JBoss Drools - Viva Le Drools

Declarative Behavioural Modelling

An Integrated AI approach

Mark Proctor

Project Lead

- The SkyNet funding bill is passed.
- The system goes online on August 4th, 1997.
- Human decisions are removed from strategic defense.
- SkyNet begins to learn at a geometric rate.
- It becomes self-aware at 2:14am Eastern time, August 29th
- In a panic, they try to pull the plug.
- And, Skynet fights back

Drools http://labs.jboss.com/drools

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://labs.jboss.com/drools/

jboss.org community driven.

Subscribe | Register | mark.proctor@jboss.com | Logout | Search...

Home | Resources | Projects | Community

Drools

Drools

- JIRA issue tracker
- Documentation
- Wiki Knowledge Base
- Downloads
- Blog
- Live Trails
- Features
- Screenshots
- Mailing Lists
- Realtime Chat (IRC)
- Subversion
- Drools.Net
- The Team
- Purchase support

Drools is a business rule management system (BRMS) and an enhanced Rules Engine implementation, ReteOO, based on Charles Forgy's Rete algorithm tailored for the Java language. More importantly, Drools provides for Declarative Programming and is flexible enough to match the semantics of your problem domain with Domain Specific Languages, graphical editing tools, web based tools and developer productivity tools.

- Product Information (training and support)
- Drools Blog
- Drools Job Board

Why Choose Drools

Open Source	Drools has a business friendly open source license that makes it free to download, use, embed, and distribute.
Declarative Programming	Allow you to say "What to do" not "How to do it".
Accessible business rules	Drools is friendly to both developers and business users. DSLs allow developers to write almost natural language semantics for rule authors. GUIs and visual metaphors (RuleFlow, Decision tables in Spreadsheets) also reduce the gap between business and IT. A web based BRMS (Business Rule Management System) provides GUIs for managing rule assets.
Logic and Data Separation	Rule engines are the ultimate in logic and data de-coupling. Increases

Transferring data from labs.jboss.com...

http://blog.athico.com

Drools - Because Knowledge Matters - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://blog.athico.com/

SEARCH BLOG | FLAG BLOG | Next Blog» | Create Blog | Sign In

Drools

Your hosts: Mark Proctor, Michael Neale, Edson Tirelli and Fernando Meyer.

Tuesday, November 06, 2007

Drools now has 1725 unit and integration tests

Posted by Mark Proctor

One of the great things about Open Source is we are totally open and transparent, so it's very easy to make a judgement on the level of quality of the software and the efforts gone into QA. On this note we would like to bring to everyone's attention that Drools now has 1725 unit and integration tests - which I think is high by anyone's standard - none of these tests were produced by code generation. This report is shown as part our Hudson built test results page, https://hudson.jboss.org/hudson/job/drools/983/testReport/.

Our Hudson build server, https://hudson.jboss.org/hudson/job/drools/, builds Drools after every commit and makes distribution zips publicly available here , so you can always get the latest trunk build for your own testing.

Technorati Links • Subscribe to this feed • Save to del.icio.us • Digg This! • Share on Facebook • Stumble It! • Post to dzone

at 4:46 PM 0 comments Links to this post

Subscribe Now

RSS Feed
Atom Feed

Drools Job Board

See more jobs
Post your job here

Powered by JobThread
Get your own job board

Blog Archive

- ▼ 2007 (113)
  - ▼ November (3)
    - Drools now has 1725 unit and integration tests
    - Irish Java Technology Conference - 9th of November...
    - More great articles
  - ▶ October (14)
  - ▶ September (10)
  - ▶ August (15)
  - ▶ July (18)

Done

Classes

Account

long accountNo
double balance

Cashflow

Date date
double amount
int type
long accountNo

AccountingPeriod

Date start
Date end

Creating Views with Triggers

date	amount	type	accountNo
12-Jan-07	100	CREDIT	1
2-Feb-07	200	DEBIT	1
18-May-07	50	CREDIT	1
9-Mar-07	75	CREDIT	1

AccountingPeriod	
start	end
01-Jan-07	31-Mar-07

Account	
accountNo	balance
1	0

increase balance for AccountPeriod Credits

```
select * from  Account acc,
     Cashflow cf, AccountPeriod ap
where acc.accountNo ==  cf.accountNo and
      cf.type == CREDIT
      cf.date >= ap.start and
      cf.date <= ap.end

 trigger : acc.balance += cf.amount
```

CashFlow		
date	amount	type
12-Jan-07	100	CREDIT
18-May-07	50	CREDIT

Account	
accountNo	balance
1	-50

decrease balance for AccountPeriod Debits

```
select * from  Account acc,
     Cashflow cf, AccountPeriod ap
where acc.accountNo ==  cf.accountNo and
      cf.type == DEBIT
      cf.date >= ap.start and
      cf.date <= ap.end

 trigger : acc.balance -= cf.amount
```

CashFlow		
date	amount	type
2-Feb-07	200	DEBIT

Creating Views with Triggers

date	amount	type	accountNo
12-Jan-07	100	CREDIT	1
2-Feb-07	200	DEBIT	1
18-May-07	50	CREDIT	1
9-Mar-07	75	CREDIT	1

AccountingPeriod	
start	end
01-Apr-07	30-Jun-07

Account	
accountNo	balance
1	0

increase balance for AccountPeriod Credits

```
select * from  Account acc,
     Cashflow cf, AccountPeriod ap
where acc.accountNo ==  cf.accountNo and
      cf.type == CREDIT
      cf.date >= ap.start and
      cf.date <= ap.end

 trigger : acc.balance += cf.amount
```

CashFlow		
date	amount	type
2-Feb-07	200	CREDIT

Account	
accountNo	balance
1	150

decrease balance for AccountPeriod Debits

```
select * from  Account acc,
     Cashflow cf, AccountPeriod ap
where acc.accountNo ==  cf.accountNo and
      cf.type == DEBIT
      cf.date >= ap.start and
      cf.date <= ap.end

 trigger : acc.balance -= cf.amount
```

CashFlow		
date	amount	type

What is a Rule

Quotes on Rule names are optional if the rule name has no spaces.

```
"  rule    <name>
       <attribute> <value>
       when
           <LHS>
       then
           <RHS>
   end
```

salience	<int>
agenda-group	<string>
no-loop	<boolean>
auto-focus	<boolean>
duration	<long>

RHS can be any valid java. Future versions will support other languages, i.e Groovy

What is a Rule

Methods that must be called directly

specific passing of instances

```
"   public void helloMark(Person person) {
        if ( person.getName().equals(   mark   ) {
            System.out.println(   Hello Mark   );
        }
```

