



# JBoss Drools

Open-Source  
Business Logic Platform

Kris Verlaenen  
Drools Developer

# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion

# What is a rule ?

*A principle or regulation governing conduct, action, procedure, arrangement, etc.*

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

# What is a rule ?

- Rule conditions = Pattern matching

```
p: Person ( age > 18, $ssn: ssn )  
DriversLicense (  
    type == "Category2",  
    ssn == $ssn )
```

- Rule consequence = Action

```
System.out.println(p.getName());  
insert(new Person());
```

# Drools language features

- Rule conditional elements
  - And, Or, Exists, Not
  - Accumulate, Collect
  - From
  - Forall
  - Temporal rules
- Rule consequence
  - Pluggable dialects (Java, MVEL)
- Functions, globals, queries, etc.

# Drools Rule Formats

- DRL
- Domain-specific language
- Decision tables
- Guided rule editor
- XML

# Drools Rule Language (DRL)

```
import org.drools.Person
global java.util.List myList

rule "RuleName" salience 20
  when
    p: Person( age > 20, name == "John" )
    r: Request( personId == (p.id) )
  then
    myList.add( r );
```

# Domain-specific Language

```
3 expander DSL.dsl
4
5 rule "Your First Rule"
6   when
7     There is a Notification of type "{type}"
8     There is a Person
9     - with age between {x} and {y}
10
11   then <> - with age between {x} and {y}
12         <> - with name "{name}"
13         <> Instance is at least {number} and field is "{value}"
14   end   <> There is a Notification of type "{type}"
15         <> There is a Person
16         <> There is an Instance with field of "{value}"
17         <> There is no current Instance with field : "{value}"
```








# Decision Tables





Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD	Record Reason
Young safe package	18, 24	LOW	1	COMPREHENSIVE	450	
		MED		FIRE_THEFT	200	Priors not relevant
		MED	0	COMPREHENSIVE	300	
		LOW		FIRE_THEFT	150	
		LOW	0	COMPREHENSIVE	150	Safe driver discount
Young risk	18,24	MED	1	COMPREHENSIVE	700	
	18,24	HIGH	0	COMPREHENSIVE	700	Location risk
	18,24	HIGH		FIRE THEFT	550	Location risk
Mature drivers	25,30		0	COMPREHENSIVE	120	Cheapest possible
	25,30		1	COMPREHENSIVE	300	
	25,30		2	COMPREHENSIVE	590	
	25,35		3	THIRD PARTY	800	High risk


# Guided Rule Editor




## Guided rule editor


▼ WHEN 


Person [p]  

 age  is less than  18 

▼ THEN 

Modify [p]   name  

▼ (options) 

