Drools
BeJUG 2010

Kris Verlaenen

Geoffrey De Smet

Agenda

Introduction

Drools Expert: rule engine

Drools Guvnor: knowledge repository

Drools Planner: automated planning

Drools Flow: workflow engine

Drools Fusion: complex event processing

Summary

Drools Platform

Introduction

Drools History

- Drools 2
  - Rete like XML Scripting language
- Drools 3
  - Based on Clips functionality
  - Iterative improves to JRules syntax with Clips functionality
- Drools 4
  - More declarative
  - Basic Rule Flow
  - Basic BRMS

Drools 5: BLiS

Drools Expert

Drools Flow

Drools Fusion

Drools Guvnor

Business Logic integration System

Motivation

A business solution usually involves the interaction between these technologies.

- Technology overlap
- Business overlap

Several (good) products on the market, better either at rule or business process or event processing

Attribute the same importance to the three complementary business modeling techniques

Drools Expert

Rule engine

Drools Expert agenda

Use case example

Initialization and usage

Rules

Drools Expert agenda

Use case example

Initialization and usage

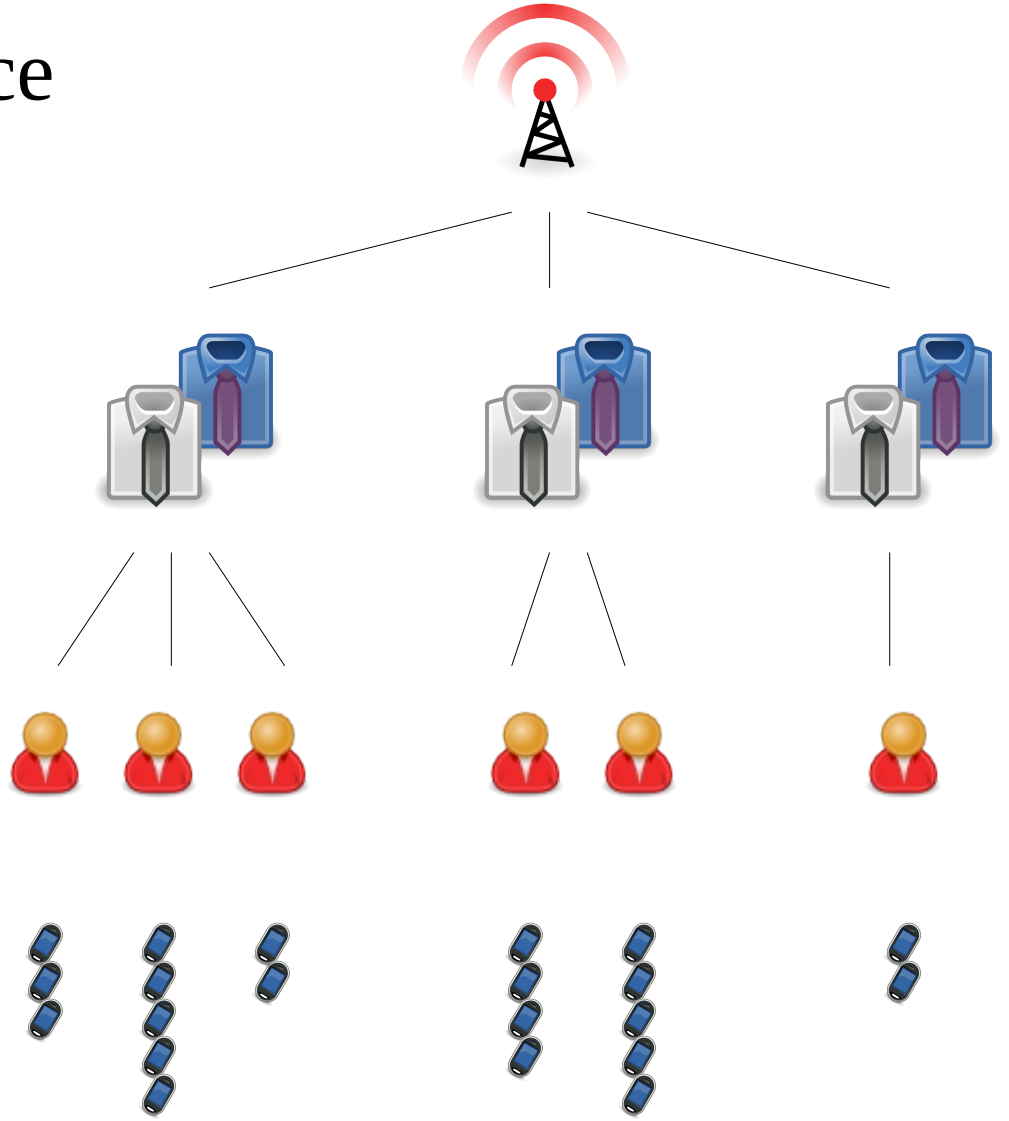
Rules

Learn by example

- Bob's phone call service
  - Network hardware/software works
- Bob's problem
  - Calculate phone call invoice
    - Per customer
  - Calculation rules change regularly
    - Discounts, promotions, …
    - *If you call someone 30 minutes on Saturday and Sunday then you can call him/her 10 free minutes on Monday!*

Customers and subscribers

- Bob's phone call service
  - 1 Network
- Customer: **1 invoice**
  - Company A
  - Company B
- Subscriber
  - Fred of A
  - Greg of A
- Phone call
  - Call 1 of Fred
  - Call 2 of Fred

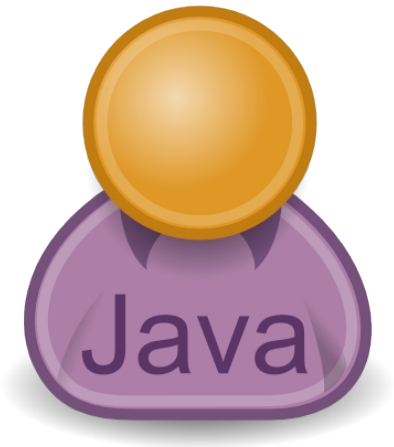


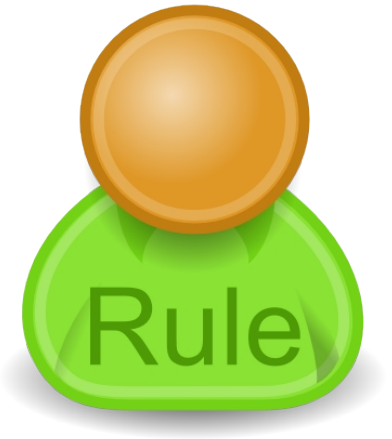
Bob's problem

Calculate phone call invoice

Bob's solution

- Hire 3 consultants
  - Java
  - SQL (MySQL, HSQLDB, ...)
    - JPA (Hibernate, TopLink, ...)
  - Rule engine (Drools, Jess, ...)





Drools Expert agenda

Use case example

Initialization and usage

Rules

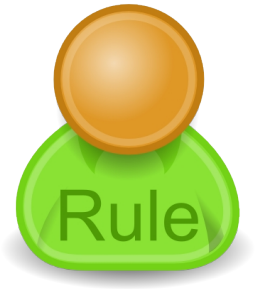
Initializing SQL

Build a DataSource

- Thread-safe singleton
- Usually light weight

```
private DataSource dataSource;

private void initSql() {
  jdbcDataSource jdbcDataSource = new jdbcDataSource();
  jdbcDataSource.setDatabase("jdbc:hsqldb:mem:droolsphone");
  jdbcDataSource.setUser("sa");
  jdbcDataSource.setPassword("");
  this.dataSource = jdbcDataSource;
}
```

Initializing Drools



Build a KnowledgeBase

- Thread-safe singleton
- Heavy weight

```
private KnowledgeBase knowlegdeBase;

private void initDrools() {
  KnowledgeBuilder builder
      = KnowledgeBuilderFactory.newKnowledgeBuilder();
  builder.add(ResourceFactory.newClassPathResource(
          "org/drools/examples/droolsphone/phoneRules1.drl"),
      ResourceType.DRL);
  if (builder.hasErrors()) {
    throw new IllegalStateException(builder.getErrors().toString());
  }

  knowledgeBase = KnowledgeBaseFactory.newKnowledgeBase();
  knowledgeBase.addKnowledgePackages(builder.getKnowledgePackages());
}
```

Using SQL

- Build a Connection
  - Not thread-safe
  - Heavy weight (pooling needed)
  - 1 Transaction per service call
- Execute query's

```
public void chargeWithSql() {
  Invoice invoice = new Invoice();
  try {
    Connection connection = dataSource.getConnection();
    PreparedStatement statement = connection.prepareStatement(...);
    ResultSet resultSet = statement.executeQuery();
    ...
  } catch (SQLException e) {
    throw new IllegalStateException("SQL example failed.", e);
  }
  System.out.println(invoice.toString());
}
```

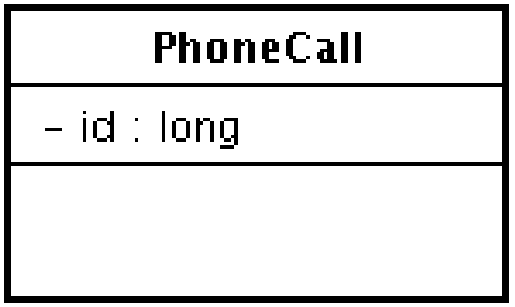
Using SQL: fill database

Data not in database

NoSQL, webservice, XML file, ...

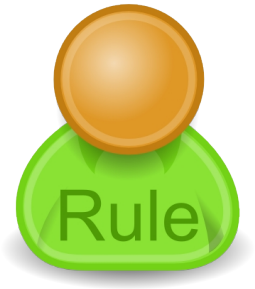
Data not in the correct model

Extra indexes, temp tables, ...

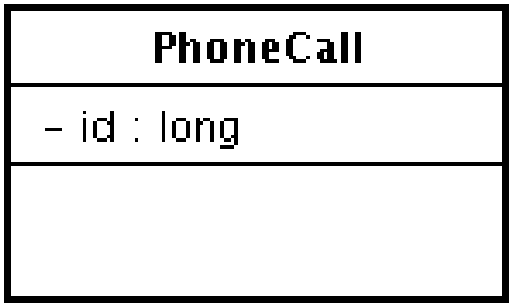



```
public void fillDatabase(List<PhoneCall> phoneCallList) {
  try {
    Connection connection = jdbcDataSource.getConnection();
    connection.createStatement().executeUpdate(
            "CREATE TABLE PhoneCall(id BIGINT PRIMARY KEY)");
    PreparedStatement phoneCallInsert = connection.prepareStatement(
            "INSERT INTO PhoneCall(id) VALUES (?)");
    for (PhoneCall phoneCall : phoneCallList) {
        phoneCallInsert.clearParameters();
        phoneCallInsert.setLong(1, phoneCall.getId());
        phoneCallInsert.executeUpdate();
    }
  } catch (SQLException e) {...}
}
```

Using Drools (Stateless)

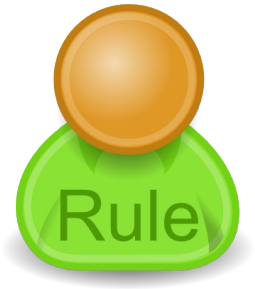


- Build a KnowledgeSession
  - Not thread-safe
  - Light weight
  - 1 stateless KnowlegdeSession per service call
- Execute on facts
  - Fact = POJO inserted into the session




```
public void chargeWithDrools(List<PhoneCall> phoneCallList) {
  Invoice invoice = new Invoice();

  StatelessKnowledgeSession session
      = knowledgeBase.newStatelessKnowledgeSession();
  session.setGlobal("invoice", invoice);
  session.execute(phoneCallList); // Insert facts and fire all rules

  System.out.println(invoice.toString());
}
```

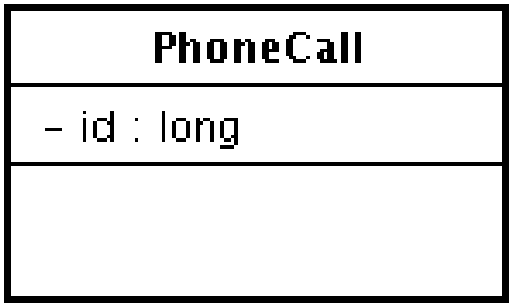
Using Drools (Stateful)



Stateful splits up:

- insert/update/remove facts
- Fire all rules on those facts

Session is reusable across service calls




```
public void chargeWithDrools(List<PhoneCall> phoneCallList) {
  Invoice invoice = new Invoice();

  StatefulKnowledgeSession session
      = knowledgeBase.newStatefulKnowledgeSession();
  session.setGlobal("invoice", invoice);
  for (PhoneCall p : phoneCallList) {
    session.insert(p);
  }
  session.fireAllRules();

  System.out.println(invoice.toString());
}
```