Rules can never be called directly

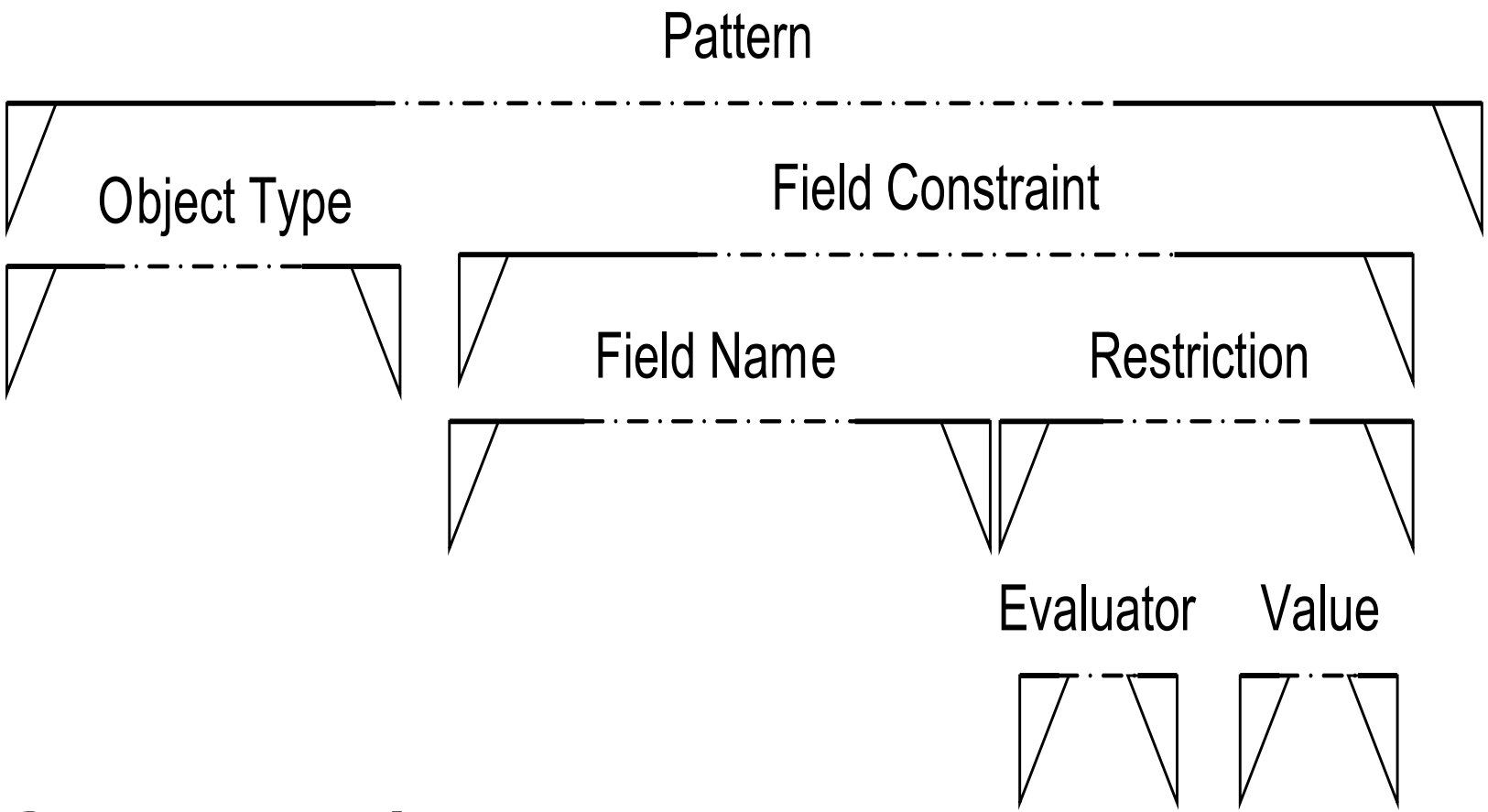
Specific instances cannot be passed.

```
"   rule   Hello Mark
        when
            Person( name                mark   )
        then
            System.out.println(   Hello Mark   );
    end
```

LHS

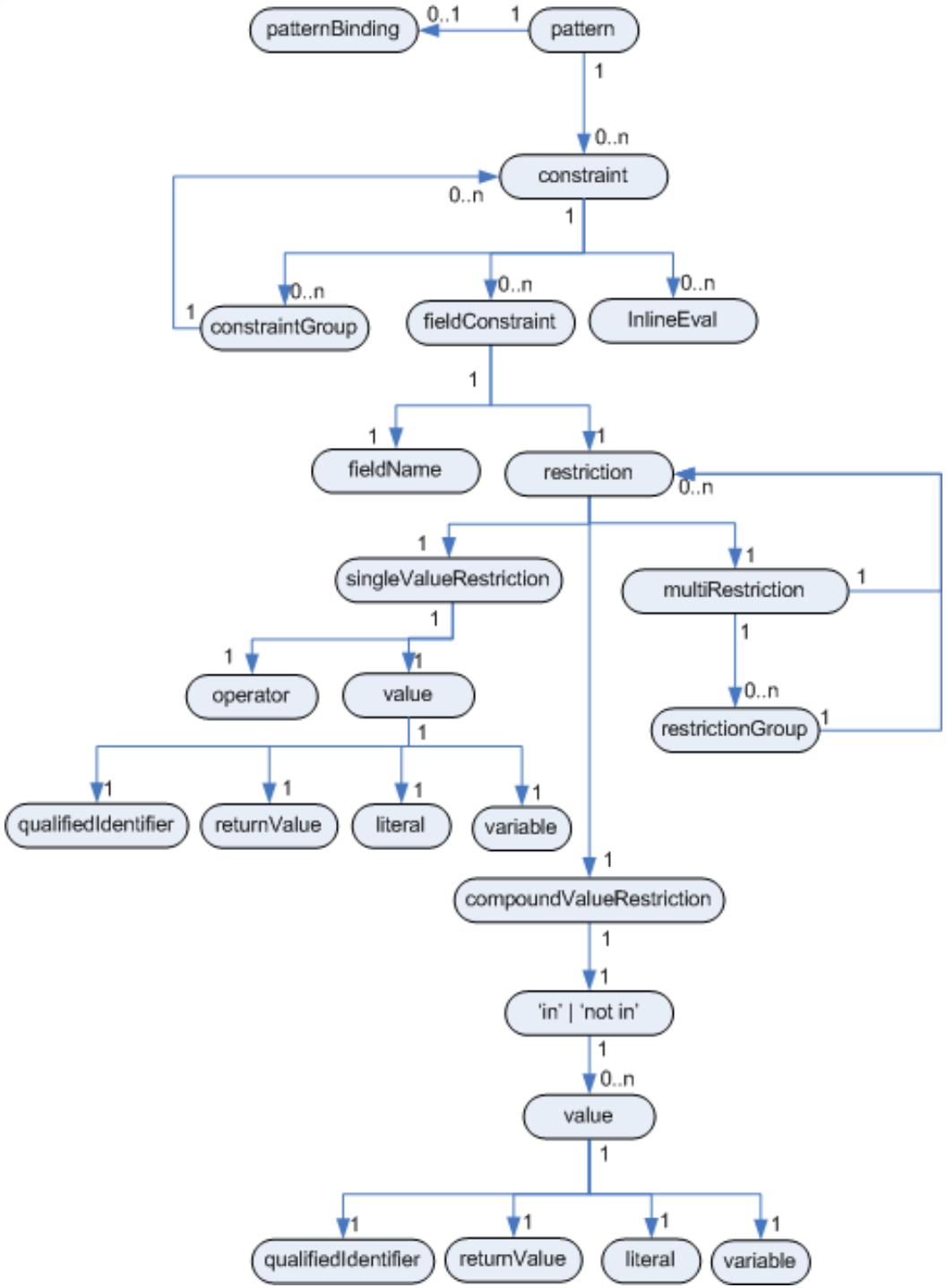
RHS

What is a Pattern



Shower(temperature == “hot”)

Anatomy of a Pattern



Our First Rule

```
select * from  Account acc,
     Cashflow cf, AccountPeriod ap
where acc.accountNo ==  cf.accountNo and
      cf.type == CREDIT
      cf.date >= ap.start and
      cf.date <= ap.end
```

Pattern

Pattern Binding

field Binding

```
rule "increase balance for AccountPeriod Credits"
     when
          ap : AccountPeriod()
          acc : Account( $accountNo : accountNo )
          CashFlow( type == CREDIT,
                    accountNo == $accountNo,
                    date >= ap.start && <= ap.end,
                    $ammount : ammount
     then
          acc.balance  += $amount;
end
```

Variable Restriction

Literal Restriction

Multri Restriction - Variable Restriction

field Binding

Consequence (RHS)

Rules as a “view”

date	amount	type	accountNo
12-Jan-07	100	CREDIT	1
2-Feb-07	200	DEBIT	1
18-May-07	50	CREDIT	1
9-Mar-07	75	CREDIT	1

AccountingPeriod	
start	end
01-Jan-07	31-Mar-07

Account	
accountNo	balance
1	0

```
rule “increase balance for AccountPeriod
      Credits”
  when
    ap : AccountPeriod()
    acc : Account( $accountNo : accountNo )

    CashFlow( type == CREDIT,
              accountNo == $accountNo,
              date >= ap.start && <= ap.end,
              $ammount : ammount )
  then
    acc.balance  += $amount;
end
```

```
rule “decrease balance for AccountPeriod
      Debits”
  when
    ap : AccountPeriod()
    acc : Account( $accountNo : accountNo )

    CashFlow( type == DEBIT,
              accountNo == $accountNo,
              date >= ap.start && <= ap.end,
              $ammount : ammount )
  then
    acc.balance  -= $amount;
end
```

