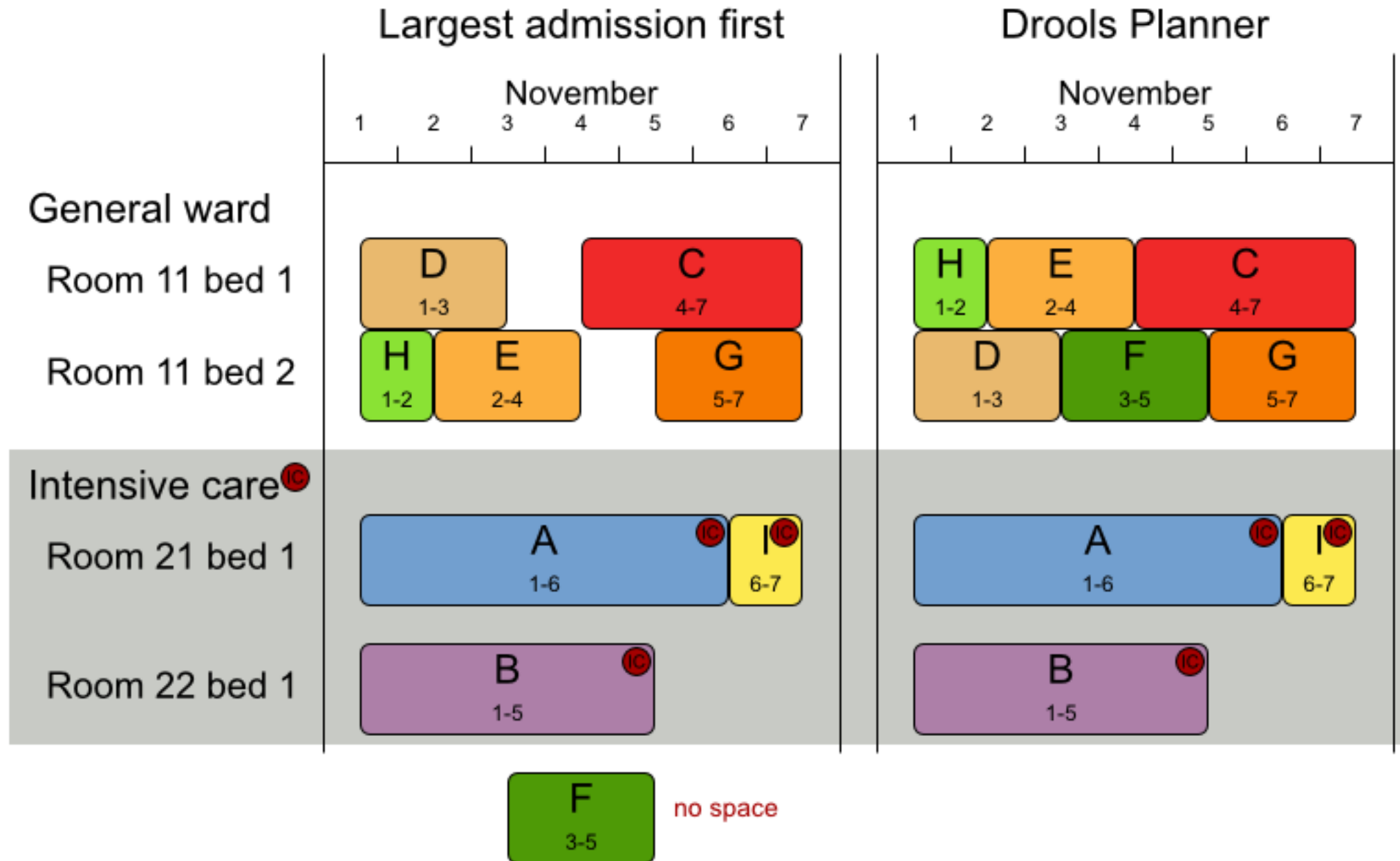
Total distance:
8.276

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

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

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
```

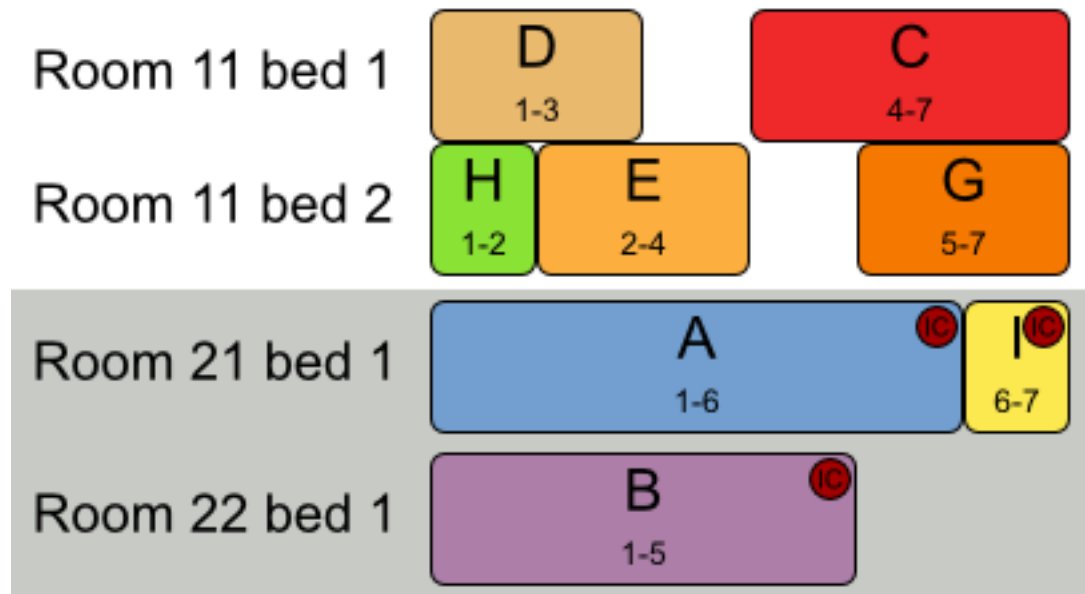
```
    not RoomEquipment(room == $room, equipment == $equipment);
```

```
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 **



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**learn. network.
experience open source.**

**Boston, May 3- 6
2011**

Register with code **RHSRPLO** and receive the
lowest discounted rate of \$895 per pass

REGISTER TODAY: REDHAT.COM/SUMMIT

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/>