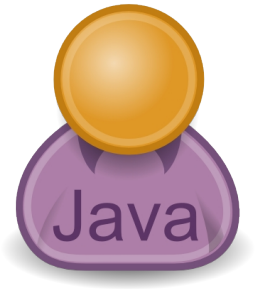
Drools Expert agenda

Use case example

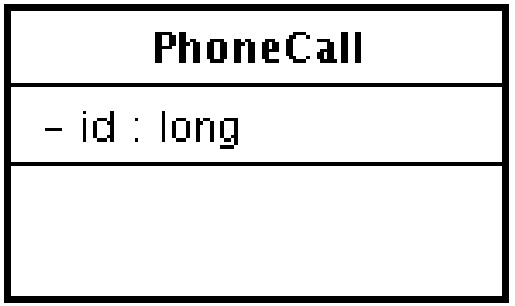
Initialization and usage

Rules

One euro per phone call: java



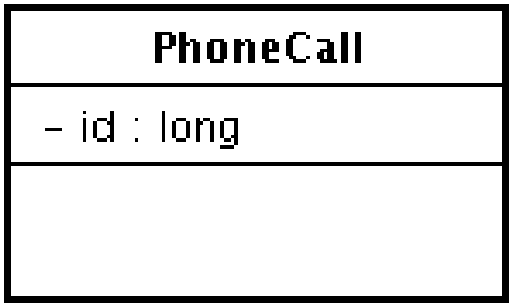
Each phone call costs 1 euro




```
Invoice invoice = new Invoice();
...

// Find PhoneCalls
for (PhoneCall p : phoneCallList) {

  // Charge PhoneCalls
  System.out.println("  Charging phone call with Java.");
  invoice.addPrice(new BigDecimal("1.00"));

}
```

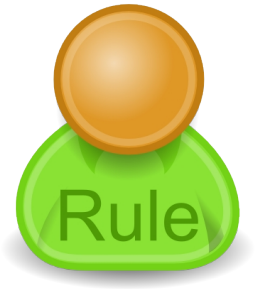
One euro per phone call: sql

Each phone call costs 1 euro

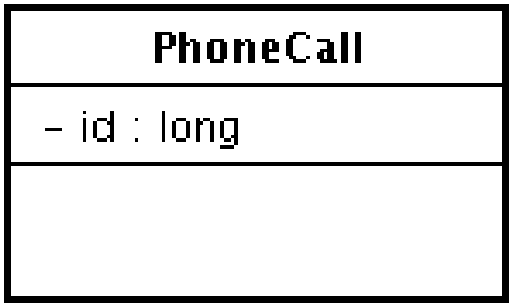



```
Invoice invoice = new Invoice();
...

// Find PhoneCalls
PreparedStatement statement = connection.prepareStatement(
    "SELECT count(*) * 1.00"
    + "  FROM PhoneCall");
ResultSet resultSet = statement.executeQuery();
resultSet.next();

// Charge PhoneCalls
System.out.println("  Charging phone calls with SQL.");
BigDecimal total = resultSet.getBigDecimal(1);
invoice.setTotal(total);
```

One euro per phone call: rule

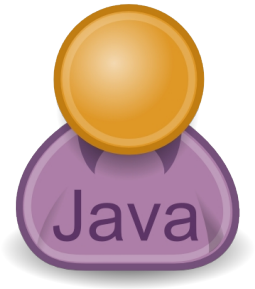


Each phone call costs 1 euro



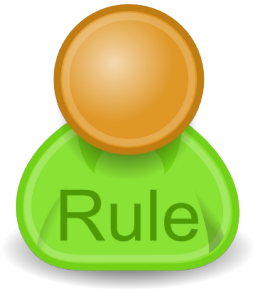

```
global Invoice invoice;
...

rule "oneEuroPerPhoneCall"

  // Find PhoneCalls
  when // Left hand side (DRL pattern syntax)
    PhoneCall()

  // Charge PhoneCalls
  then // Right hand side (Java code)
    System.out.println("  Charging phone call with Drools.");
    invoice.addPrice(new BigDecimal("1.00"));

end
```

One euro per phone call



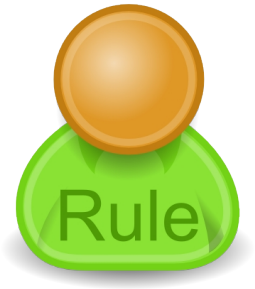

```java
for (PhoneCall p : phoneCallList) {
  System.out.println("  Charging phone call with Java.");
  invoice.addPrice(new BigDecimal("1.00"));
}
```


```
PreparedStatement statement = connection.prepareStatement(
    "SELECT count(*) * 1.00"
    + "  FROM PhoneCall");
ResultSet resultSet = statement.executeQuery();
resultSet.next();
System.out.println("  Charging phone calls with SQL.");
BigDecimal total = resultSet.getBigDecimal(1);
invoice.setTotal(total);
```

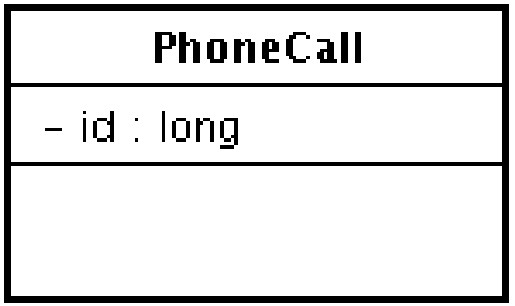



```
rule "oneEuroPerPhoneCall"
  when // Left hand side (DRL pattern syntax)
    PhoneCall()
  then // Right hand side (Java code)
    System.out.println("  Charging phone call with Drools.");
    invoice.addPrice(new BigDecimal("1.00"));
end
```

Pattern binding

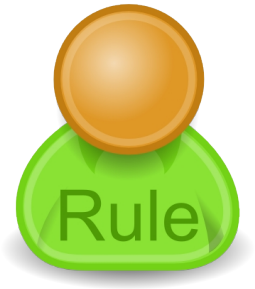


Variables have a dollar sign




```
rule "oneEuroPerPhoneCall"

  when
    $p : PhoneCall() // Pattern binding

  then
    System.out.println("  Charging phone call (" + $p.getId()
        + ") with Drools.");
    invoice.addPrice(new BigDecimal("1.00"));

end
```

Literal restriction

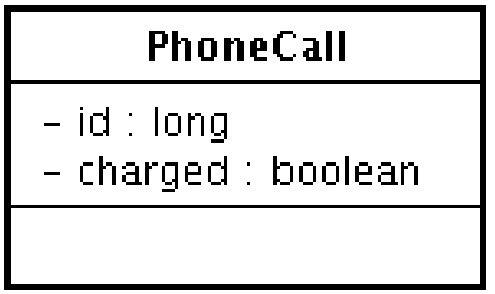


charged uses the getter

public boolean isCharged() {…}

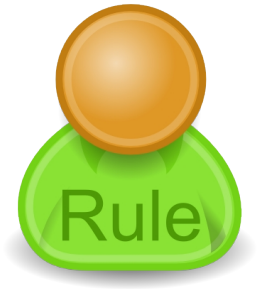
== uses equals() method

except for primitive types



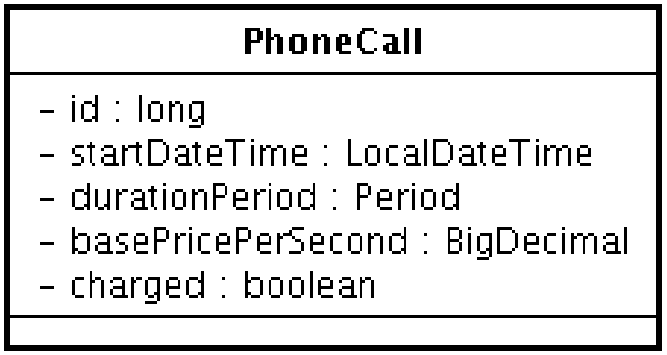

```
rule "oneEuroPerPhoneCall"

  when
    $p : PhoneCall(charged == false) // Literal restriction

  then
    $p.setCharged(true);
    invoice.addPrice(new BigDecimal("1.00"));

end
```

Multiple restrictions



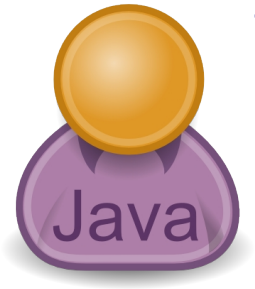
Multiple restrictions

AND logic



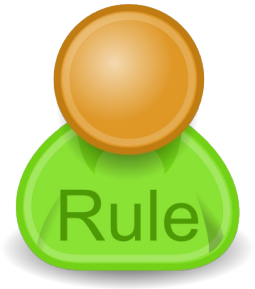

```
rule "morningPhoneCall"
  when
    $p : PhoneCall(charged == false, startDateTime.hourOfDay < 12)
  then
    modify ($p) {
      setBasePricePerSecond(new BigDecimal("0.05"));
    }
end

rule "afternoonPhoneCall"
  when
    $p : PhoneCall(charged == false, startDateTime.hourOfDay >= 12)
  then
    modify ($p) {
      setBasePricePerSecond(new BigDecimal("0.20"));
    }
end
```

Morning and afternoon calls 1/2



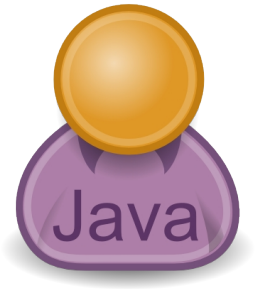

```
for (PhoneCall p : phoneCallList) {
  if (!p.isCharged()) {
    if (p.getStartDateTime().getHourOfDay() < 12) {
        p.setBasePricePerSecond(new BigDecimal("0.05"));
    } else {...}
  }
}
```


```
PreparedStatement statement = connection.prepareStatement(
        "SELECT SUM(0.05 * durationPeriod)"
        + "  FROM PhoneCall"
        + "  WHERE charged = 0"
        + "  AND EXTRACT (HOUR FROM startDateTime) < 12");
```

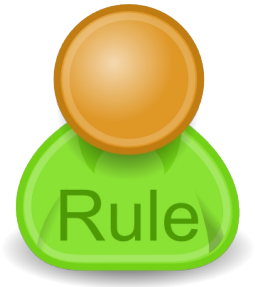



```
rule "morningPhoneCall"
  when
    $p : PhoneCall(charged == false,
        startDateTime.hourOfDay < 12)
  then
    modify ($p) {
      setBasePricePerSecond(new BigDecimal("0.05"));
    }
end
```

Morning and afternoon calls 2/2




```
for (PhoneCall p : phoneCallList) {
  if (!p.isCharged() && p.getBasePricePerSecond() != null) {
    BigDecimal price = p.getBasePricePerSecond().multiply(
        BigDecimal.valueOf(p.getDurationPeriodInSeconds()));
    invoice.addPrice(price);
    p.setCharged(true);
  }
}
```




```
rule "chargePhoneCalls"
  when
    $p : PhoneCall(charged == false,
        basePricePerSecond != null)
  then
    BigDecimal price = $p.getBasePricePerSecond().multiply(
        BigDecimal.valueOf($p.getDurationPeriodInSeconds()));
    invoice.addPrice(price);
    modify ($p) {
      setCharged(true);
    }
end
```

DRL versus Java

- DRL is declarative
  - Java is imperative
- Find a pattern in a String
  - Use a regular expression (declarative)

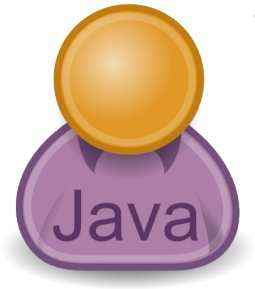
```
"\d+(\.\d+)*"
```

  - Process the result with Java
- Find a pattern in a Object heap
  - Use a rule engine (declarative)

```
PhoneCall(charged == false,
    startDateTime.hourOfDay >= 12)
```

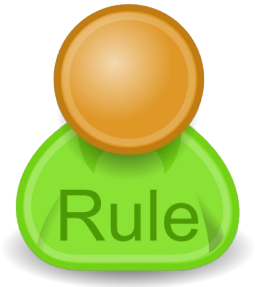
  - Process the result with Java

Holiday calls



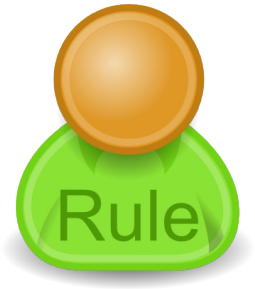

```
for (PhoneCall p : phoneCallList) {
  if (!p.isCharged()) {
    if (HolidayUtil.isHoliday(p.getStartDate())) {
      p.setBasePricePerSecond(new BigDecimal("0.07"));
    } else {...}
  }
}
```

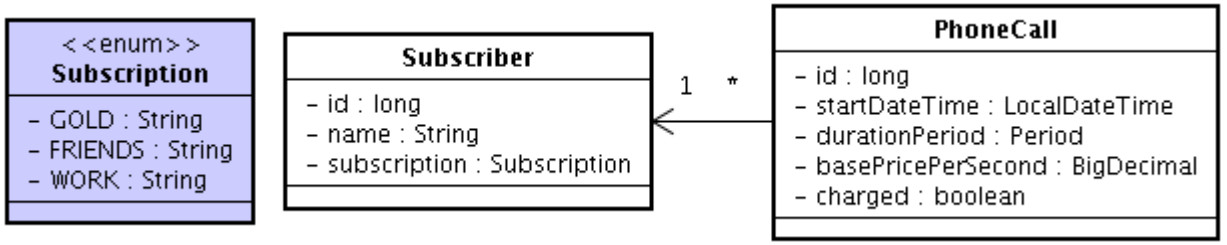

```
// SQL has no isHoliday function
```




```
rule "holidayCall"
  when
    $p : PhoneCall(charged == false,
         eval(HolidayUtil.isHoliday(startDate)))
  then
    modify ($p) {
      setBasePricePerSecond(new BigDecimal("0.07"));
    }
end
```