CashFlow		
date	amount	type
12-Jan-07	100	CREDIT
18-May-07	50	CREDIT

CashFlow		
date	amount	type
2-Feb-07	200	DEBIT

Account	
accountNo	balance
1	-50

Rules as a “view”

date	amount	type	accountNo
12-Jan-07	100	CREDIT	1
2-Feb-07	200	DEBIT	1
18-May-07	50	CREDIT	1
9-Mar-07	75	CREDIT	1

AccountingPeriod	
start	end
01-Apr-07	30-Jun-07

Account	
accountNo	balance
1	0

```
rule “increase balance for AccountPeriod
      Credits”
  when
    ap : AccountPeriod()
    acc : Account( $accountNo : accountNo )

    CashFlow( type == CREDIT,
              accountNo == $accountNo,
              date >= ap.start && <= ap.end,
              $ammount : ammount )
  then
    acc.balance  += $amount;
end
```

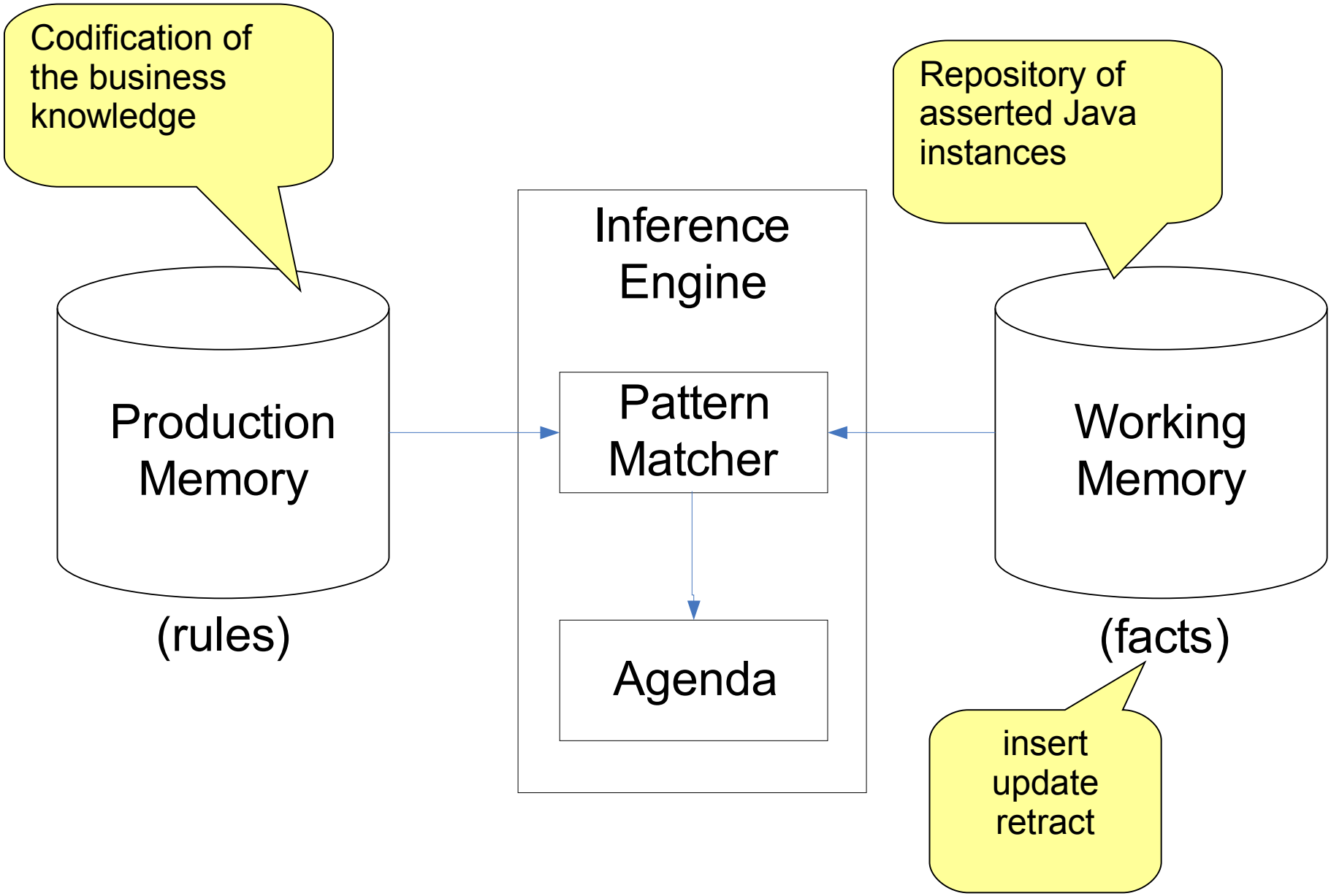
```
rule “decrease balance for AccountPeriod
      Debits”
  when
    ap : AccountPeriod()
    acc : Account( $accountNo : accountNo )

    CashFlow( type == DEBIT,
              accountNo == $accountNo,
              date >= ap.start && <= ap.end,
              $ammount : ammount )
  then
    acc.balance  -= $amount;
end
```