salience  

# XML Rule Language

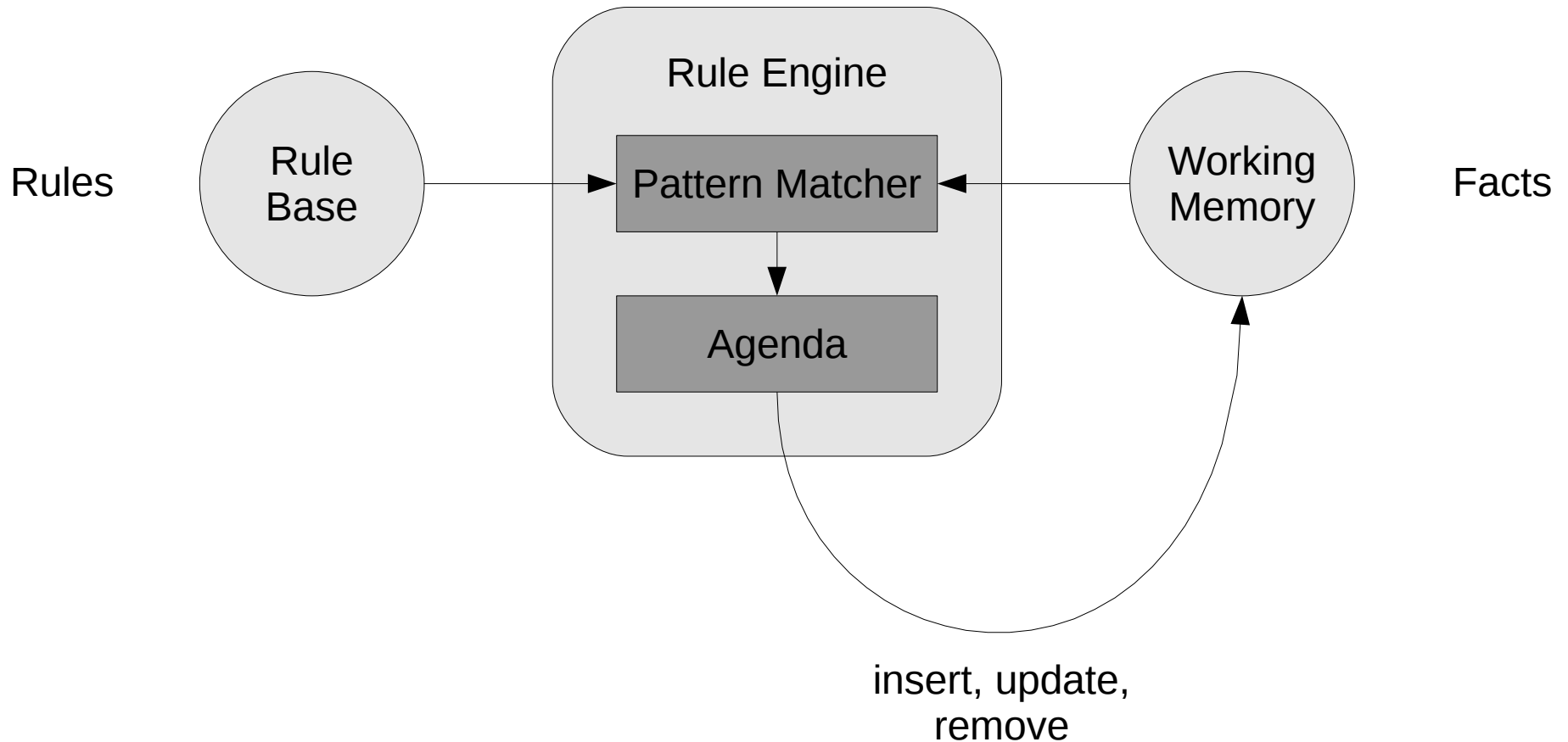
```
<rule name="simple_rule">
  <rule-attribute name="saliency" value="10" />
  <lhs>
    <pattern identifier="$i" object-type="Integer">
      <from>
        <accumulate>
          <pattern object-type="Cheese"></pattern>
          <init> int total = 0; </init>
          <action> total += $cheese.getPrice(); </action>
          <result> new Integer( total ) ); </result>
        </accumulate>
      </from>
    </pattern>
  </lhs>
  <rhs> list1.add( $cheese ); </rhs>
</rule>
```

# Why rules ?

- Separate logic from application
- Understandability
  - Declarative, higher-level
- Speed and scalability
  - ReteOO
- Global enforcement, maintainability and agility
  - Logic is centralized, embrace change
- Traceability