Subscriptions: rules



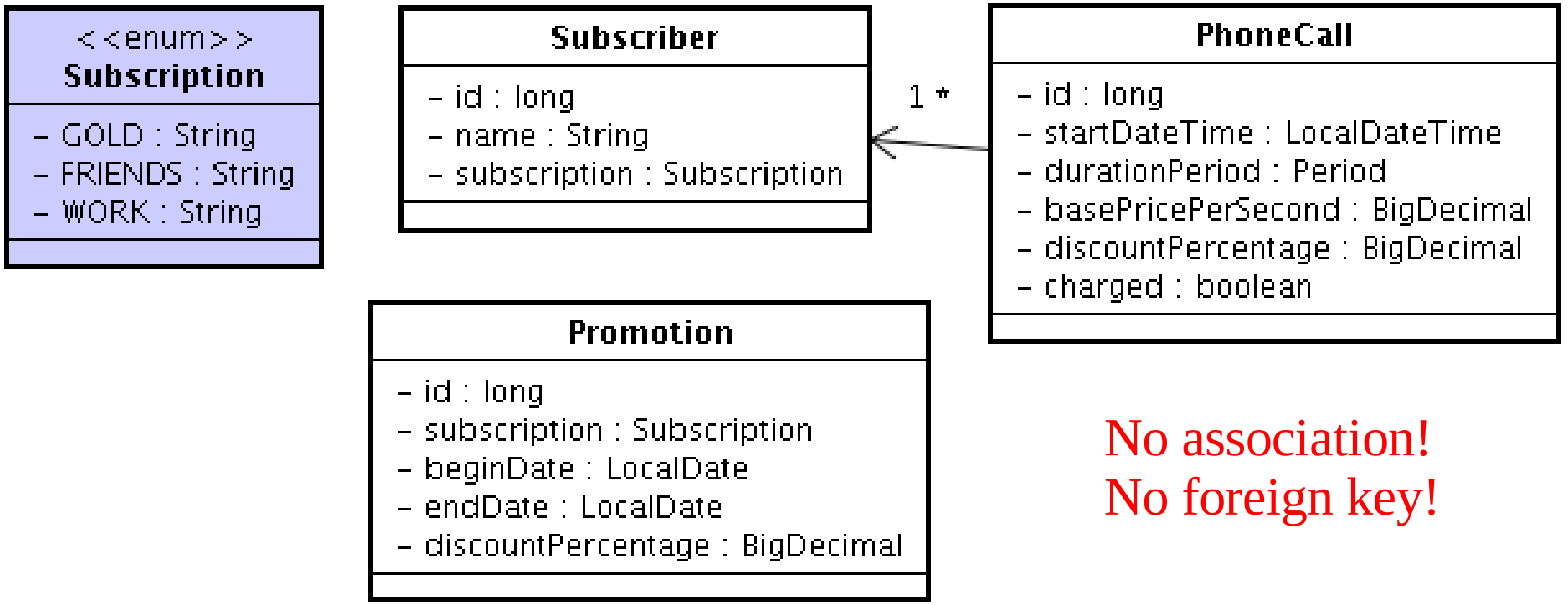



```
rule "gold basePricePerSecond" when
    $s : Subscriber(subscription == Subscription.GOLD)
    $p : PhoneCall(subscriber == $s)
  then ... setBasePricePerSecond 0.05 ... end

rule "friends basePricePerSecond good days" when
   $s : Subscriber(subscription == Subscription.FRIENDS)
   $p : PhoneCall(subscriber == $s,
        startDateTime.dayOfWeek == DateTimeConstants.SATURDAY
            || startDateTime.dayOfWeek  == DateTimeConstants.SUNDAY)
  then ... setBasePricePerSecond 0.01 ... end

rule "work basePricePerSecond good hours" when
    $s : Subscriber(subscription == Subscription.WORK)
    $p : PhoneCall(subscriber == $s,
        startDateTime.hourOfDay >= 9 && < 17)
  then ... setBasePricePerSecond 0.02 ... end
```

Promotions are a problem

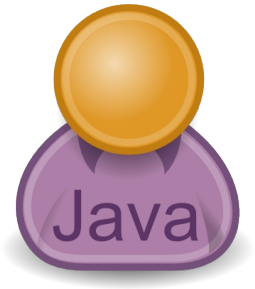
Promotion: discount percentage

- For subscription
- Between beginDate and endDate



No association!
No foreign key!

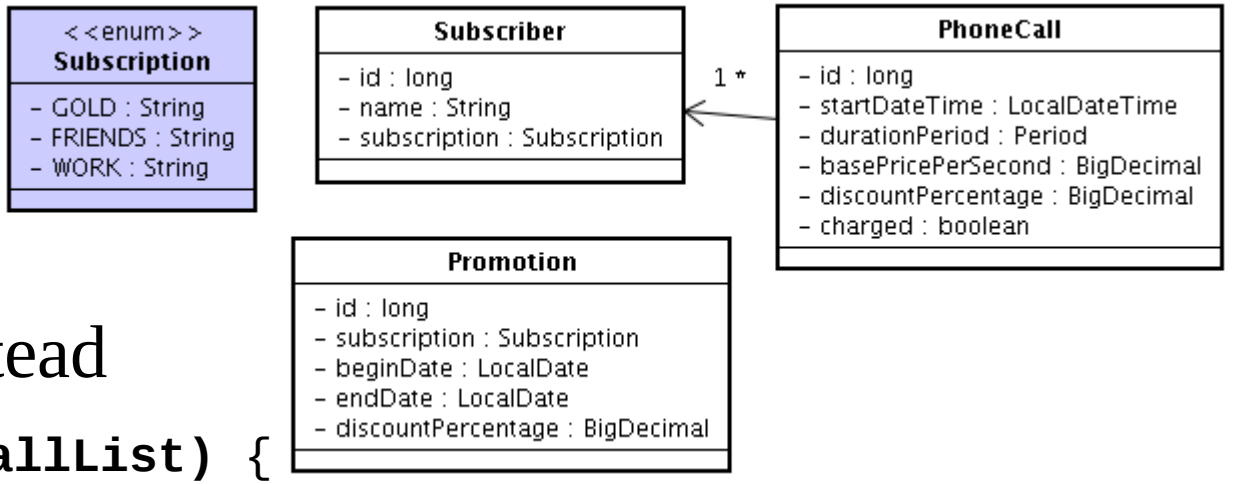
Promotions: java



Unreadable

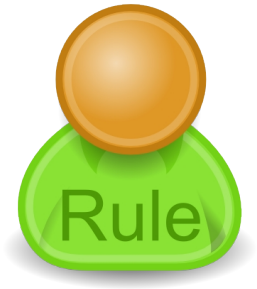
Not scalable

Use Map instead



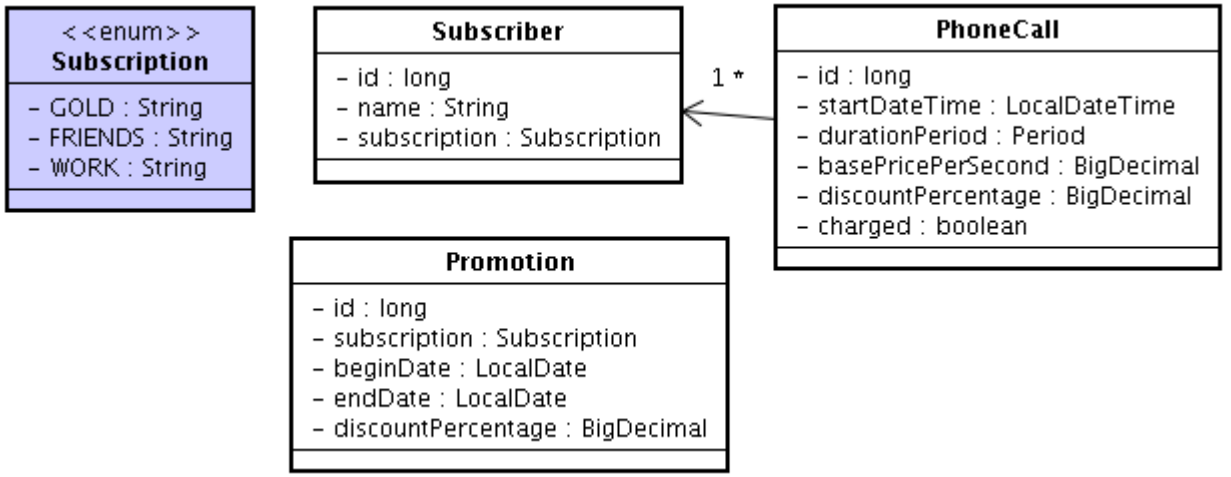

```
for (PhoneCall p : phoneCallList) {
  LocalDate startDate = p.getStartDate();
  Promotion promo = null;
  // TODO Does not scale, look up in Map instead!
  for (Promotion possiblePromo : promotionList) {
    if (p...getSubscription().equals(possiblePromo.getSubscription())
        && possiblePromo.getBeginDate().compareTo(startDate) <= 0
        && possiblePromo.getEndDate().compareTo(startDate) >= 0) {
      promo = possiblePromo; break;
    }
  }
  if (promo != null) {
    // Apply discount of promotions
    p.setDiscountPercentage(promo.getDiscountPercentage());
  } else {...}
}
```

Promotions: rules



Readable

Scalable




```
rule "Apply discount of promotions"
  when
    $p : PhoneCall($subscription : subscriber.subscription,
         $startDate : startDate)
    $pr : Promotion(subscription == $subscription,
         beginDate <= $startDate, endDate >= $startDate)
  then ... setDiscountPercentage $pr.getDiscountPercentage() ... end

rule "Apply discount without promotion"
  when
    $p : PhoneCall($subscription : subscriber.subscription,
         $startDate : startDate)
    not Promotion(subscription == $subscription,
         beginDate <= $startDate, endDate >= $startDate)
  then ... setDiscountPercentage 0.00 ... end
```

DRL is powerful

Conditional elements

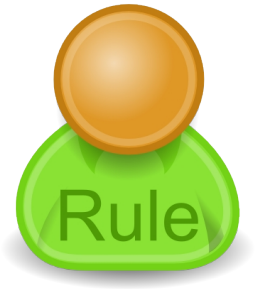
- not, exists, forall

from

- collect
- accumulate
  - sum, max, … (and custom functions)
- hibernate query

```
rule "Find People for given zip code"
when
  $zipCode : ZipCode()
  Person() from $hibernate.getNamedQuery("Find People")
                          .setParameters( [ "zipCode" : $zipCode ] )
                          .list()
```

Charge only longest call per hour



salience

Rules with higher salience are run first

A rule can stop another rule from firing

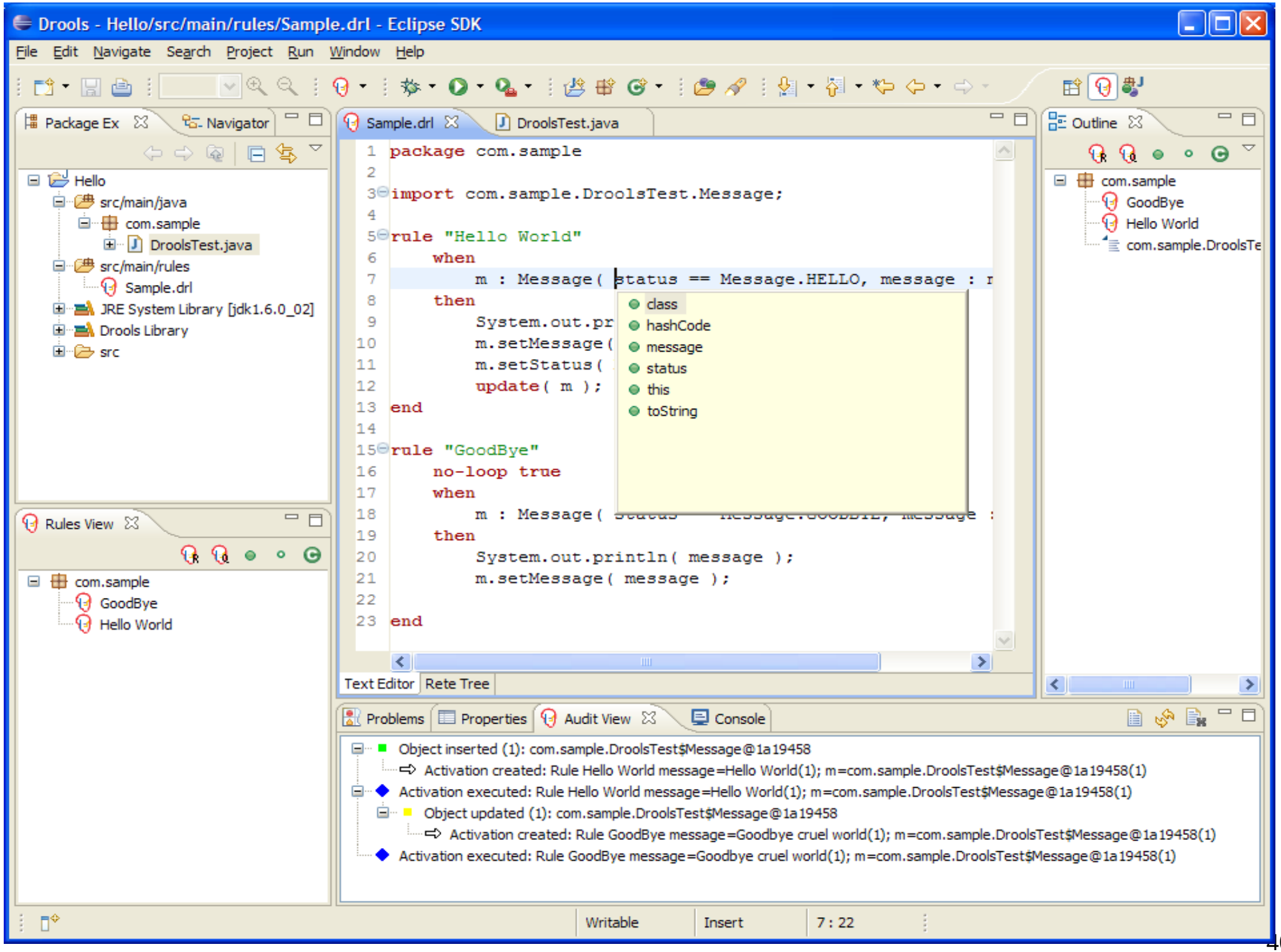
```
rule "only longest call per hour is charged"
    salience 10 // Do this rule first
  when
    $p : PhoneCall(basePricePerSecond == null,
        $id : id, $s : subscriber,
        $startDate : startDate, $hourOfDay : startDateTime.hourOfDay,
        $durationPeriodInSeconds : durationPeriodInSeconds)
    exists PhoneCall(id != $id, subscriber == $s,
        startDate == $startDate,
        startDateTime.hourOfDay == $hourOfDay,
        durationPeriodInSeconds > $durationPeriodInSeconds)
  then
    modify ($p) {
      setBasePricePerSecond(new BigDecimal("0.00"));
    }
end
```