CashFlow		
date	amount	type
2-Feb-07	200	CREDIT

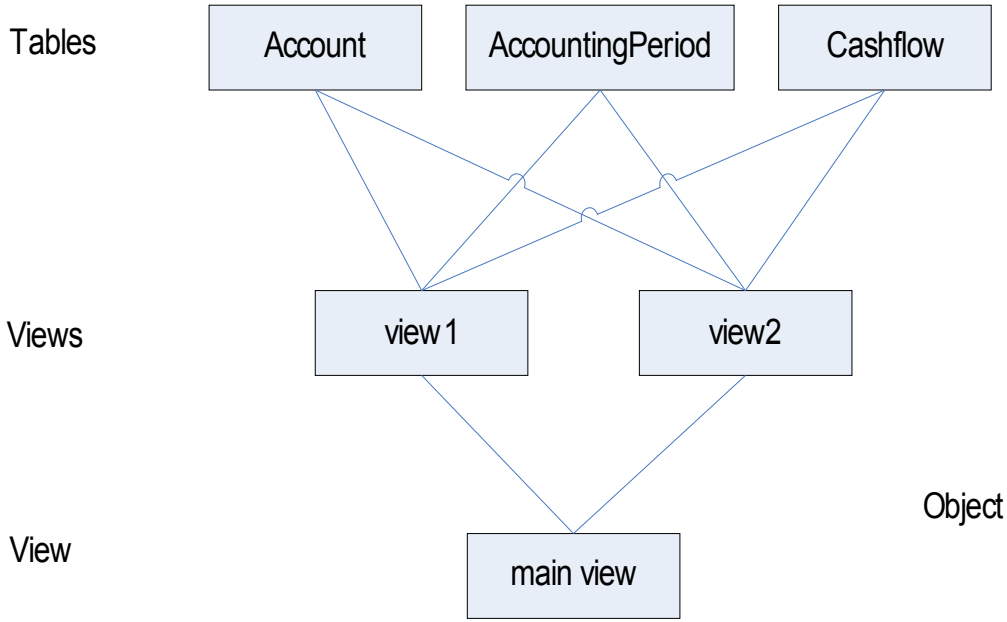
CashFlow		
date	amount	type

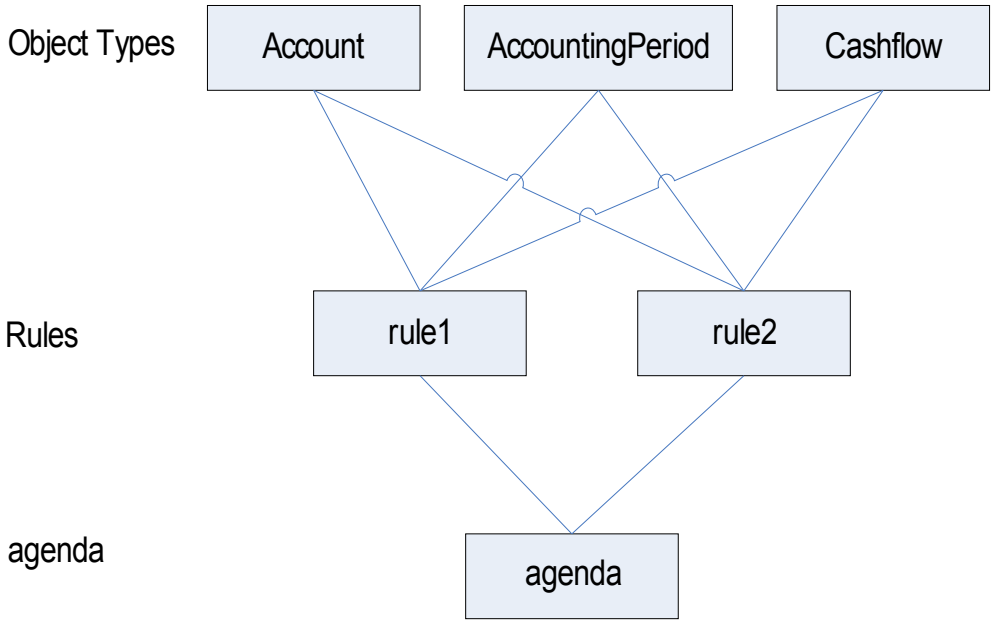
Account	
accountNo	balance
1	150

What is a Production Rule System



Production Rule System Approximated by SQL and Views





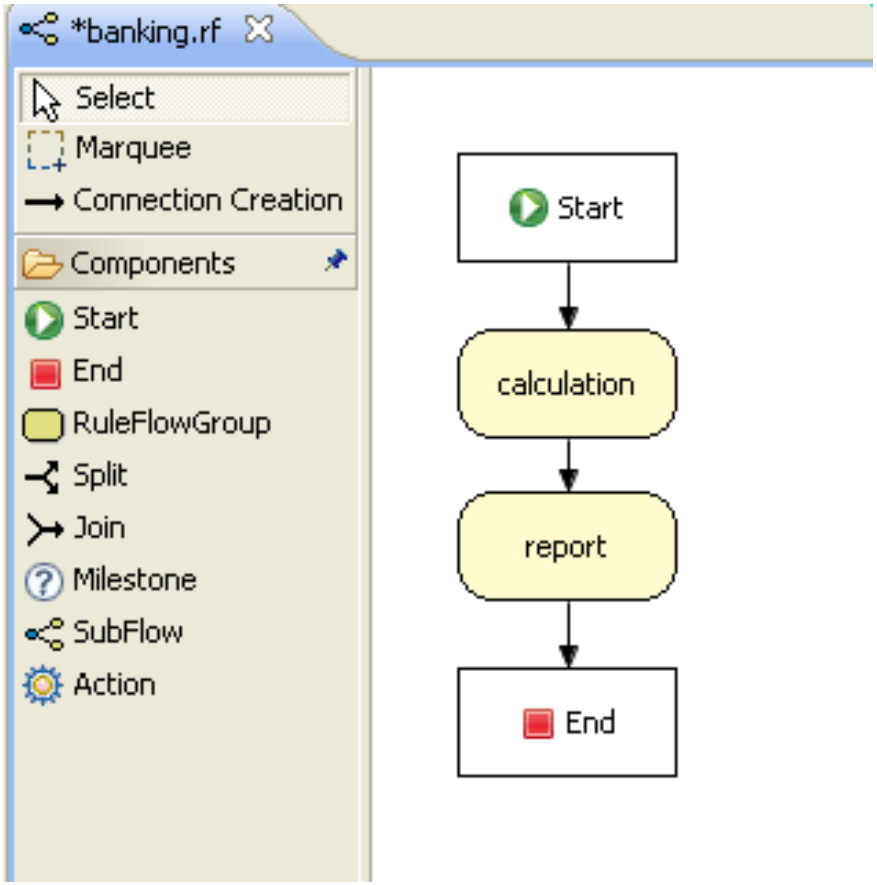
Conflict Resolution with Salience

Salience

```
rule “Print blance for AccountPeriod”
        salience -50
    when
        ap : AccountPeriod()
        acc : Account( )
    then
        System.out.println( acc.accountNo + “ : “ acc.balance );
end
```

Agenda		
1	increase balance	arbitrary
2	decrease balance	
3	increase balance	
4	print balance	

RuleFlow




```
rule "increase balance for AccountPeriod Credits"
      ruleflow-group "calculation"
   when
      ap : AccountPeriod()
      acc : Account( $accountNo : accountNo )
      CashFlow( type == CREDIT,
                accountNo == $accountNo,
                date >= ap.start && <= ap.end,
                $ammount : ammount )
   then
      acc.balance  += $amount;
end
```

ruleflow-group

```
rule "Print blance for AccountPeriod"
      ruleflow-group "report"
   when
      ap : AccountPeriod()
      acc : Account( )
   then
      System.out.println( acc.accountNo + " : " acc.balance );
end
```

Using 'not' existential

```
not Bus( color == “red” )
```

not

```
rule “increase balance for AccountPeriod Credits”
    when
        ap : AccountingPeriod( )
        not AccountingPeriod( start < ap.start)
        acc : Account( $accountNo : accountNo )

        CashFlow( type == CREDIT,
                  accountNo == $accountNo,
                  date >= ap.start && <= ap.end,
                  $ammount : ammount )
    then
        acc.balance  += $amount;
end
```

'not', 'exists', 'forall'

Using 'not' existential

```
rule "Test 01 - Credit Cashflow"
        salience 100
   when
       UpdatingAccount( $account : account )
       CurrentAccountingPeriod( $start : start, $end : end )
       Number( $sum : doubleValue )
            from accumulate( $c : Cashflow( type==Cashflow.CREDIT,
                                            account == $account,
                                            date >= $start && <= $end,
                                            $amount : amount )
                             sum( $c.getAmount() ) )
   then
       System.out.println( "CREDIT : " + $end + " : " + $sum );
       $account.balance += $sum);
end
```

from accumulate

'from', 'collect', 'accumulate'