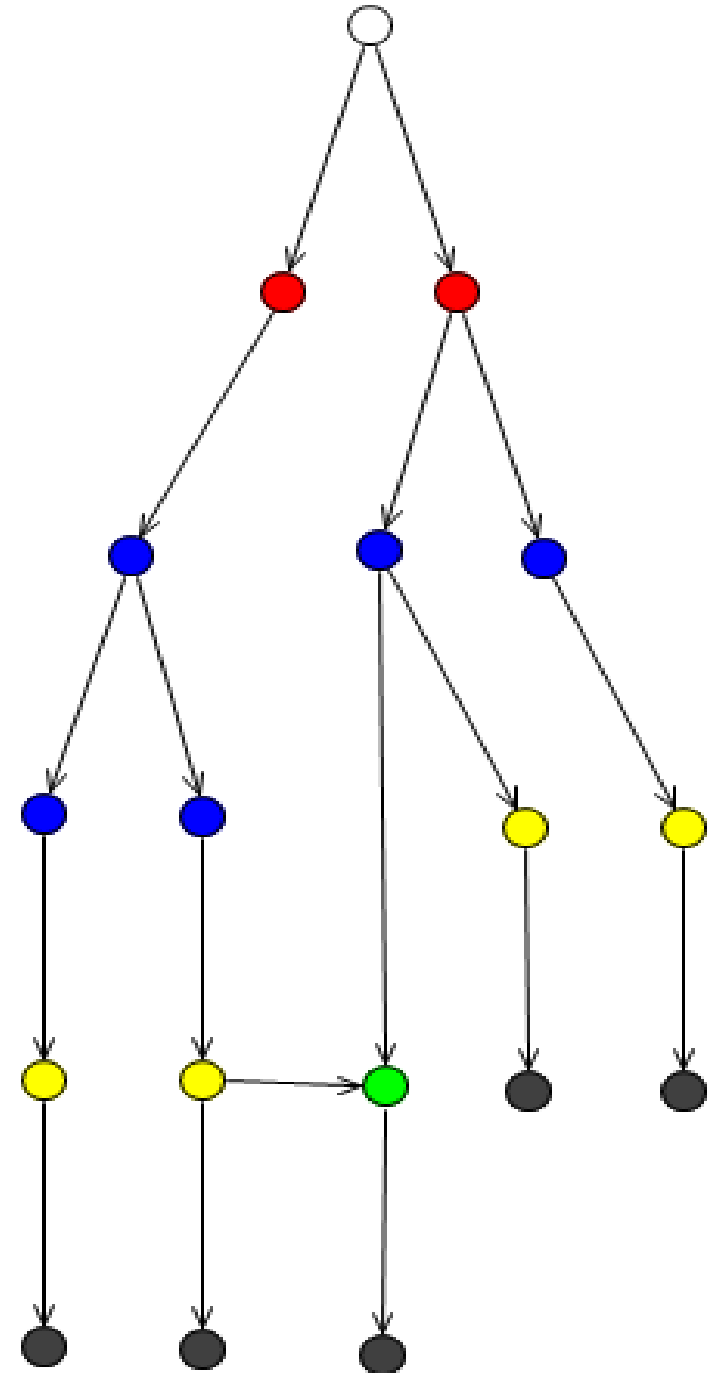
# How does a rule engine work ?

Production rule engine using forward chaining



# ReteOO

- Rule base is compiled into a discrimination network
- Data propagates through network
- Node sharing



# Other Drools Features

- Static rule analysis
  - Detect redundancy, completeness, etc.
- Audit (event listeners)
- JSR-94 compliant (Java Rule Engine API)
- Binary persistence strategy
- Community extensions
  - Uncertainty
  - Planning

# Drools Code example

```
// Create a new rule base
```

```
PackageBuilder builder = new PackageBuilder();
```

```
builder.addPackageFromDrl( ... );
```

```
RuleBase ruleBase = RuleBaseFactory.newRuleBase();
```

```
ruleBase.addPackage( builder.getPackage() );
```

```
// Create working memory
```

```
WorkingMemory workingMemory =  
    ruleBase.newStatefulSession();
```

```
// Insert facts
```

```
Message message = new Message( "Hello World" );
```

```
workingMemory.insert( message );
```

```
// Execute
```

```
workingMemory.fireAllRules();
```



# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion

# Eclipse IDE

- Assist developer in
  - Authoring
  - Executing
  - Testing
  - Debugging
  - Managing
- Extends Eclipse workbench with specific views, editors, wizards, perspective, actions, builder, etc.