Drools Eclipse

Eclipse-based Tooling

Features

- Wizards
- Drools runtimes (classpath)
- Textual Rule Editor
  - Syntax highlighting, code completion, error detection, outline view, etc.
- Guided Editor
- Debugging
  - Rule breakpoints
  - Debug views

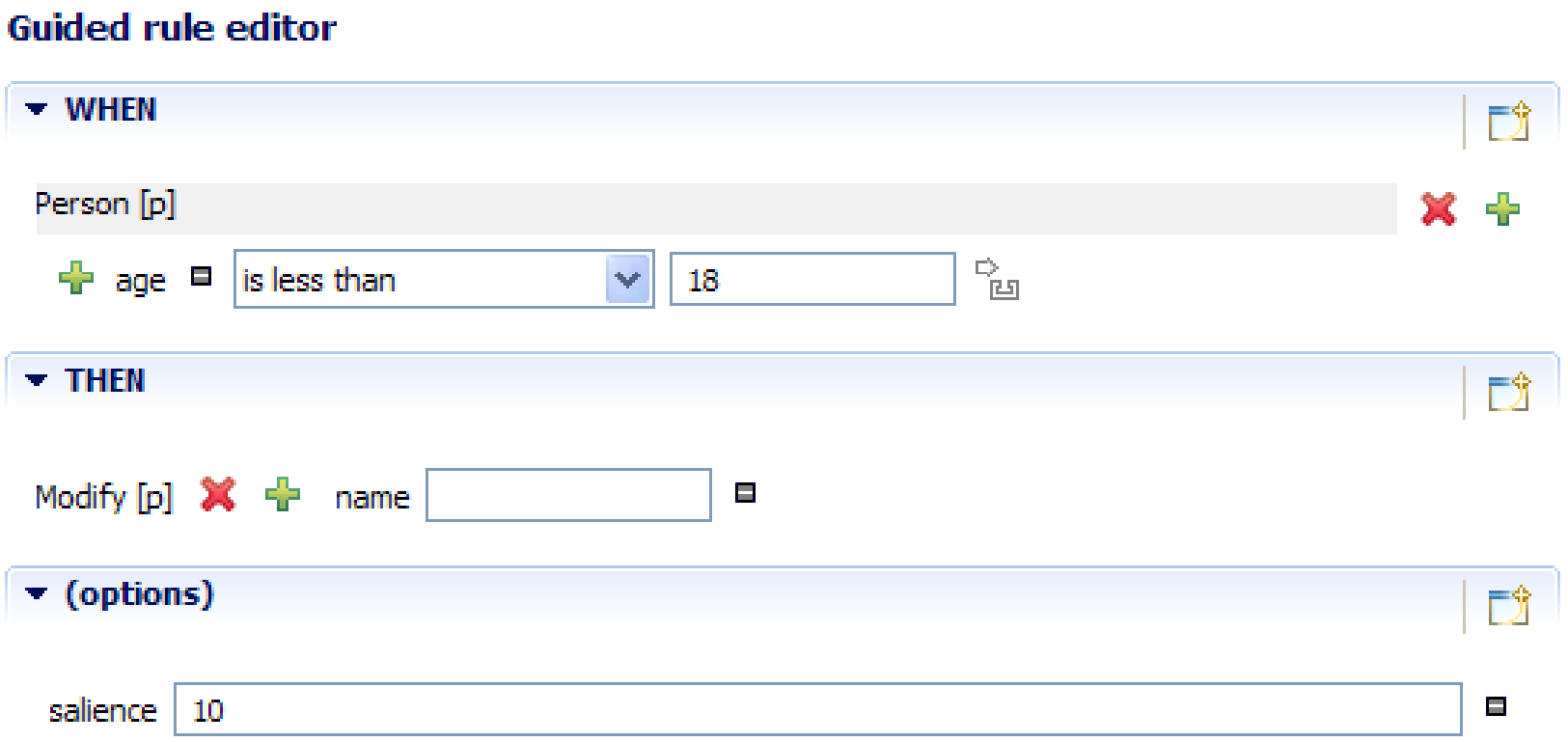

Drools - Hello/src/main/rules/Sample.drl - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Package Ex
Navigator
Hello
src/main/java
com.sample
DroolsTest.java
src/main/rules
Sample.drl
JRE System Library [jdk1.6.0_02]
Drools Library
src
Rules View
com.sample
GoodBye
Hello World
Sample.drl
DroolsTest.java
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(
19 then
20 System.out.println(message);
21 m.setMessage(message);
22
23 end
class
hashCode
message
status
this
toString
Text Editor
Rete Tree
Outline
com.sample
GoodBye
Hello World
com.sample.DroolsTe
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)
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$Message@1a19458(1)
Activation executed: Rule GoodBye message=Goodbye cruel world(1); m=com.sample.DroolsTest$Message@1a19458(1)
Writable
Insert
7 : 22

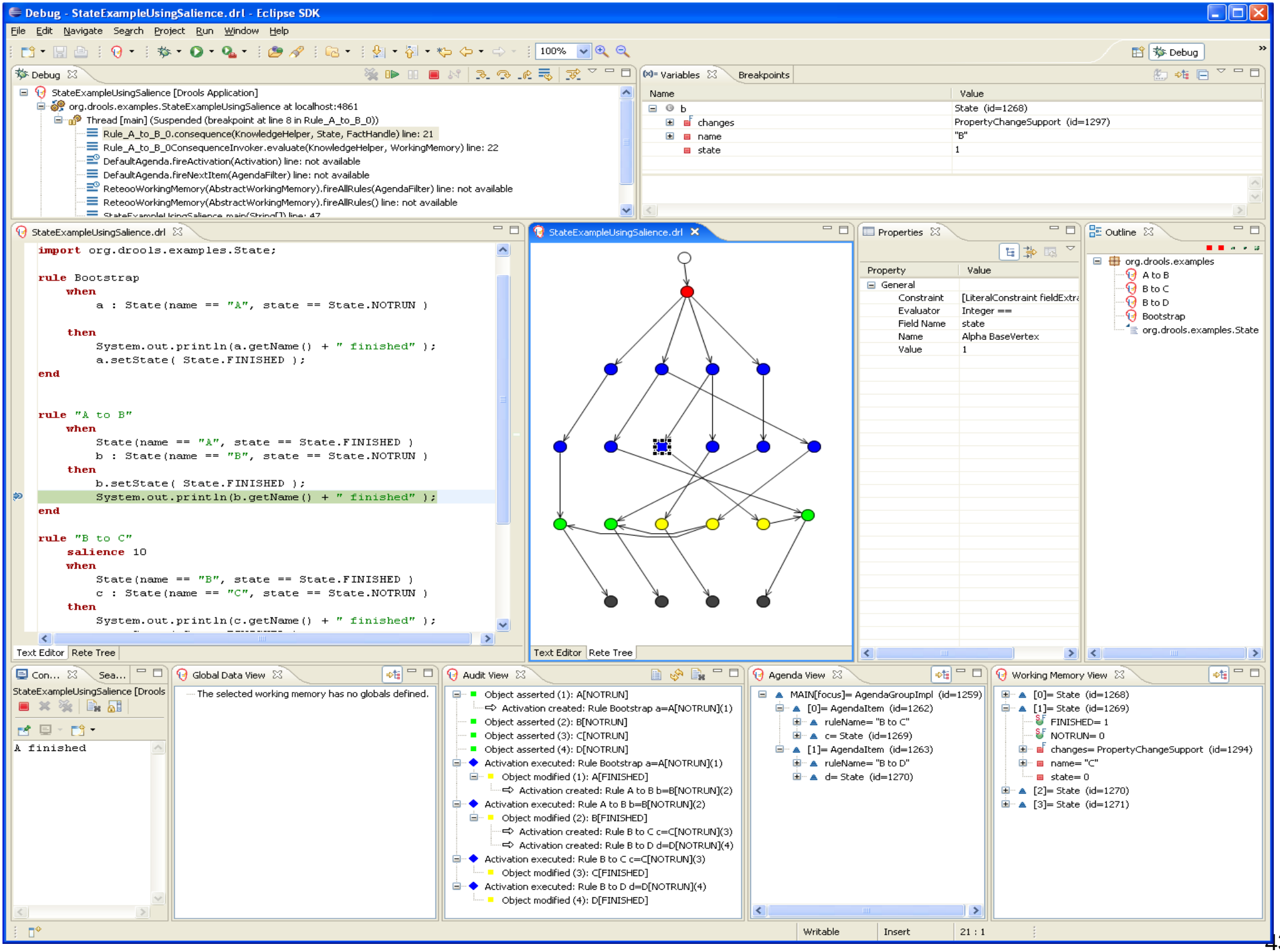
Domain-specific Rule Editor

```
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
12
13
14 end
15
```

- with age between {x} and {y}
- with name "{name}"
- Instance is at least {number} and field is "{value}"
- There is a Notification of type "{type}"
- There is a Person
- There is an Instance with field of "{value}"
- There is no current Instance with field : "{value}"

Guided Editor




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
StateExampleUsingSalience.drl
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