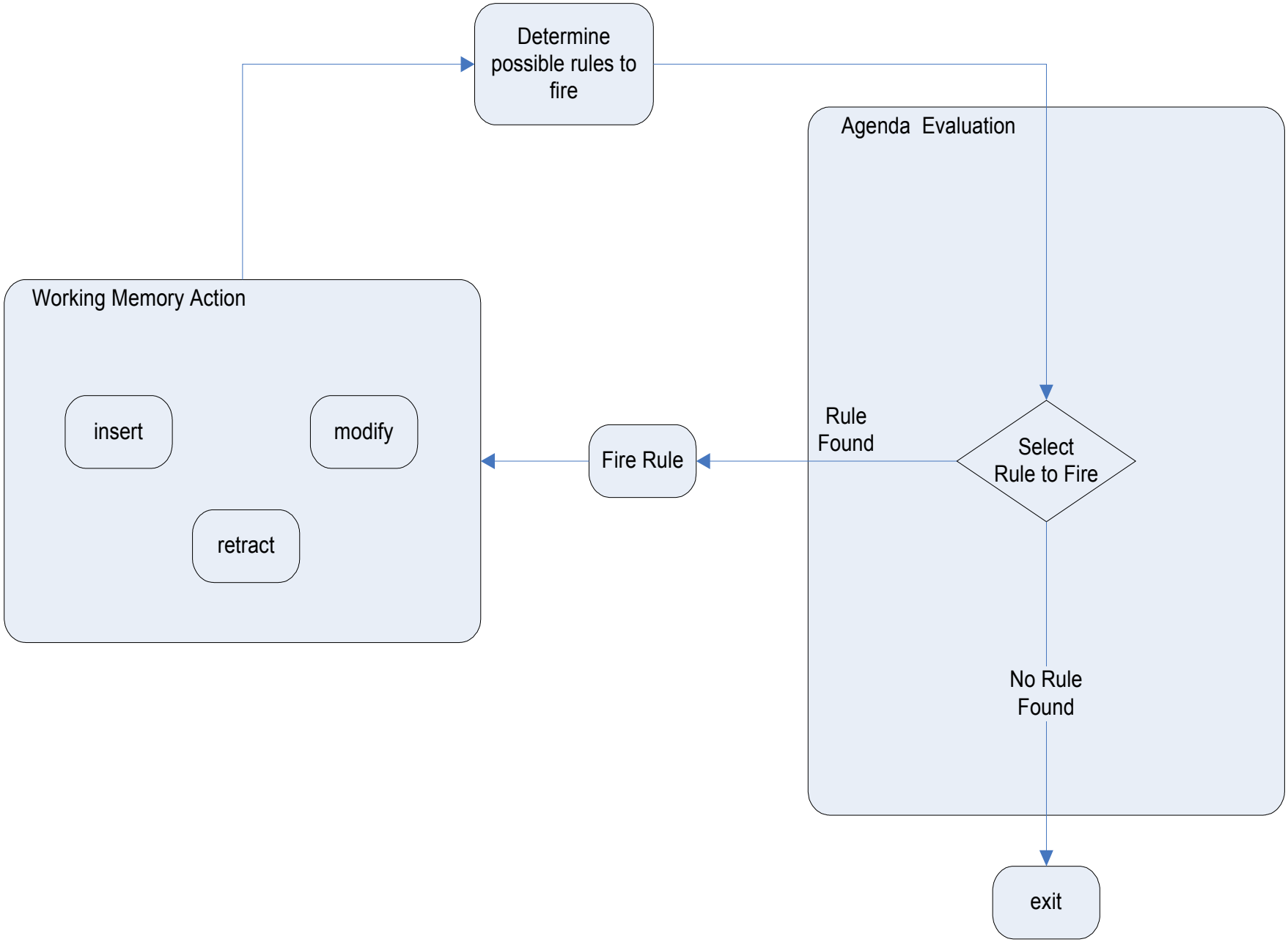
Package

Namespace for all package members

Imports can be used in functions and rules. Uses valid java import syntax

```
package com.sample

import java.util.Map
import com.sample.Cheese

global Cheese cheese

function void exampleFunction(Cheese cheese) {
    System.out.println( cheese );
}

rule   A Cheesy Rule
    when
        &.
    then
        &.
end
```

Two Phase System



Drools | Features

- Engine
  - Full Rete Implementation -- with high performance indexing
  - Dynamic RuleBases
  - Stateful and Stateless Execution Modes
  - Async operations
  - Rete and Sequential Rete
  - Rule Agent
  - Optional Data Shadowing
  - Pluggeable Dialects
- Propositional Logic
  - Literal Restriction
  - Variable Restriction
  - Return Value Restriction
  - Jointed and dis-jointed Connectives allowed - '&&' '||'
  - inline-Eval

Features

- First Order Logic (Quantifiers)
  - And
  - Or
  - Exists
  - Not
  - Accumulate
  - Collect
  - From
  - Forall
  - Nesting of any CE inside of 'and' and 'or'
  - Support for both infix and prefix 'and'/or' CEs
  - Nesting and Chaining of 'from', 'accumulate', 'collect'

Drools Features

- Execution Control
  - Conflict Resolution (salience) Now pluggeable
  - Agenda Filters
  - Agenda Groups
  - Activation Groups
  - Rule Flow
  - Attributes (no-loop, lock-on-active)
- Temporal Rules
  - Scheduler for rule duration will fire when a rule is true for X duration
- Truth maintenance
  - Logical Insertions
- Event Model
  - Working Memory, Agenda, Rule Flow and Rule Base

Eclipse IDE

Debug - StateExampleUsingSalience.drl - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Debug

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

Variables | Breakpoints

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

StateExampleUsingSalience.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" );
```

Text Editor | Rete Tree

Properties

Property	Value
General	
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

Con... | Sea...

StateExampleUsingSalience [Drools

```
A finished
```

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

Eclipse IDE

StateExampleUsingSalience.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" );
```

Text Editor | Rete Tree

Eclipse IDE

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)

Eclipse IDE

Debug - HelloWorld.drl - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Debug | Java

Debug

- HelloWorldExample (1) [Drools Application]
  - org.drools.examples.HelloWorldExample at localhost:1916
    - Thread [main] (Suspended (breakpoint at line 7 in Rule_GoodBye_0))
      - Rule_GoodBye_0.consequence(KnowledgeHelper, HelloWorldExample$Message, FactI
      - Rule_GoodBye_0ConsequenceInvoker.evaluate(KnowledgeHelper, WorkingMemory) li
      - DefaultAgenda.fireActivation(Activation) line: 545
      - DefaultAgenda.fireNextItem(AgendaFilter) line: 509

Variables | Breakpoints

Name	Value
m	HelloWorldExample$Message
message	"Goodbyte cruel world"

Goodbyte cruel world

HelloWorld.drl

```
package org.drools.examples

import org.drools.examples.HelloWorldExample.Message;

rule "Hello World"
    dialect "mvel"
    when
        m : Message( status == Message.HELLO, message : message )
    then
    System.out.println( message );
    modify ( m ) { message = "Goodbyte cruel world",
                   status = Message.GOODBYE };
end

rule "GoodBye"
    dialect "java"
    no-loop true
    when
        m : Message( status == Message.GOODBYE, message : message )
    then
        System.out.println( message );
        m.setMessage( message );
end
```