Package Explorer

- Hello
  - src/main/java
    - com.sample
      - DroolsTest.java
  - src/main/rules
    - Sample.drl
  - JRE System Library [jdk1.6.0\_02]
  - Drools Library
  - src

```

1 package com.sample
2
3 import com.sample.DroolsTest.Message;
4
5 rule "Hello World"
6   when
7     m : Message( status == Message.HELLO, message : r
8   then
9     System.out.pr
10    m.setMessage(
11    m.setStatus(
12    update( m );
13 end
14
15 rule "GoodBye"
16   no-loop true
17   when
18     m : Message( status == Message.GOODBYE, message :
19   then
20     System.out.println( message );
21     m.setMessage( message );
22
23 end
    
```

- class
- hashCode
- message
- status
- this
- toString

Outline

- com.sample
  - GoodBye
  - Hello World
  - com.sample.DroolsTe

Rules View

- com.sample
  - GoodBye
  - Hello World

Problems Properties Audit View Console

```

Object inserted (1): com.sample.DroolsTest$Message@1a19458
  ↳ Activation created: Rule Hello World message=Hello World(1); m=com.sample.DroolsTest$Message@1a19458(1)
Object inserted (1): com.sample.DroolsTest$Message@1a19458
  ↳ Activation executed: Rule Hello World message=Hello World(1); m=com.sample.DroolsTest$Message@1a19458(1)
Object updated (1): com.sample.DroolsTest$Message@1a19458
  ↳ Activation created: Rule GoodBye message=Goodbye cruel world(1); m=com.sample.DroolsTest$Message@1a19458(1)
Object updated (1): com.sample.DroolsTest$Message@1a19458
  ↳ Activation executed: Rule GoodBye message=Goodbye cruel world(1); m=com.sample.DroolsTest$Message@1a19458(1)
    
```

Debug Console

```

StateExampleUsingSalience [Drools Application]
  org.drools.examples.StateExampleUsingSalience 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
    StateExampleUsingSalience.main(String[]) line: 47
  
```

Variables

Name	Value
b	State (id=1268)
changes	PropertyChangeSupport (id=1297)
name	"B"
state	1

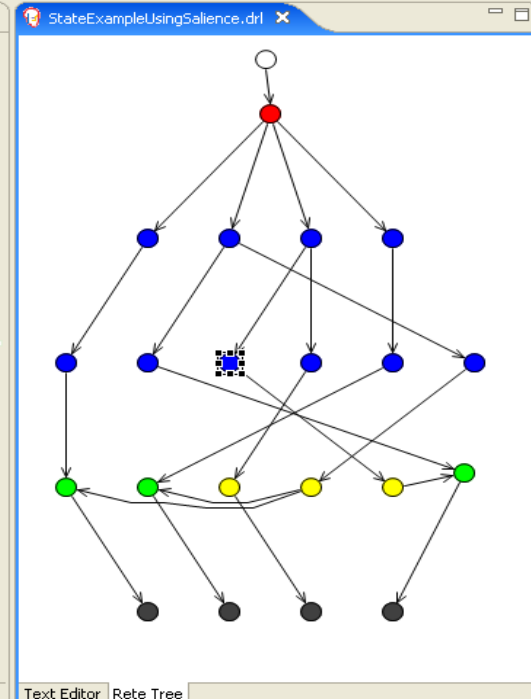
```

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
  
```



Properties

Property	Value
Constraint	[LiteralConstraint fieldExtra
Evaluator	Integer ==
Field Name	state
Name	Alpha BaseVertex
Value	1

Outline

- org.drools.examples
  - A to B
  - B to C
  - B to D
  - Bootstrap
  - org.drools.examples.State

Global Data View

StateExampleUsingSalience [Drools]

A finished

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)

# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion

# Drools Guvnor

*Web-based rule management, storage, editing and deployment environment*

- Rule editing
  - text, guided, decision tables, etc.
- Version control
- Categorization
- Build and deploy
- Scenarios

# Guvnor Rule Editing

JBoss Business Rules Management System - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:8080/drools-jbrms/org.drools.brms.JBRMS/JBRMS.jsp#Rules

## JBoss Rules

Find and edit rules.