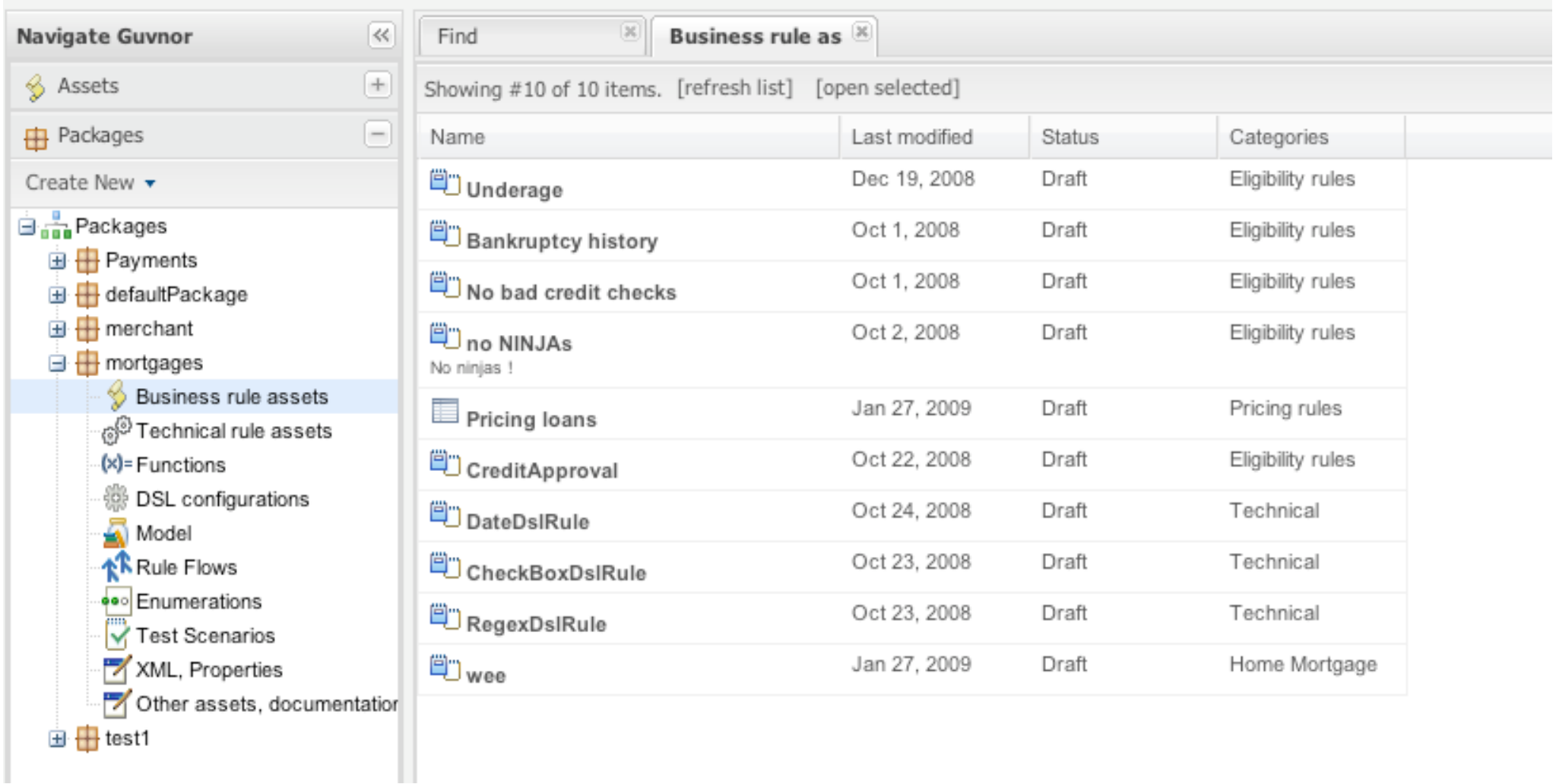
Drools Guvnor

Knowledge Repository

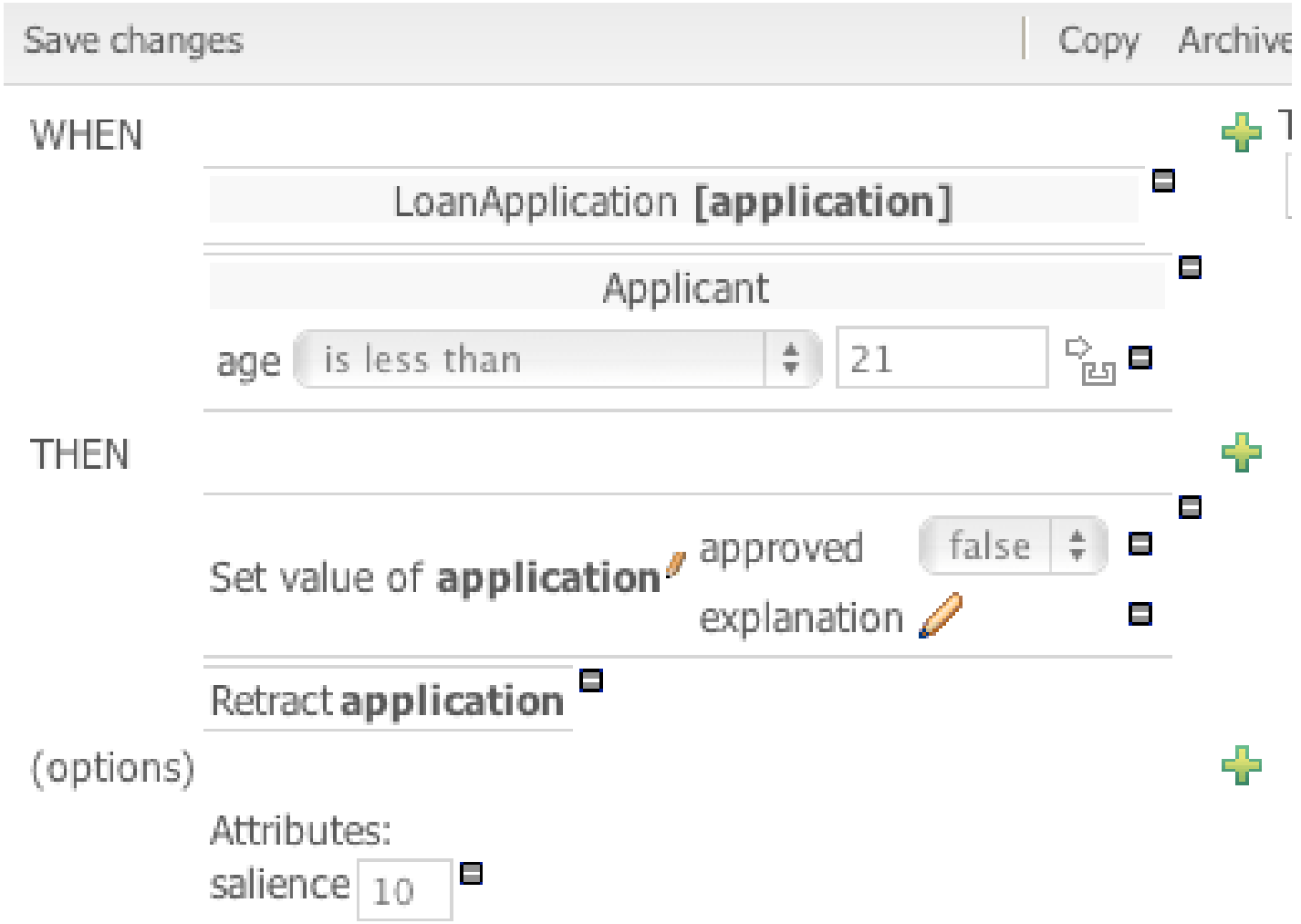
Guvnor

- Technology
  - JCR (JSR-170) backend
  - Seam + GWT frontend
  - WebDav
  - Eclipse synchronisation plugin
  - Role based security
- Authoring
  - Decision Tables
  - Guided Editor
- QA
  - Scenario Testing + Rule Verification

Business asset storage



Guided Editor



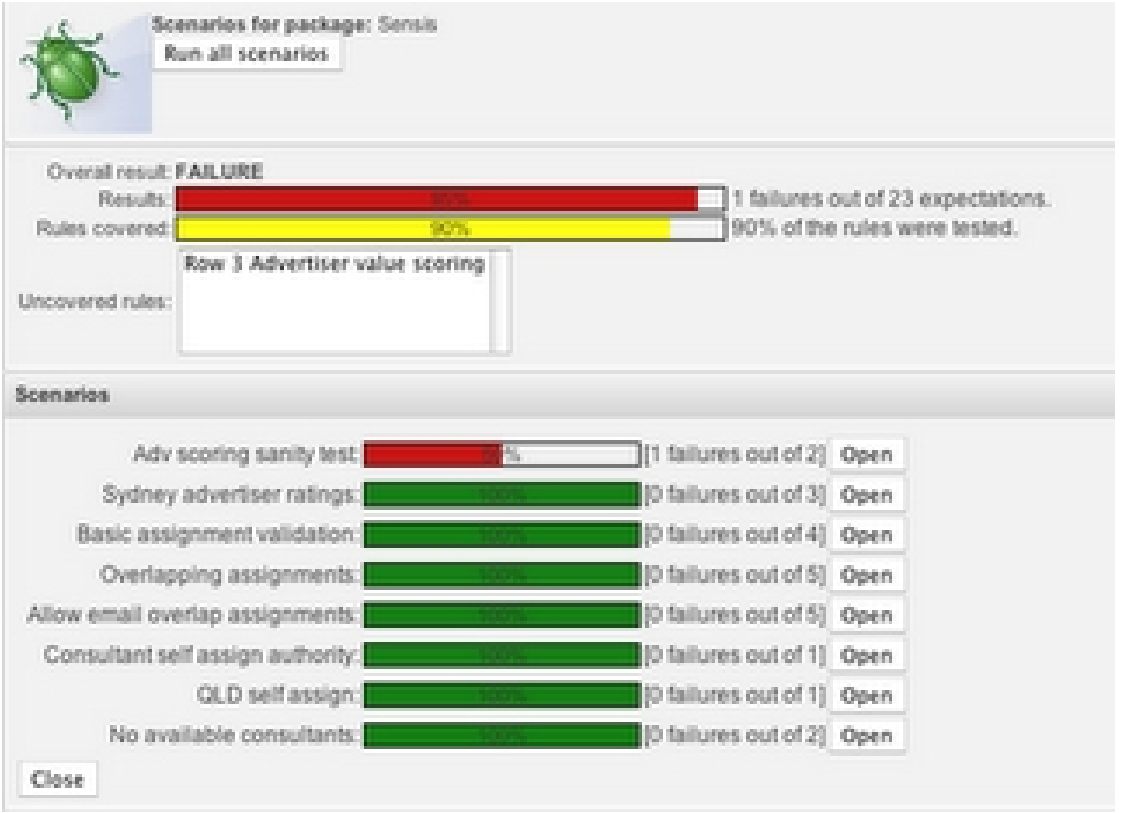
Decision Tables

Decision table

Modify...

	Description	Advertiser type	age is at least	Postcode gre...	Postcode les...		Set the value...	Set the rea...
Advertiser type: Agency (3 Items)								
1	Good suburbs	Agency	10	4000	4100	→	42	Loyal
2	Good suburbs	Agency		2000	2100	→	43	Good region
4	Good suburbs	Agency		2200	2300	→	43	Good region
Advertiser type: Partner (1 Item)								
3	Partners	Partner	1			→	49	Other

Scenario Testing



Drools Planner

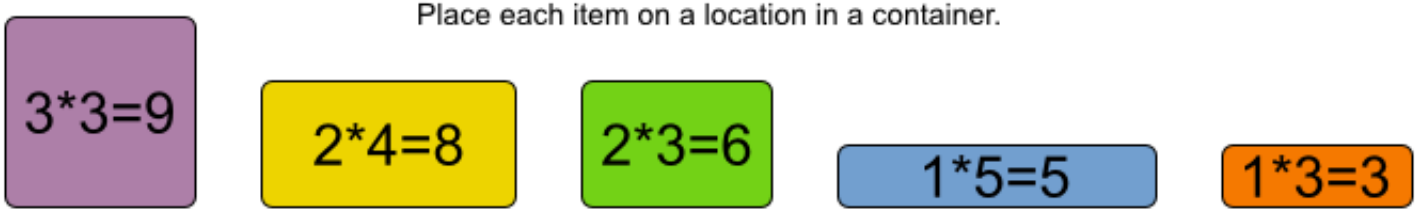
Automated planning

Drools Planner agenda

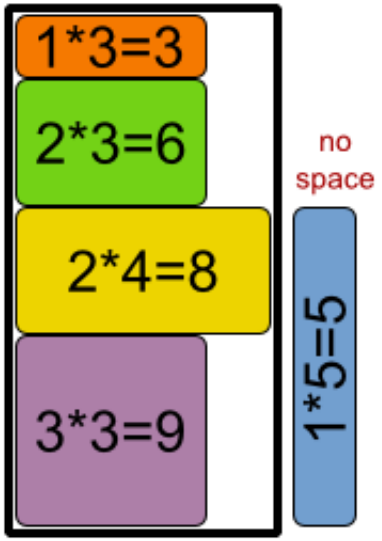
- Bin packaging
  - What is NP complete?
- Employee shift rostering
  - *AKA Nurse rostering*
  - Hard constraints
  - Soft constraints
- Patient admission schedule
  - *AKA Hospital bed planning*
  - How many possible solutions?
- Other use cases

Bin packaging

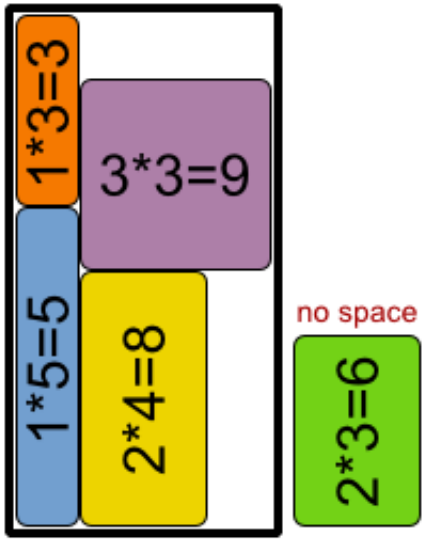
Place each item on a location in a container.



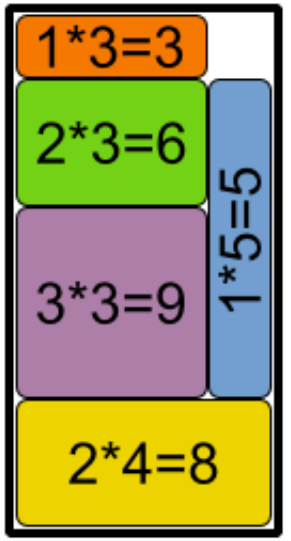
Largest size first



Largest side first



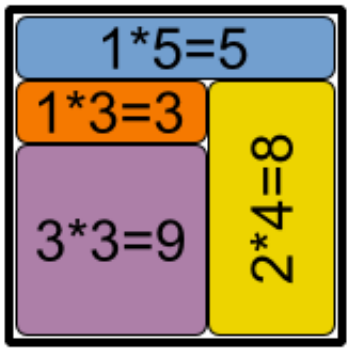
Drools Planner



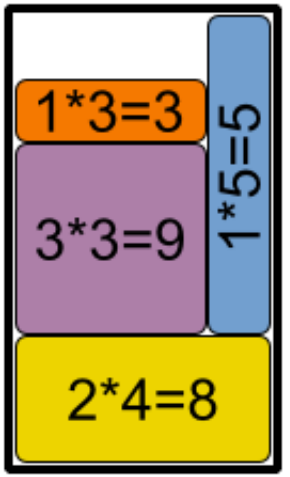
Bin packaging is NP complete

When do we put 2*4=8 into the container?

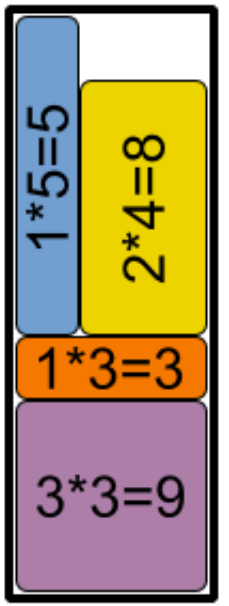
Second, of course!



First, of course!

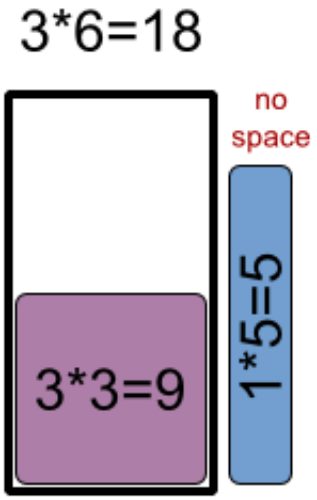


Last, of course!