Text Editor | Rete Tree

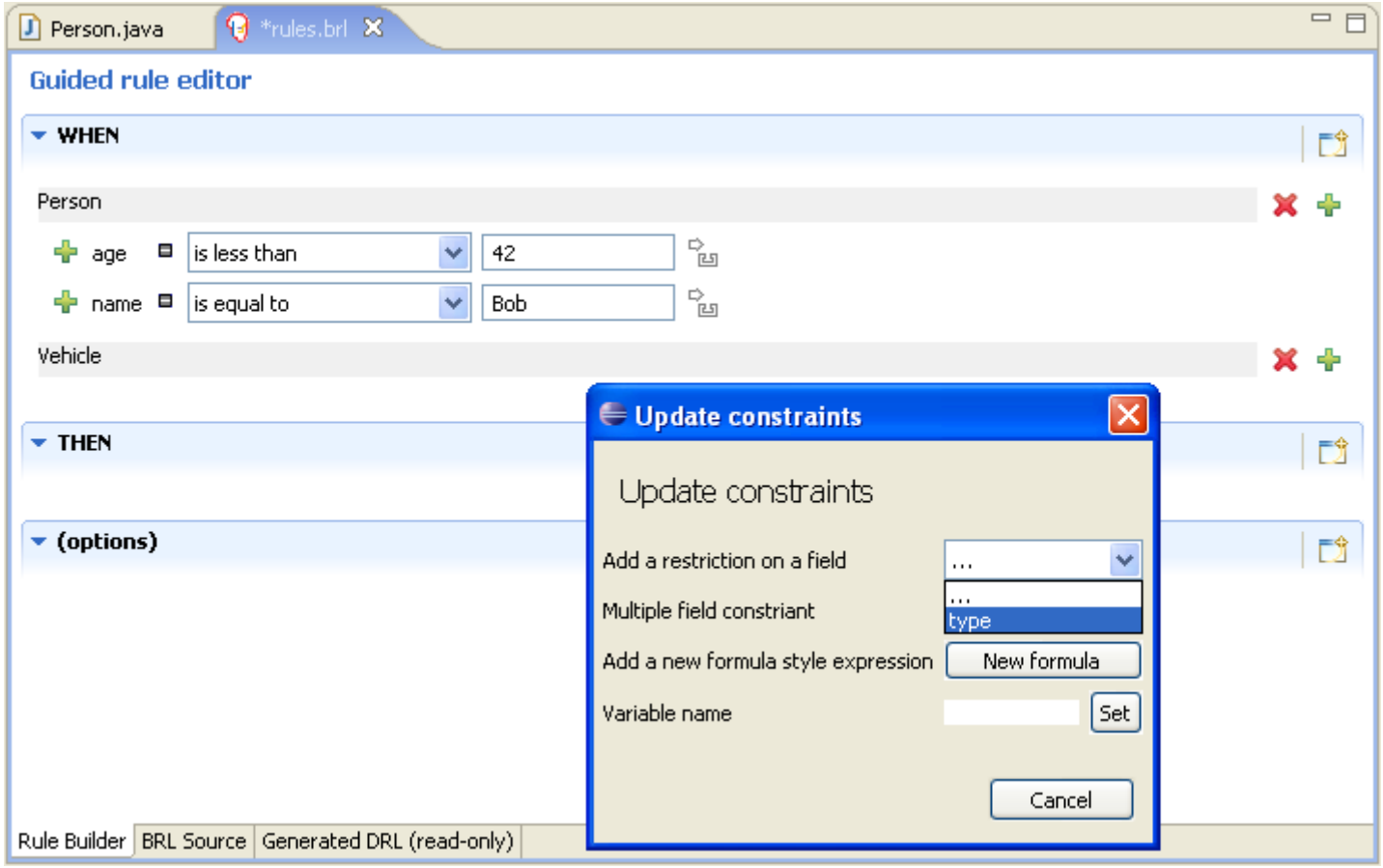
Outline

- org.drools.examples
  - GoodBye
  - Hello World
  - org.drools.examples.HelloWorldExample.Message

Console | Tasks | Agenda View | Global Data View | Rules View | Working Memory View

- [0]= HelloWorldExample$Message (id=24)
  - GOODBYE= 1
  - HELLO= 0
  - message= "Goodbyte cruel world"
  - status= 1

Guided Editor (Eclipse)



DSLs (Eclipse)

```
rule "Driver in unsafe area for marginal age"
    when
        Policy type is 'COMPREHENSIVE'
        Driver is less than 25 years old
        Driver has a location risk profile of 'HIGH'

    then
                                                    rginal age driver in high risk a
end
rule "Dr
    when
                                                    f 'MED'

                                                    ims

    then
        Reject Policy with explanation : 'Driver in that area is too risky -
end

rule "Driver unsafe for third party"
    when
        Policy type is 'THIRD_PARTY'
        Driver has had more than 2 prior claims
```

- Driver has a location risk profile of '{risk}'
- Driver has an age of at least {age}
- Driver has had more than {prior} prior claims
- Driver has had {number} prior claims
- Driver is between {lower} and {upper} years old
- Driver is greater than {age} years old
- Driver is less than {age} years old
- Policy has not been rejected
- Policy type is '{type}'

Text Editor | Rete Tree

DSLs (Eclipse)

ticketing.dsl

Editing Domain specific language: [/drools-examples-drl/src/main/rules/org/drools/examples/ticketing.dsl]

Description:

Language Expression	Rule Language Mapping	Object	Scope
There is a customer ticket with status of "{status}"	customer : Customer( ) ticket : Tick...		[condition]
There is a "{subscription}" customer with a ticket status of "{status}"	customer : Customer(subscription =...		[condition]
Log "{message}"	System.out.println("{message} ");		[consequen...
Escalate the ticket	ticket.setStatus("Escalate"); update...		[consequen...
Send escalation email	sendEscalationEmail(customer, ticke...		[consequen...

Expression: Edit

Mapping: Remove

Object: Add

Sort by: Sort

Copy

Drools Decision Tables (Excel/OpenOffice)

	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

Drools BRMS

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

Gmail | post to del.icio.us | my del.icio.us | Browse Project - JBos... | Google Calendar | Red Hat -- Intranet H... | Nopaste | Brisbane Times - New...

JBoss Rules

- Info
- Rules
- Packages
- Deployment
- Admin

Find and edit rules.

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]

Board

There is no 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

Drools BRMS

JBoss Business Rules Management System

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

User: fmeyer [Sign Out]

JBoss Rules

- Info
- Rules
- Packages
- Deployment
- Admin

Administer the repository

Manage categories | Manage stat

Create a new top level category.

Cateogory name: Fibonacci

Description: fibonacci category

OK | Cancel

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

Current categories:
- HR
- Finance
- Draft

Refresh view:

Create a new category:

Delete the currently selected category:

Drools BRMS

JBoss Business Rules Management System

The page at http://localhost:8080 says:

The snapshot called: NewSnapshot was successfully created.

OK

User: fmeyer [Sign Out]

Info
Rules
Packages
Deployment
Admin

Configure and view package

Explore

DemoPackage
PlayPackage
com.sample
Business rules
Technical rules
Functions
DSL
Model
default

Info
Edit Package configuration
Build, validate and deploy

Create a snapshot for deployment.

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.

Choose or create snapshot name: NEW: NewSnapshot

Comment: commenting

Create new snapshot

Done

BRMS

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

Gmail | post to del.icio.us | my del.icio.us | Browse Project - JBos... | Google Calendar | Red Hat -- Intranet H... | Nopaste | Brisbane Times - New...

Viewing source for: Rule_1

```
rule "Rule_1"
        when
                Person( age <= 42 , age > 21 )
                b : Board( )
                not Board( cost > 1200 )
        then
                b.setCost( 1200 );
end
```

View source | Validate

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

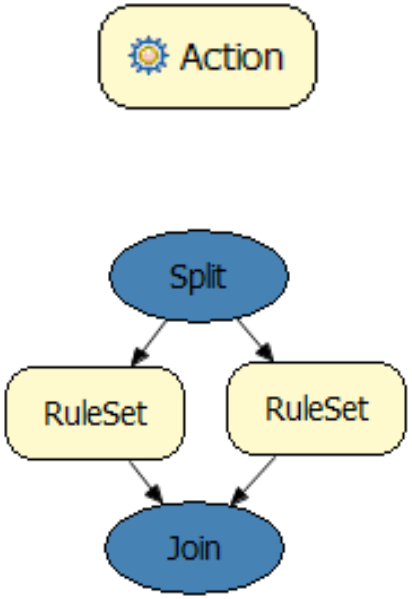
Done

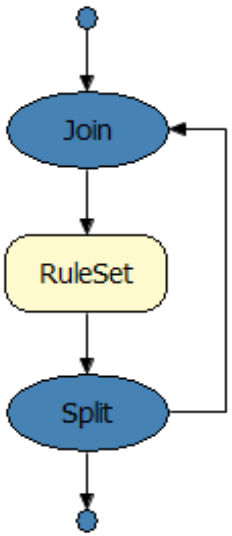
Rule Flow

- Unifies Rules and Processes in a single engine
  - Rules (LHS When) and expressions can be used in splits, milestones etc
    - creates a much richer model
  - Rules and Processes see, reason and react and process the same data
    - Do not have send messages between two different engines
  - Multiple instances, of different processes, can be executing at the same time in a single engine.
  - Processes and Rules interactive with each other.
    - A Process or Rule can change data, which can impact how another rule or process is executing.
  - Integrated Tooling and APIs
    - Single api for execution
    - Audit logging and visual Audit tools
    - Single server side tooling for storage, configuration and management and deployment