Rules Packages Deployment Admin

Explore Rule\_2 MyDT Rule\_1

Status: **[Draft]** Save changes Copy Archive

**IF**

- Person
  - age less than or equal to 42
  - age greater than 21
- Board [b]
- There is no Board
  - cost greater than 1200

**THEN**

- Set [b] cost 1200

(options)

View source Validate

<documentation>

**Rule\_1**

Categories: Finance HR/Awards/QAS

Modified on: Thu 10 May 2007 03:16:47 PM EST  
by: michael  
Note: whoops  
Version: 5  
Created on: Thu 10 May 2007 10:49:41 AM EST  
Created by: michael  
Format: brxml

Package: DemoPackage

Subject:

Type:

External link:

Source:

Version history

Done

# Guvnor Categorization

JBoss Business Rules Management System

http://localhost:8080/drools-jbrms/org.drools.brms.JBRMS/JBRMS.jsp

User: fmeyer [Sign Out]

## JBoss Rules

Administer the repository

Manage categories Manage status

Create a new top level category.

Category name: Fibonacci

Description: fibonacci category

OK Cancel

Categories aid in managing large numbers of rules/assets. A shallow hierarchy is recommended.

**Current categories:**

- + HR
- + Finance
- + Draft

Refresh view:

Create a new category:

Delete the currently selected category:



# Guvnor Deployment

The screenshot displays the JBoss Business Rules Management System (Guvnor) web interface. The browser window title is "JBoss Business Rules Management System". The address bar shows "http://localhost:8080". The user is logged in as "User: fmeyer" with a "Sign Out" link.

A notification dialog box is open, stating: "The page at http://localhost:8080 says: The snapshot called: NewSnapshot was successfully created." with an "OK" button.

The main interface shows a sidebar with navigation options: Info, Rules, Packages, Deployment, and Admin. The "Packages" section is expanded, showing a tree structure with "com.sample" selected. Under "com.sample", there are options for Business rules, Technical rules, Functions, DSL, and Model. A "default" package is also visible.

The main content area displays "Info" for the selected package, with options to "Edit Package configuration" and "Build, validate and deploy". A "Create a snapshot for deployment" dialog box is open, featuring a clock icon and the following text: "A package snapshot is essentially a read only 'locked in' and labelled view of a package at a point in time, which can be used for deployment." The dialog includes a "Choose or create snapshot name:" field with a radio button selected for "NEW:" and the value "NewSnapshot". A "Comment:" field contains the text "commenting". A "Create new snapshot" button is visible at the bottom of the dialog.

The status bar at the bottom left of the browser window shows "Done".

# Guvnor Test Scenarios

**Drools**

Rules Packages Deployment Admin **QA**


QA

Test Scenarios in packages:


- com.billasurf.finance
- com.billasurf.hrman
- com.billasurf.manufacturing
- com.billasurf.manufacturing.plant**
- com.billasurf.sales
- defaultPackage


Analysis

Find Scenarios for com.billasurf.manufacturing.plant



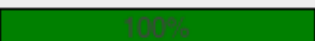
 **Scenarios for package: com.billasurf.manufacturing.plant**  
Run all scenarios

Overall result: **FAILURE**

Results:  66% 1 failures out of 3 expectations.

Rules covered:  100% 100% of the rules were tested.

**Scenarios**

test1:	 100%	[0 failures out of 1]	<input type="button" value="Open"/>
test2:	 66%	[1 failures out of 1]	<input type="button" value="Open"/>
test3:	 100%	[0 failures out of 1]	<input type="button" value="Open"/>

# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion

# Drools Flow

*A workflow engine combining processes and rules*

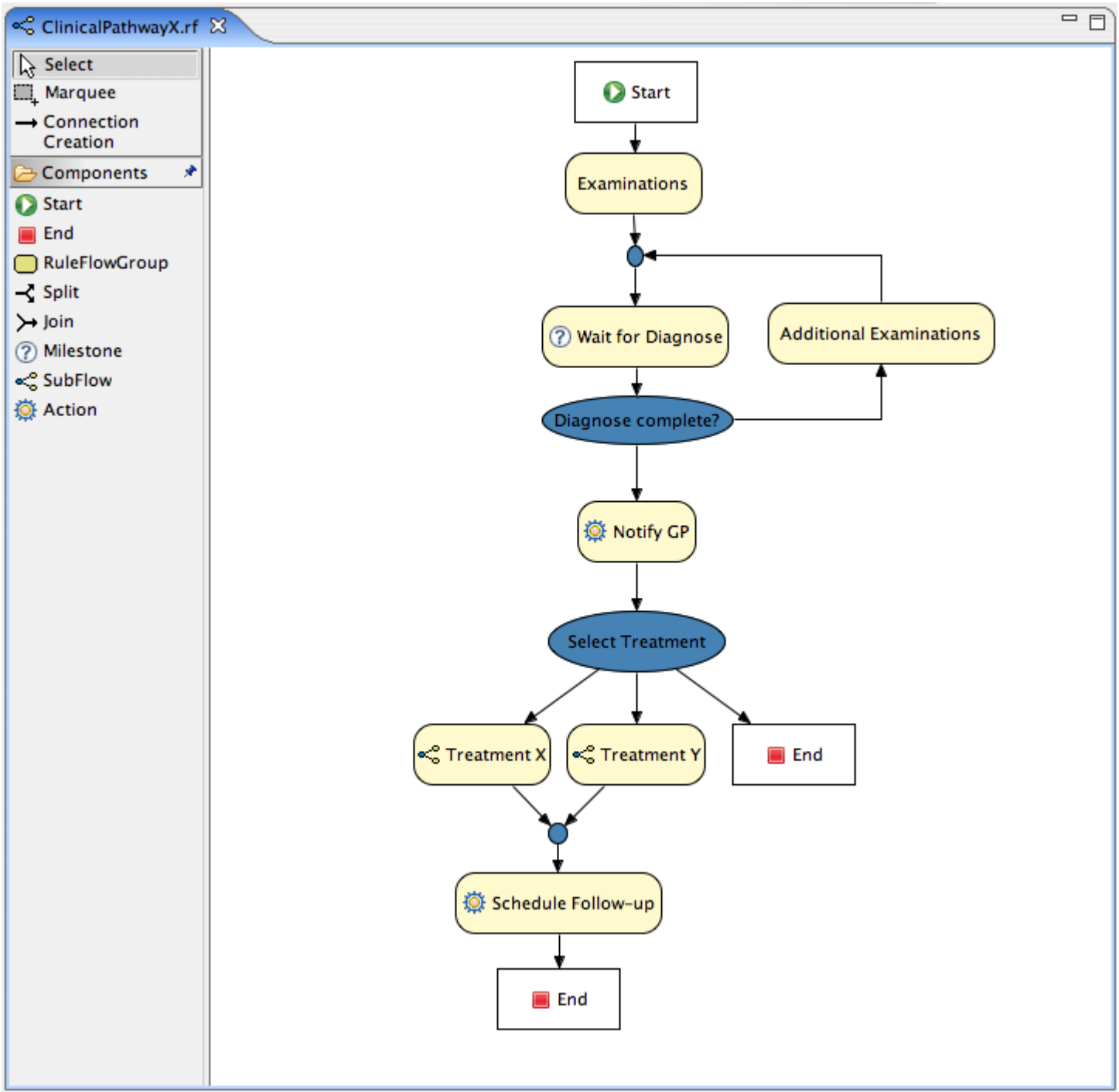
- Integration
  - From loose coupling (decision services)
  - To advance integration (process rules)
- Unification
  - Rules and processes are different types of business knowledge assets
  - Tooling (IDE, repository, management, etc.)