A given solution can be verified fast.
There is no efficient way to find a solution

Bin packaging is NP complete



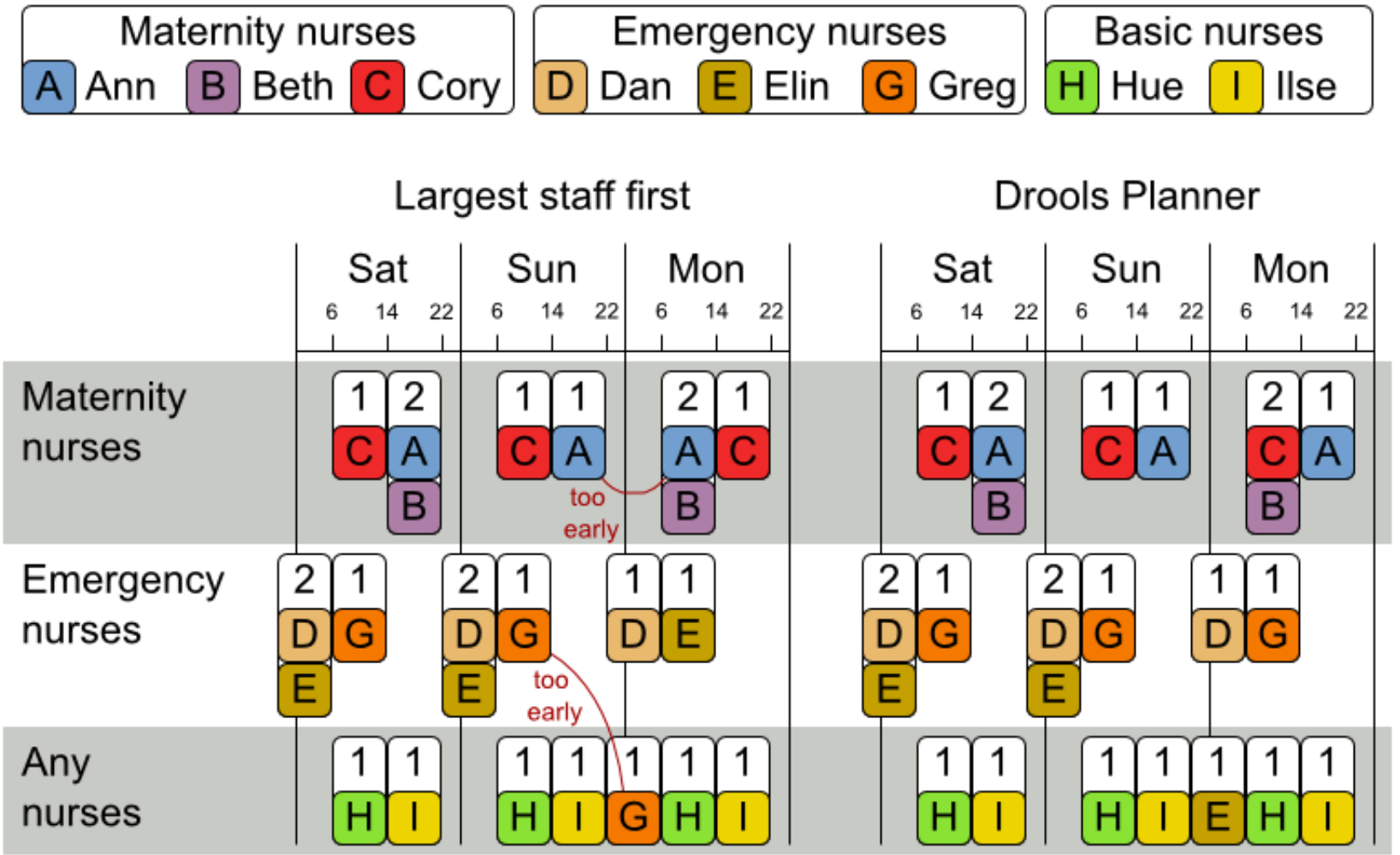
This container of size 18 can not hold these 2 items with a total size of 14.

There is no easy way to verify if there is even a feasible solution.

Drools

Employee shift rostering

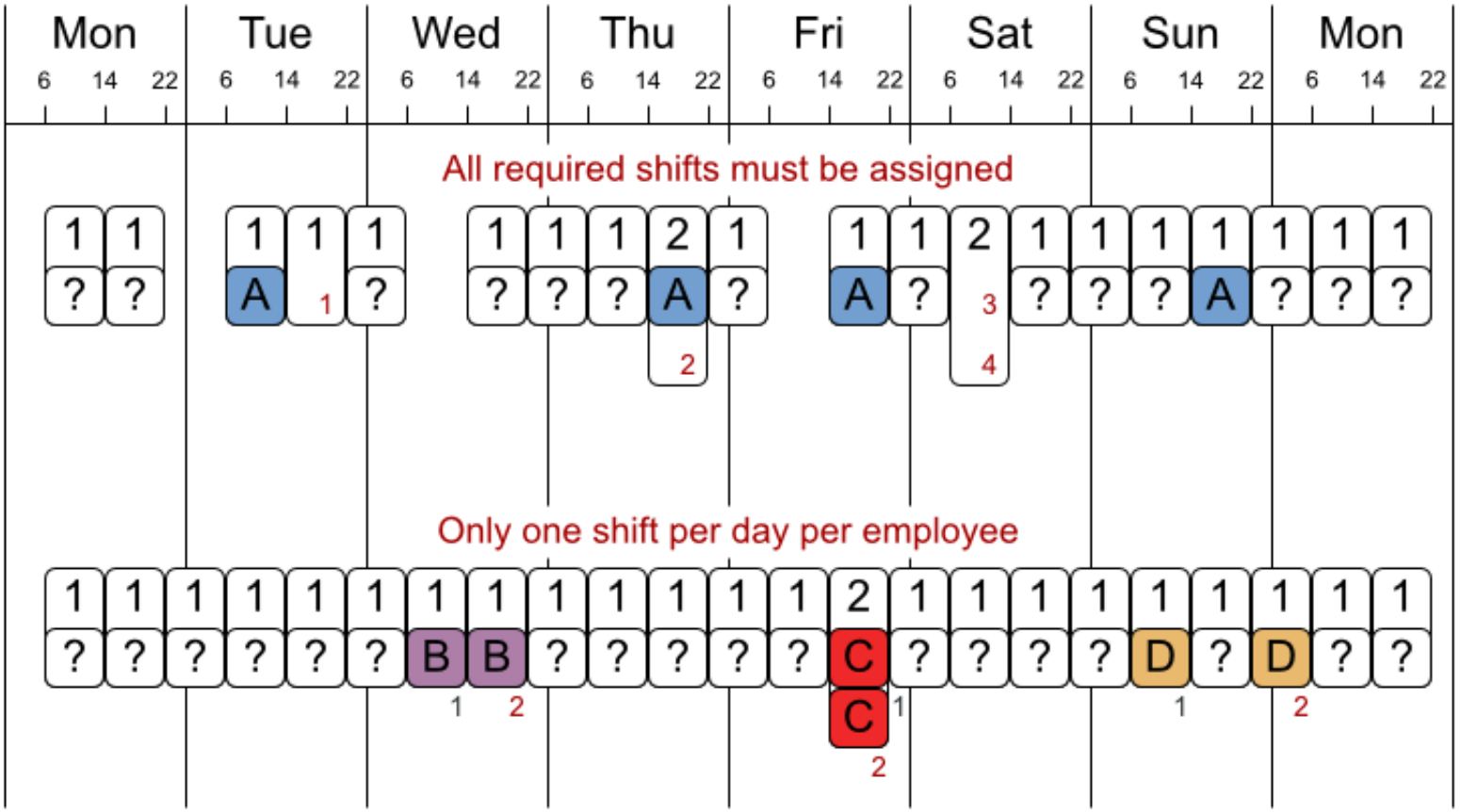
Populate each work shift with a nurse.



Drools

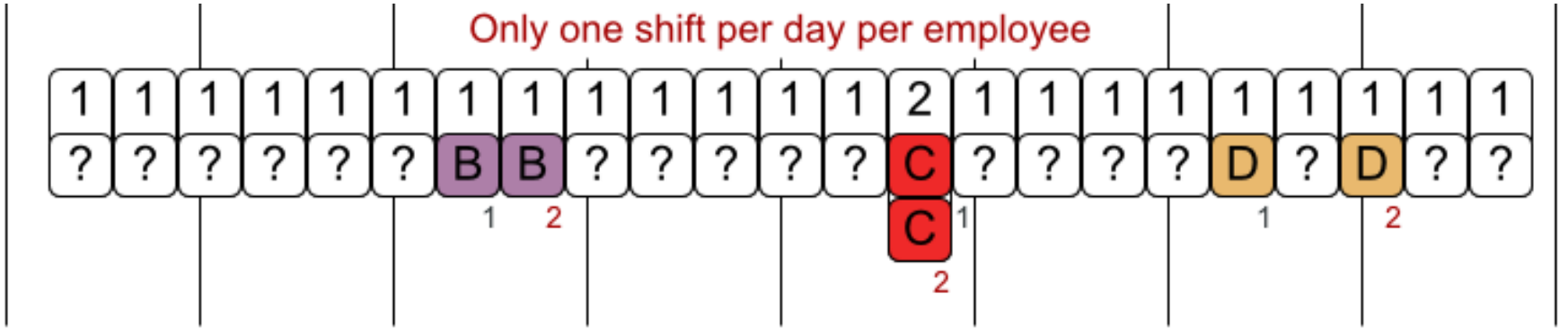
Employee shift rostering

Hard constraints



No hard constraint broken => solution is feasible

Hard constraint implementation

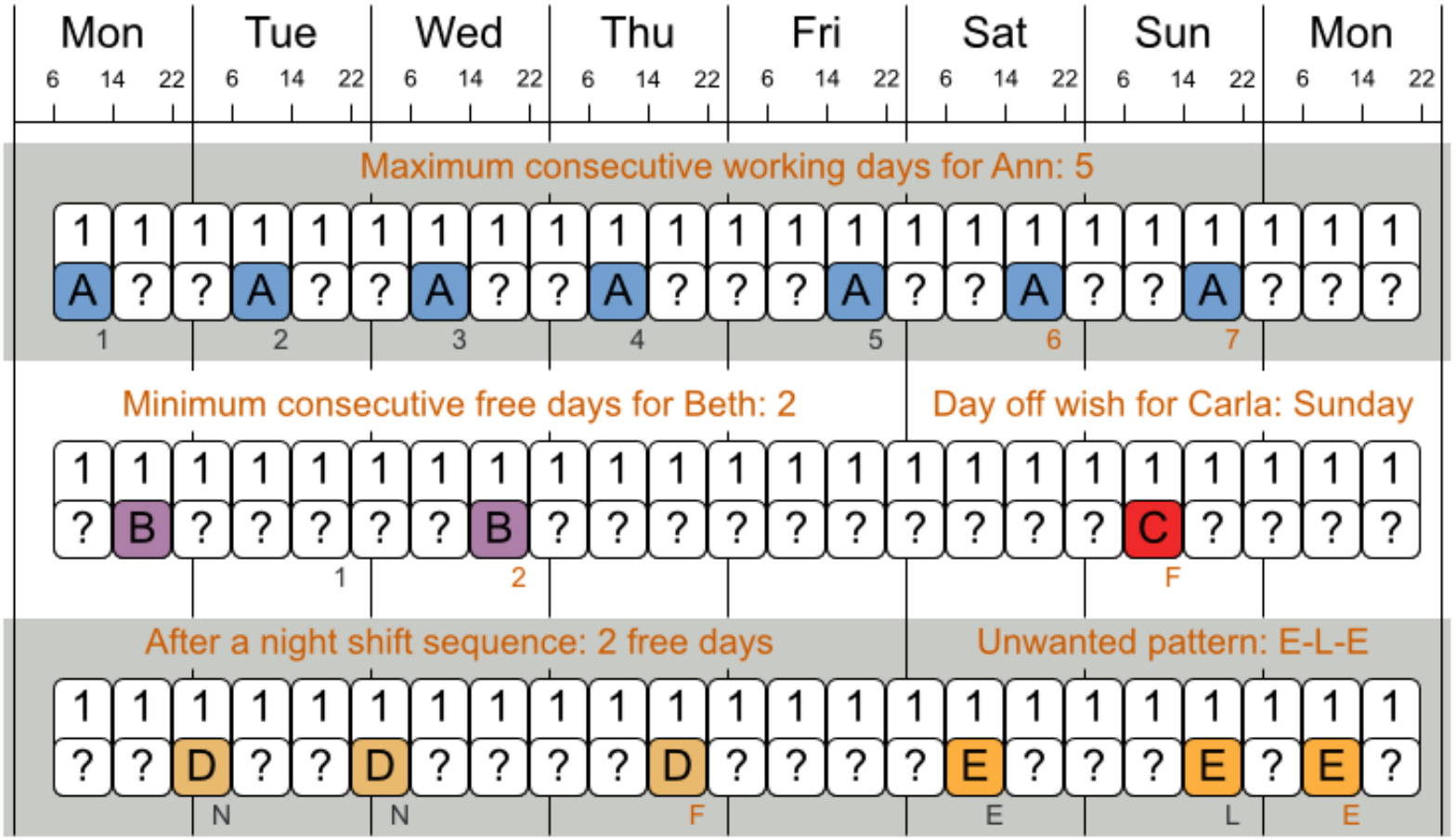



```
// a nurse can only work one shift per day
rule "oneShiftPerDay"
  when
    $left : EmployeeAssignment(
        $employee : employee,
        $shiftDate : shiftDate,
        $leftId : id
    );
    $right : EmployeeAssignment(
        employee == $employee,
        shiftDate == $shiftDate,
        id > $leftId);
  then
    // Lower the hard score with a weight ...
end
```

Drools

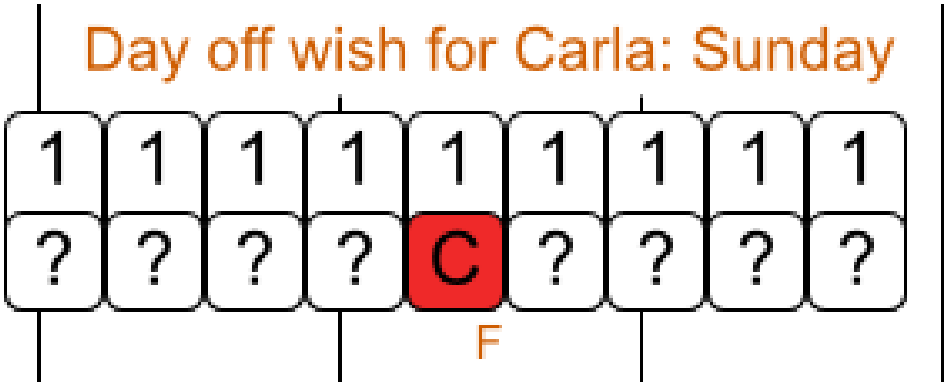
Employee shift rostering

Soft constraints



There are many more soft constraints...