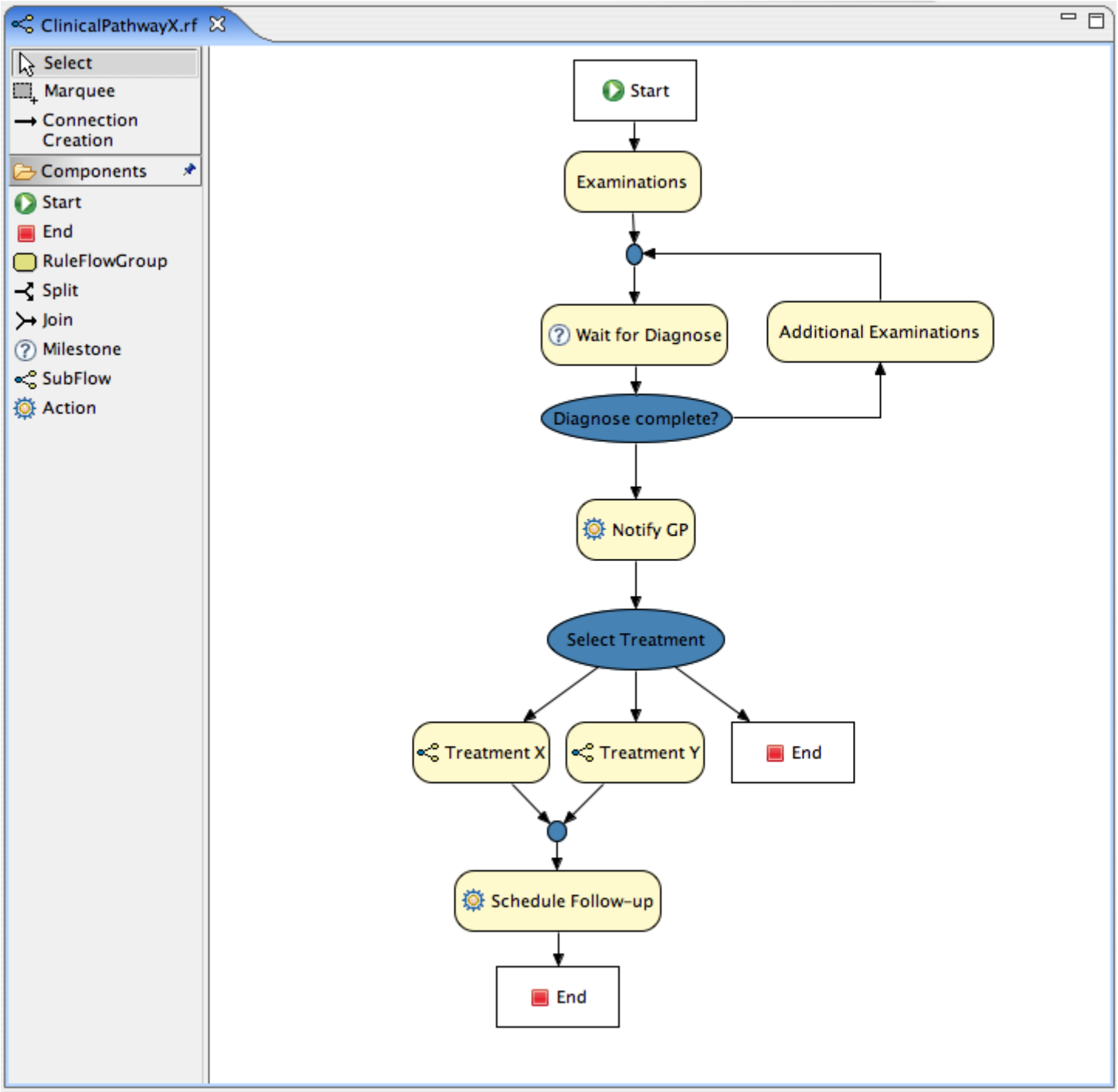
Ruleflow features

- Rule set nodes
- Control flow
  - Sequence
  - Parallelism (split / join)
  - Choice
- Nodes
  - Actions
  - Milestone (= state)
  - Subflows
  - Looping





Rule Flow - Process Diagram

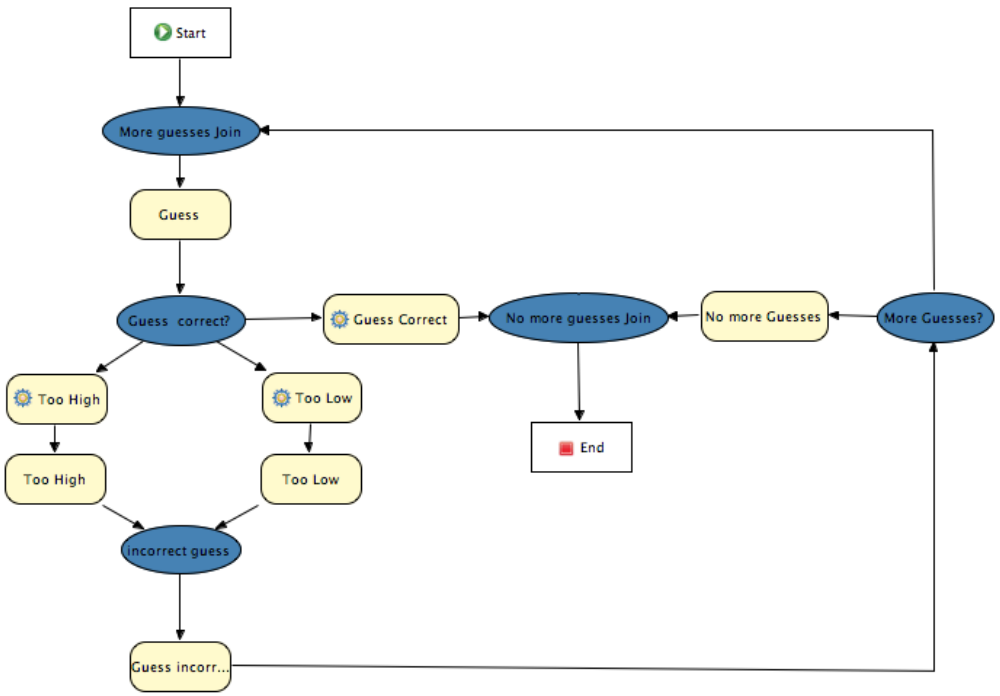


Rule Flow - Rules and Processes

Java – NumberGuess.drl – Eclipse SDK – /Users/fmeyer/projects/droolsprojects

*NumberGuess.rf

Select
Marquee
Connection Creation
Components
Start
End
RuleFlowGroup
Split
Join
Milestone
SubFlow
Action



NumberGuess.drl

```
26          insert( new Guess( i ) );
27  end
28
29  rule "Record the highest Guess"
30      ruleflow-group "Too High"
31      no-loop
32      when
33          game : Game( biggestGuess : biggest )
34          Guess( $value : value > biggestGuess )
35      then
36          modify ( game ) { biggest = $value };
37  end
38
39  rule "Record the lowest Guess"
40      ruleflow-group "Too Low"
41      no-loop
42      when
43          Game( smallestGuess : smallest )
44          Guess( $value : value < smallestGuess )
45      then
46          modify ( game ) { smallest = $value };
47  end
48
49  rule "Guess incorrect, retract Guess"
50      ruleflow-group "Guess incorrect"
51      when
52          guess : Guess()
53      then
54          retract( guess );
55  end
56
57
58  rule "No more Guesses notification"
```