# Why use rules in processes

- **Simplicity**
  - Complex decisions are sometimes easier to specify using rules
- **Separate life cycle and different scope => Agility**
  - Change rules without having to change process, rules can be reused accross processes or outside processes
- **Declarative and higher-level**
  - Focus on what instead of how, rules are additive
- **Granularity**
  - Easy to write rules for specific circumstances, processes should be more generic
- **Performance**

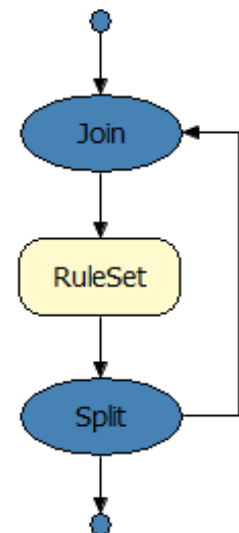
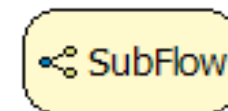
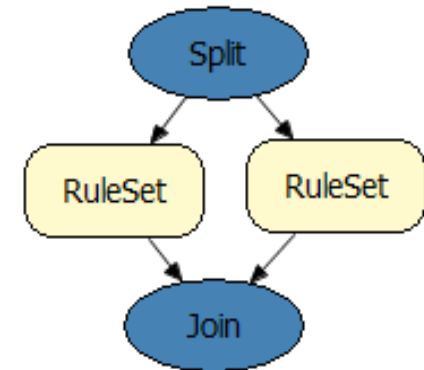
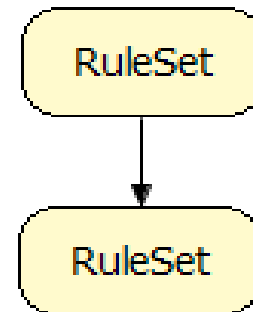
# Drools Flow

- Unifies rules and processes in a single engine
  - Ability to use rules everywhere in your process
    - Decision nodes, constraints, exception and event handling, task assignment, etc.
  - Processes and rules see, reason and react on the same data
    - No data passing or synchronization
  - Processes and rules interact
  - Integrated tooling



# Drools Flow

- Control flow
  - Sequence, Paralellism
  - Choice, Looping
- Data flow (variables)
- Nodes
  - Action, Milestone (State), Subflow, Timer, Composite, Human Tasks
  - Exceptions, Timers





# Additional Features

- Extensible process framework
  - Reusable set of core nodes
- Binary persistence of process instances
- Domain-specific work items
  - Plug in your own work nodes
  - Simplifies integration and testing
- Integrated debug and audit

# Domain-specific Processes

The image displays a BPMN editor interface with a process flow diagram and a 'New Application' dialog for an email action.

**Process Flow Diagram:**

```
graph LR; Start[Start] --> Split((Split)); Split --> Email[Email]; Split --> Log1[Log]; Email --> Join((Join)); Log1 --> Join; Join --> Log2[Log]; Log2 --> End[End];
```

**New Application Dialog (Email Action):**

Recipients:

Type	Display Name	E-Mail
to	Tom Schindl	tom.schindl@bestsolution.at
bcc	Boris Bokowski	boris_bokowski@ca.ibm.com
cc	Tod Creasey	tod_creasey@ca.ibm.com

add delete

from:

reply to:

Subject:

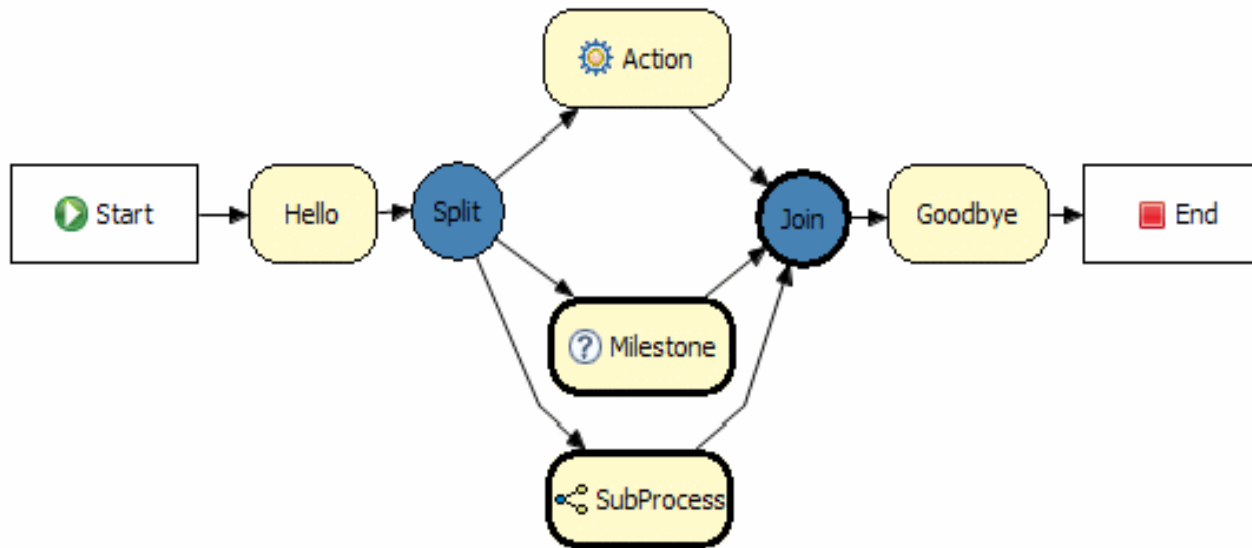
type:  text  html

Body:

**Properties Panel:**

Property	Value
From	me@mail.com
Name	Email
Subject	Email
Text	Hello !
To	you@mail.com

# Integrated debug and audit



Object inserted (1): com.sample.RuleFlowTest\$Message@17200b4  
⇒ Activation created: Rule Hello World m=com.sample.RuleFlowTest\$Message@17200b4(1); message=Hello World(1)

RuleFlow started: ruleflow[com.sample.ruleflow]

- RuleFlow node triggered: Start in process ruleflow[com.sample.ruleflow]
- RuleFlow node triggered: Hello in process ruleflow[com.sample.ruleflow]
- RuleFlowGroup activated: hello[size=1]

Activation executed: Rule Hello World m=com.sample.RuleFlowTest\$Message@17200b4(1); message=Hello World(1)

- Object updated (1): com.sample.RuleFlowTest\$Message@17200b4
- ⇒ Activation created: Rule GoodBye m=com.sample.RuleFlowTest\$Message@17200b4(1); message=Goodbye cruel world(1)

RuleFlowGroup deactivated: hello[size=0]

- RuleFlow node triggered: Goodbye in process ruleflow[com.sample.ruleflow]
- RuleFlowGroup activated: goodbye[size=1]

Activation executed: Rule GoodBye m=com.sample.RuleFlowTest\$Message@17200b4(1); message=Goodbye cruel world(1)

RuleFlowGroup deactivated: goodbye[size=0]

- RuleFlow node triggered: End in process ruleflow[com.sample.ruleflow]
- RuleFlow completed: ruleflow[com.sample.ruleflow]

# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion

# Drools Fusion

*Temporal reasoning system to provide Complex Event Processing (CEP) capabilities*

- Processing multiple events from an event cloud with the goal of identifying the meaningful events within the event cloud.
  - Event detection
  - Event correlation
  - Event abstraction

# Requirements for event processing

- Event Detection
  - From an event cloud, select all the meaningful events and only them.
- [Temporal] Event Correlation
  - Ability to correlate events and facts declaring temporal and non-temporal constraints between them.
  - Ability to reason over event aggregation
- Event Abstraction
  - Ability to declare composite complex events from simple atomic events

# Requirements for event processing

- Event Detection
  - Leverages all the power and expressiveness of the rule language
  - Extension for stream support: entry-points
- [Temporal] Event Correlation
  - Temporal extension to Rete Algorithm: before, after, meets, met-by, overlaps, overlapped-by, during, contains, starts, started-by, finishes, finished-by, concurrent
- Event Abstraction
  - Sliding window support

# Overview

- Drools Expert
  - What is a rule?
  - Why rules?
  - How does a rule engine work?
- Drools Eclipse IDE
- Drools Guvnor
- Drools Flow
- Drools Fusion



# Questions ?

Drools Homepage

<http://www.jboss.org/drools/>

Drools Blog

<http://blog.athico.com/>

Drools Chat

[#drools](irc://codehaus.org)

Drools Mailing List

[rules-users@lists.jboss.org](mailto:rules-users@lists.jboss.org)