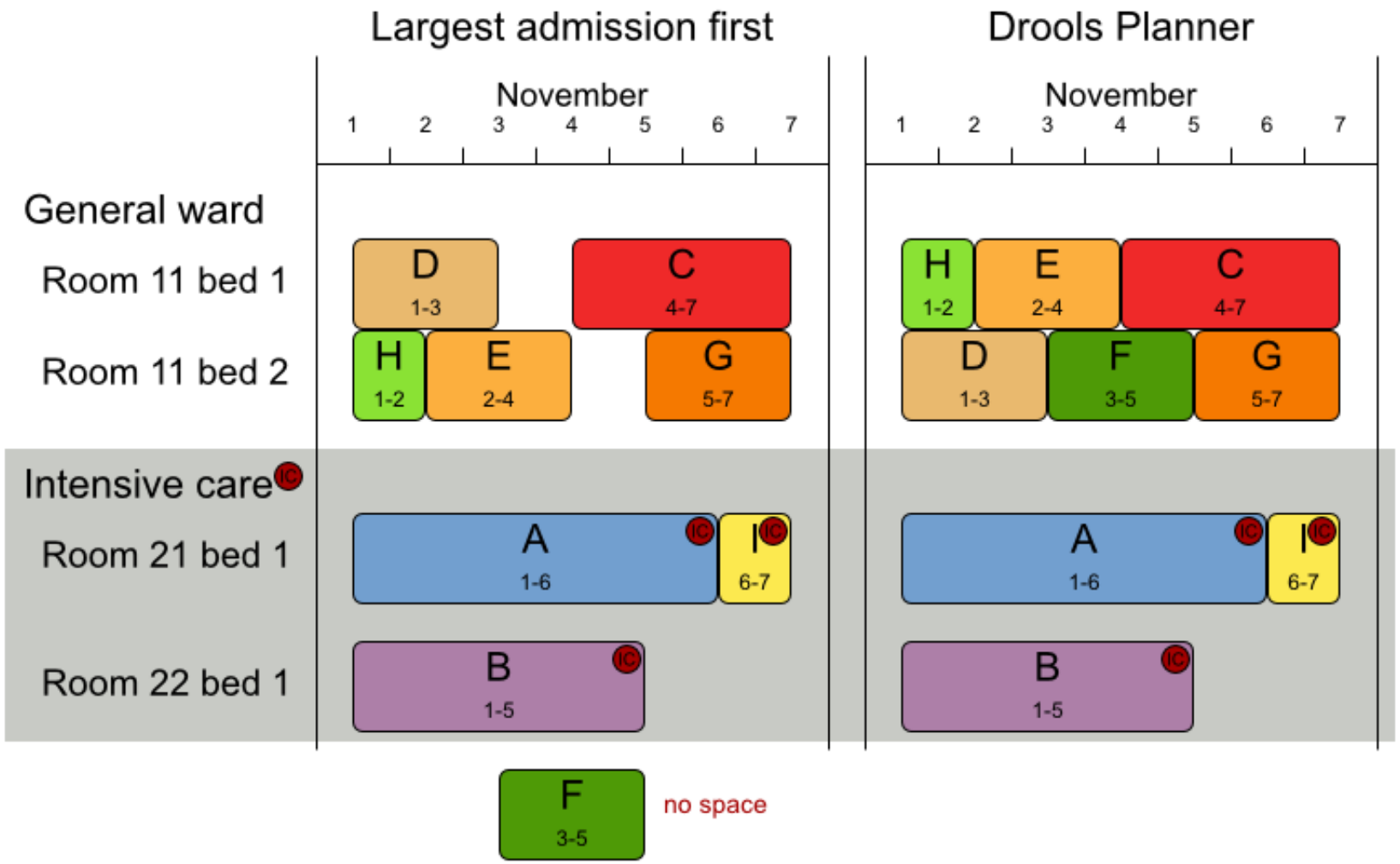
Soft constraint implementation




```
rule "dayOffRequest"
  when
    $dayOffRequest : DayOffRequest(
        $employee : employee,
        $shiftDate : shiftDate,
        $weight : weight
    );
    $employeeAssignment : EmployeeAssignment(
        employee == $employee,
        shiftDate == $shiftDate
    );
    then
      // Lower the soft score with the weight $weight ...
end
```

Patient admission schedule

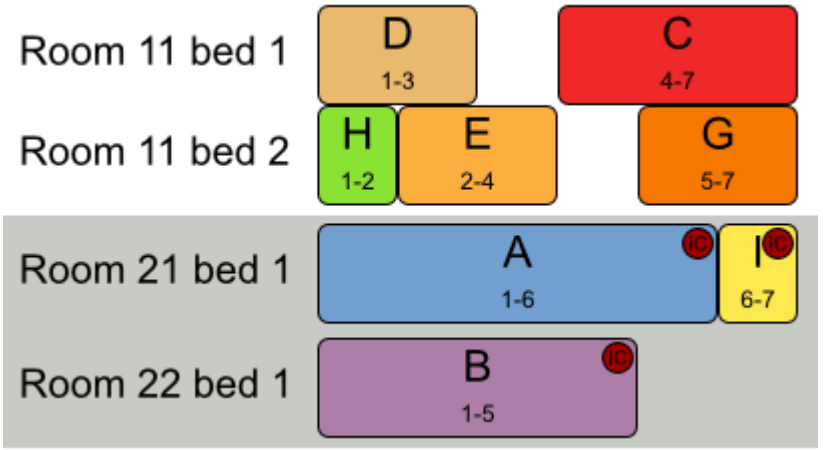
Assign each patient a hospital bed.



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)

Needle in a haystack

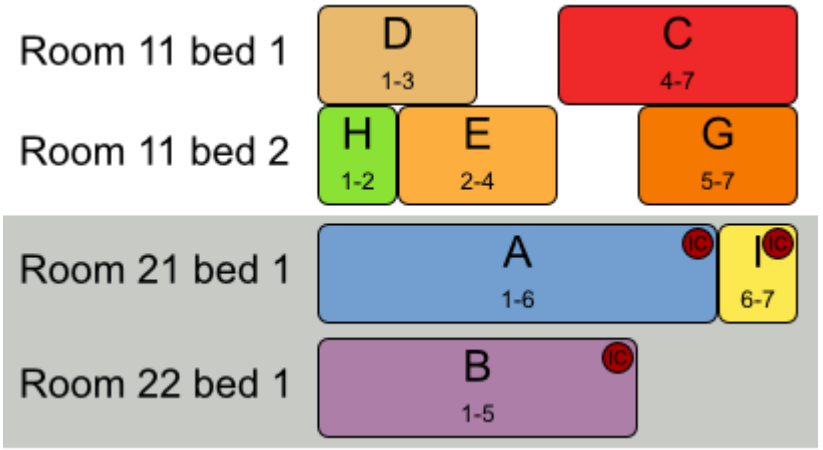


How many possible solutions?

- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

Numbers from a real dataset

Needle in a haystack



How many possible solutions?

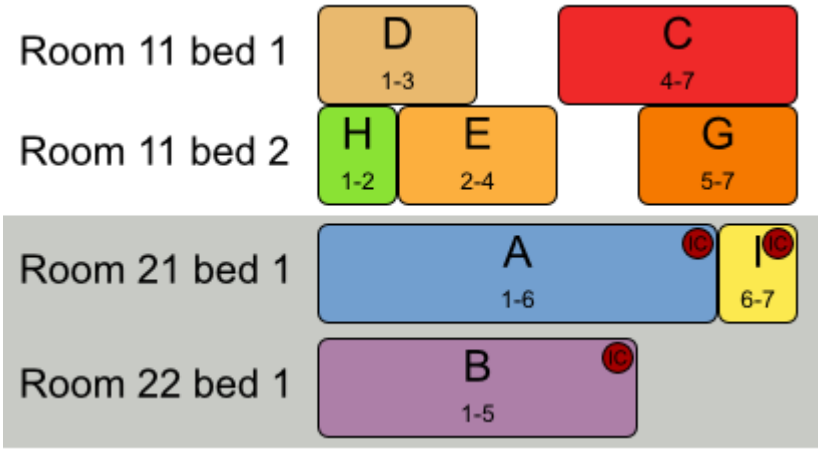
- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

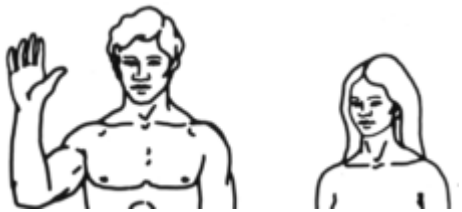
> inhabitants of Gent (BE)?

- 250 000 inhabitants

Source: wikipedia

Needle in a haystack





Source: NASA (wikipedia)

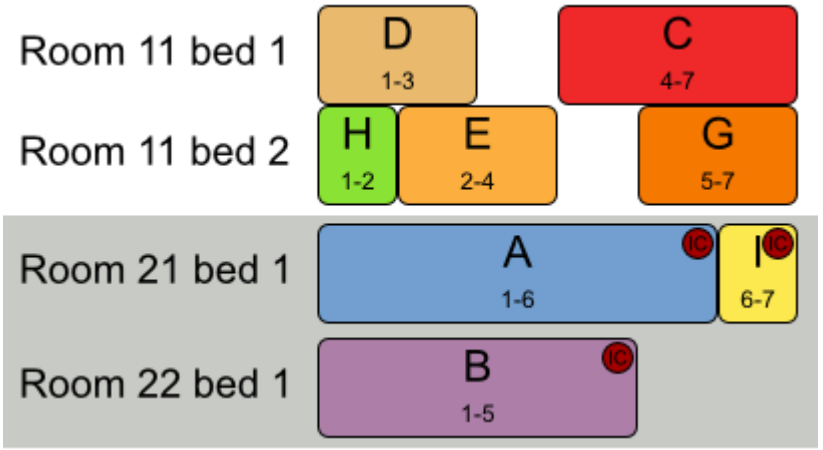
How many possible solutions?

- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

> humans?

- 7 000 000 000 humans

Needle in a haystack

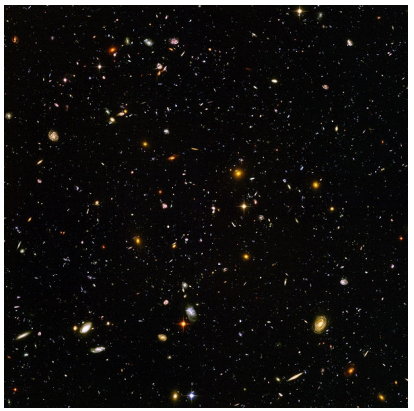


How many possible solutions?

- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

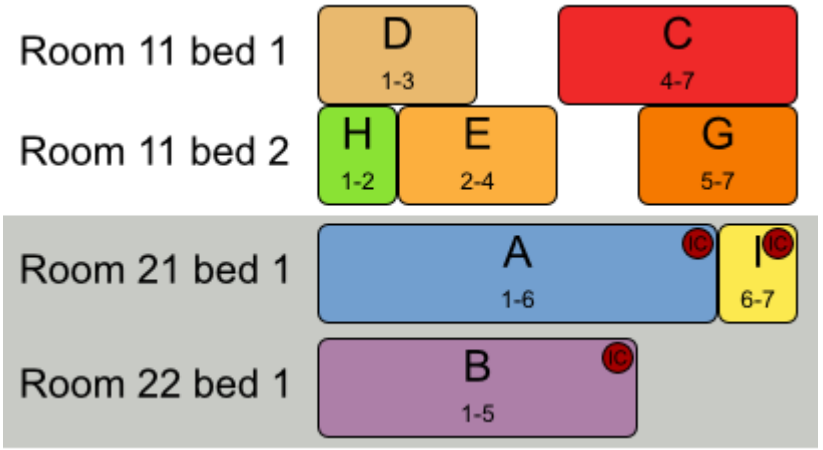
> minimum atoms in the observable universe?