Text Editor | Rete Tree

Package Explorer | Rules View

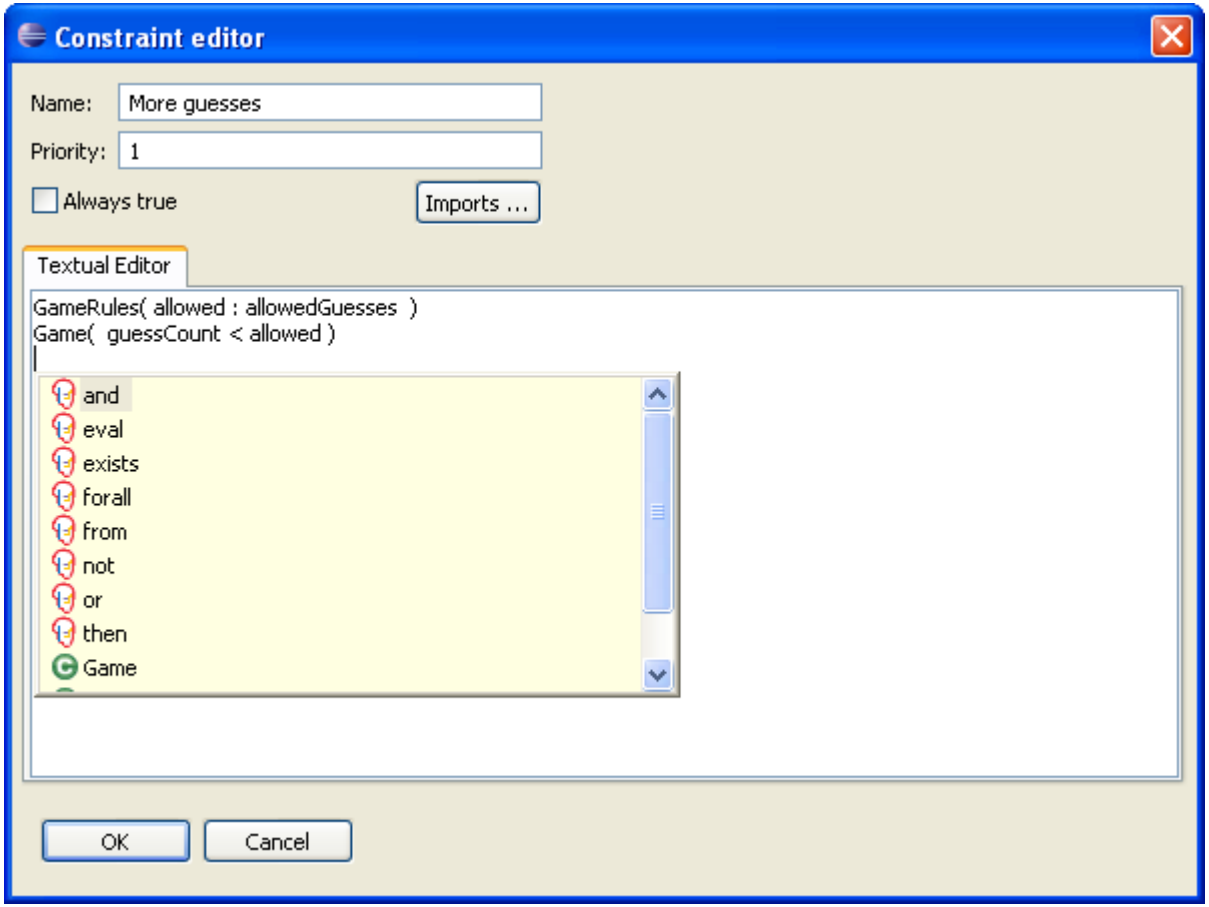
- org.acme.insurance
- org.benchmarks.waltz
- org.drools
- org.drools.benchmark.manners
- org.drools.benchmark.waltzdb
- org.drools.compiler
- org.drools.examples
  - A to B
  - A to B
  - Apply 10% discount if total purcahses is over 100
  - B to C

Outline

- org.drools.examples
  - Get user Guess
  - Guess incorrect, retract Guess
  - No more Guesses notification
  - Record the highest Guess
  - Record the lowest Guess
  - java.io.BufferedReader
  - java.io.InputStreamReader
  - org.drools.examples.NumberGuessExample.Game
  - org.drools.examples.NumberGuessExample.GameRules

Writable | Insert | 60 : 9

Rule Flow - Split Constraint Editor



Unified auditing

- Activation executed: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
  - Object removed (2): Diagnose: Diagnose disease X: Type unknown
    - Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
    - Activation cancelled: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
    - Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
  - RuleFlowGroup activated: Examinations[size=2]
  - RuleFlow started: ClinicalPathwayX[org.drools.examples.cdss.ClinicalPathwayX]
- Activation executed: Rule Examination1
- Activation executed: Rule Examination2
- RuleFlowGroup deactivated: Examinations[size=0]
- RuleFlowGroup activated: AdditionalExaminations[size=2]
- Object inserted (2): Diagnose: Diagnose disease X: Type unknown
  - Activation created: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
  - Activation created: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
  - Activation created: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
  - Activation created: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
- Activation executed: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
  - Object removed (2): Diagnose: Diagnose disease X: Type unknown
    - Activation cancelled: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
    - Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
    - Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
- Activation executed: Rule Examination3
- RuleFlowGroup deactivated: AdditionalExaminations[size=0]
- RuleFlow completed: TreatmentY[org.drools.examples.cdss.TreatmentY]
- RuleFlow started: TreatmentY[org.drools.examples.cdss.TreatmentY]
- RuleFlow completed: ClinicalPathwayX[org.drools.examples.cdss.ClinicalPathwayX]
- Object inserted (2): Diagnose: Diagnose disease X: Type 2

Rule Flow - Unified BRMS

Explore

- org.acme.insurance.base
  - Business rule assets
  - Technical rule assets
  - Functions
  - DSL
  - Model

15 items.

*	Name	Last modified	Status
	Insurance extra itens percent	Sep 20, 2007	Production
	Insurance Calcule	Sep 20, 2007	Production
	Driver is underage	Sep 20, 2007	Production
	New licenced Driver	Sep 20, 2007	Production
	Driver Single Young Male Driver factor	Aug 28, 2007	Production
	Driver Mature Married With Young Child factor	Aug 28, 2007	Production
	Priory Claimed Driver	Aug 28, 2007	Production
	Day Vehicle Place	Aug 28, 2007	Production
	Night Vehicle Place	Aug 28, 2007	Production
	Driver wants an extra Car	Aug 28, 2007	Production
	Driver wants glass coverage	Aug 28, 2007	Production
	Driver wants non related expenses coverage	Aug 28, 2007	Production
	insuranceProcess	Aug 28, 2007	Production
	approve	Aug 28, 2007	Production
	rejection	Aug 28, 2007	Production

Whats coming in Q1?

- Engine
  - Stateful High Availability
- Event Stream Processing, Complex Event Processing
  - time windows (fixed, since, until)
  - date comparisons between objects (before, same, after)
- RuleFlow
  - Persistence
  - Timers
  - More complex workflow patterns
  - Pluggeable tasks
- BRMS
  - UI improvements
  - ACL Security
  - Scenario Testing
  - Decision Tables

RuleFlow Pluggeable Tasks

RuleFlowTest.java | ruleflow.rf | ruleflow.drl

Select
Marquee
Connection Creation
Components
Start
End
RuleFlowGroup
Split
Join
Milestone
SubFlow
Action
Tasks
Log
Email

Start → Split → Email → Join → Log → End
Split → Log → Join

Problems | Properties | Audit View | Error Log | Cons

Property
From
Name
Subject
Text
To

New Application

general | attachments | connection | error handling

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

BRMS UI Improvements

Rules | Packages | Deployment | Admin

- Drools BRMS
  - Packages
    - org.drools.example.BRMS
      - Business Rules
      - Tecnical Rules
      - Rule Flows
      - Function
      - DSL
      - Model
    - org.drools.example.Insurace
      - Business Rules
      - Tecnical Rules
      - Rule Flows
      - Function
      - DSL
      - Model
    - org.drools.example.Sudoku
      - Business Rules
      - Tecnical Rules
      - Rule Flows
      - Function
      - DSL
      - Model

Rules | Packages | Deployment | Admin

- Drools BRMS
  - Categories
    - Insurance
      - Calculate
      - Risk Assesment
  - Status
    - New
    - Draft
    - Production
    - Inactive
  - Analytics
    - Business Rules

Rules | Packages | Deployment | Admin

- Drools BRMS
  - Admin
    - Categories
    - Archived Assets
    - State
    - Import/Export
    - Users
    - Security

Demo

Questions?

- Dave Bowman: All right, HAL; I'll go in through the emergency airlock.
- HAL: Without your space helmet, Dave, you're going to find that rather difficult.
- Dave Bowman: HAL, I won't argue with you anymore! Open the doors!
- HAL: Dave, this conversation can serve no purpose anymore. Goodbye.

Joshua: **Greetings, Professor Falken.**
Stephen Falken: **Hello, Joshua.**
Joshua: **A strange game. The only winning move is not to play. How about a nice game of chess?**