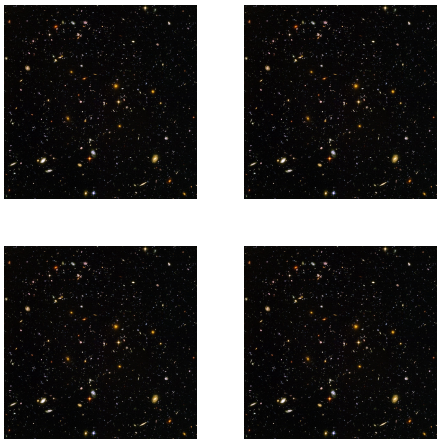
- 10^80



Source: NASA and ESA (wikipedia)

Needle in a haystack





Source: NASA and ESA (wikipedia)

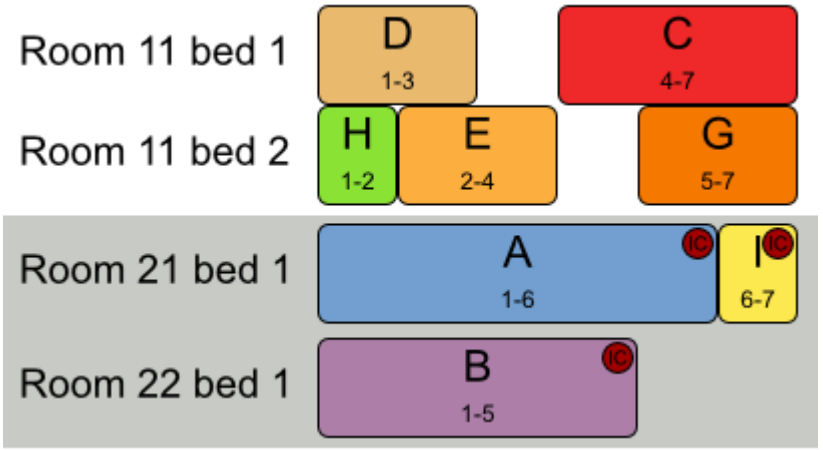
How many possible solutions?

- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

> atoms in the universe
if every atom is a universe
of atoms?

(10^80)^80 = 10^6400

Needle in a haystack

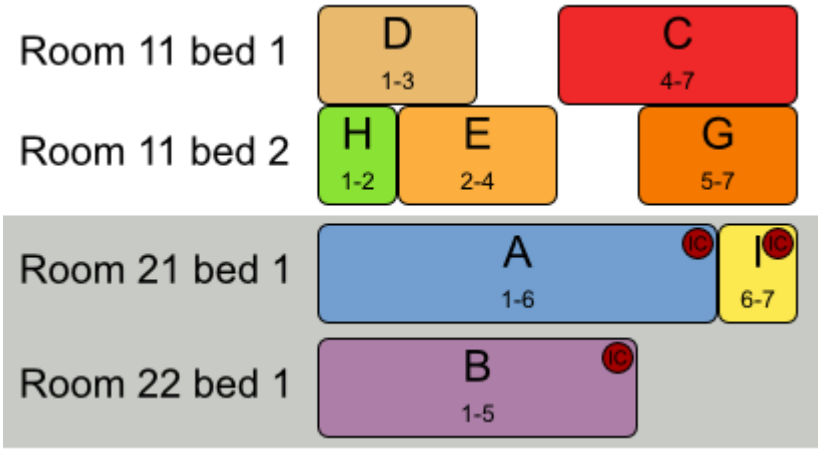


How many possible solutions?

- 310 beds
  - in 105 rooms
  - in 4 departments
- 84 nights
- 2750 patients (admissions)

A little over 10^6851

Do the math



1 patient

310 beds

310 ways to schedule 1 patient

2 patients

310 * 310 = 96 100

3 patients

310 * 310 * 310 = 29 791 000

2750 patients

310 * 310 * ... * 310

310^2750

= a little over 10^6851

A little over 10^6851

```
17565400009613647714189309909847019528176031676612802408947467896773536824694320
25377105894429532838896584732142182052033826938421876324335318655320329177415624
34019697177044291997231955740197313574392215860109009013187756616189881491857442
89903563474940721372704649669447515706252843900344472960295273324597346780837674
88697227533874455473777198677633776549677388281039595554235192833141437843928340
51328923587442242154997922875864744269141114570273352582691671031946927906652177
66902628732489519488236496931246157419949856522543510386164584826962224171420152
25565850232385350464349705325292272792440640617915932702527496869105242581827012
17106705526418375034351087944819610624220027292887379804041858752339124281780095
58605680683578864680145557998942327371019832139824665975180911386722774004539981
34278552385168663637443267160148549642311093017595330858041676796683206809536596
46569395830894437089458443238882278162963824641222099807169369905394994720275907
38400791514217875461942733015467480308665074008529461146577114465977072581447925
88212290682716057000722801705649967418814850790871678616492253646749058716362694
56894529270615321506374546152336416456127910274106084164763806424690851439804640
67453971429400369608313929289399595696359958354101721624055729520838609453039985
59272628937624385694142906376790391997138724432513602703448146045970566585076 80
95764769840369232215532708782795742398666571567290512950859701002128560257873450
34666683590797377984104207041334053480226788236770435009014979334517769826461063
28117889455452701285996652060625309878869936718080636701237289582496335799276496
97991233610444564616874109815224930933169104649370892996558804470140748763978122
10684054337706530175130912335567383560436635091489083375525519539844805718811296
85807650768511219249940528633766810704609502399987122465510787717422067936021241
05730014911043812216621387647568988345883813404108921585448372002290085339308167
94663631470201197595487045022087615873490295940409113638894083753062801416644858
```

A little over 10^6851

70757942487218045035508810158461046502812782292385633846174494690914238485407798
37976573852840248463244968564240141089763763217269495446550923090738150172870668
68402081731644263484686141346509306107283418762923467113106773326485539514229919
89751468506508069513397904821612697383659788320905691994864296149528670873271380
18006650770249052559419638332728972636228024127885657959189574420249964658137384
98299648678707389648424263725804209284739296524664530660893065815727451390562561
81505240205578741292314133858985615181677968313683876880079649763914095651402272
04777610845144506688836696844897279181666903094579081039689572524839918882203466
75664835223276850061950801785251074912552450542389767056205475823297598574505575
83834141364747838395082871684183199560673441708538602779816347949276957201268066
62737283137076807355893467941027682428304918329951886951690865417997171855081020
26756284570976172802328890960381286165431282000890478132235199027141966946398442
73986565523012106816309974964021700560537928432528631741787455155275823033005847
63710721074772137206636934675415209083984961138990816735390734923436827652228741
07306375553429574524282669680025732278499336914490634182865013110363140489605282
49465982665132492491072491788618403253474529440348798670244615185355092357283764
93638760707623418191667075526942154653033728468983877312232305317179427144435853
36388068489698718841685828476130163980130066330161405037431756112554842734192914
35502775849577615159921009571496639402549077872124227731739936370100132762333353
47390808021571900433548715701062002052376364885669272869945947160341077659253581
65207459958613365778774312937767961242646970494187951860105460569752264242274554
46011031581123438684685838655333776830823720356749120051129394691394743488410658
61354353779545576065515784960221442891508405618131446589507592449826384604047449
56226545521579338675725427458383708893912437023584592443865610814799055455700844
91443709439642235090455604548030317754849613813010298858282615659336373785985294

A little over 10^6851

```
58667337266411706925088329158776619790296442059252886388879455197093987039774900
08729398961010317200010000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
```

A little over 10^6851

```
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000
```

- The search space is big!
  - Compare with WWW size
    - 22 020 000 000 pages
- Each possible solution
  - 2750 patients scheduled into 310 beds
  - Still need to calculate the score!

Throw hardware at it?

Calculate 10^9 scores per ms

- Impossible today!
- 31 579 200 000 ms in 1 year
  - < 10^11 ms in 1 year
- 10^9 * 10^11 scores per year
  - = 10^20 scores per year

How many years? 10^6851 / 10^20

- = 10^6831 years

CPU 1000 times faster

- It becomes 10^6828 years

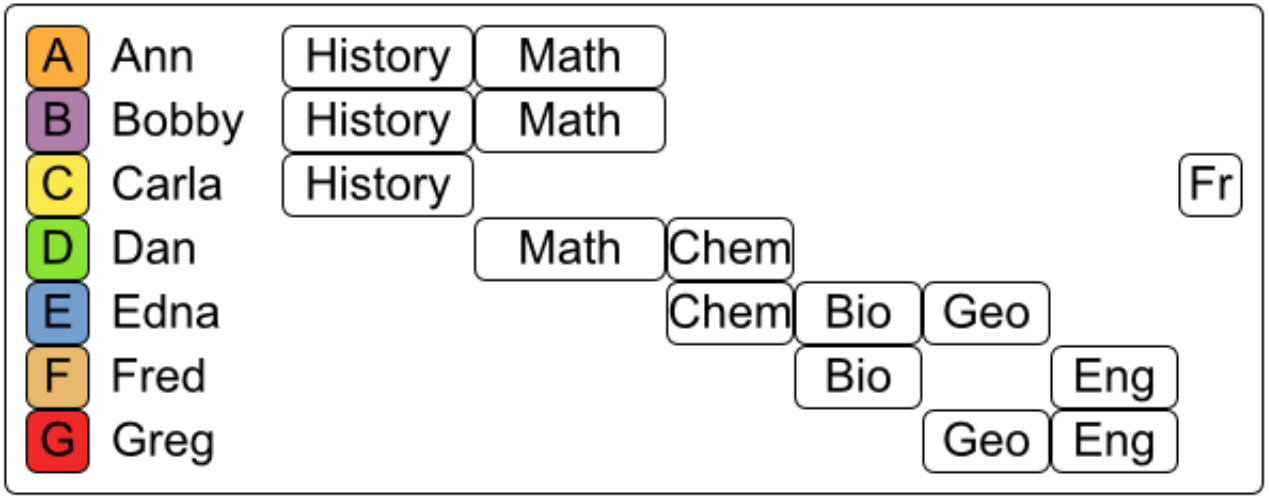
A dose of reality

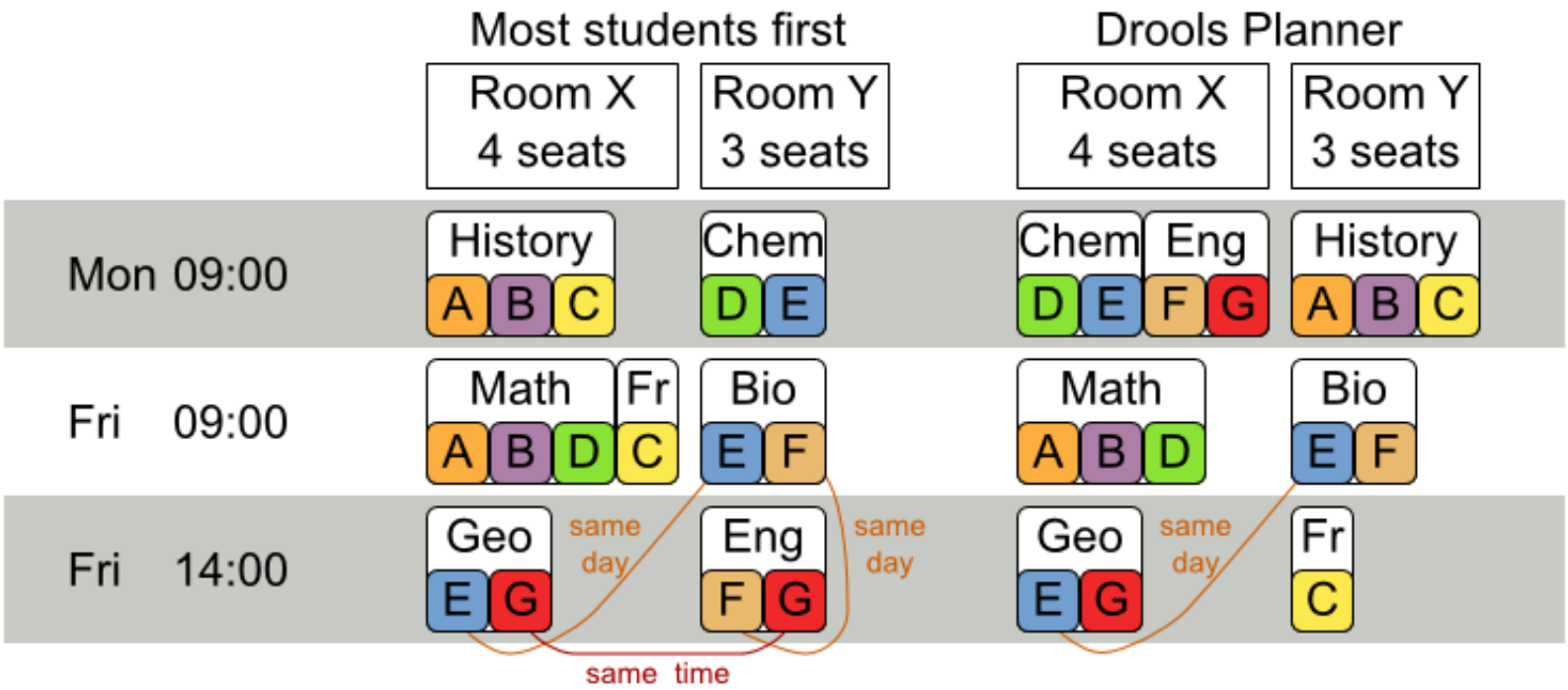
- Find the optimal solution?
  - Of a real world planning problem?
- Not in our lifetimes!
- Who cares?
  - Beat the human planner(s): easy!
    - Spend less resources
      - Save more money
      - Save the environment
    - Make more people happy
  - Always room to improve our algorithms

Drools

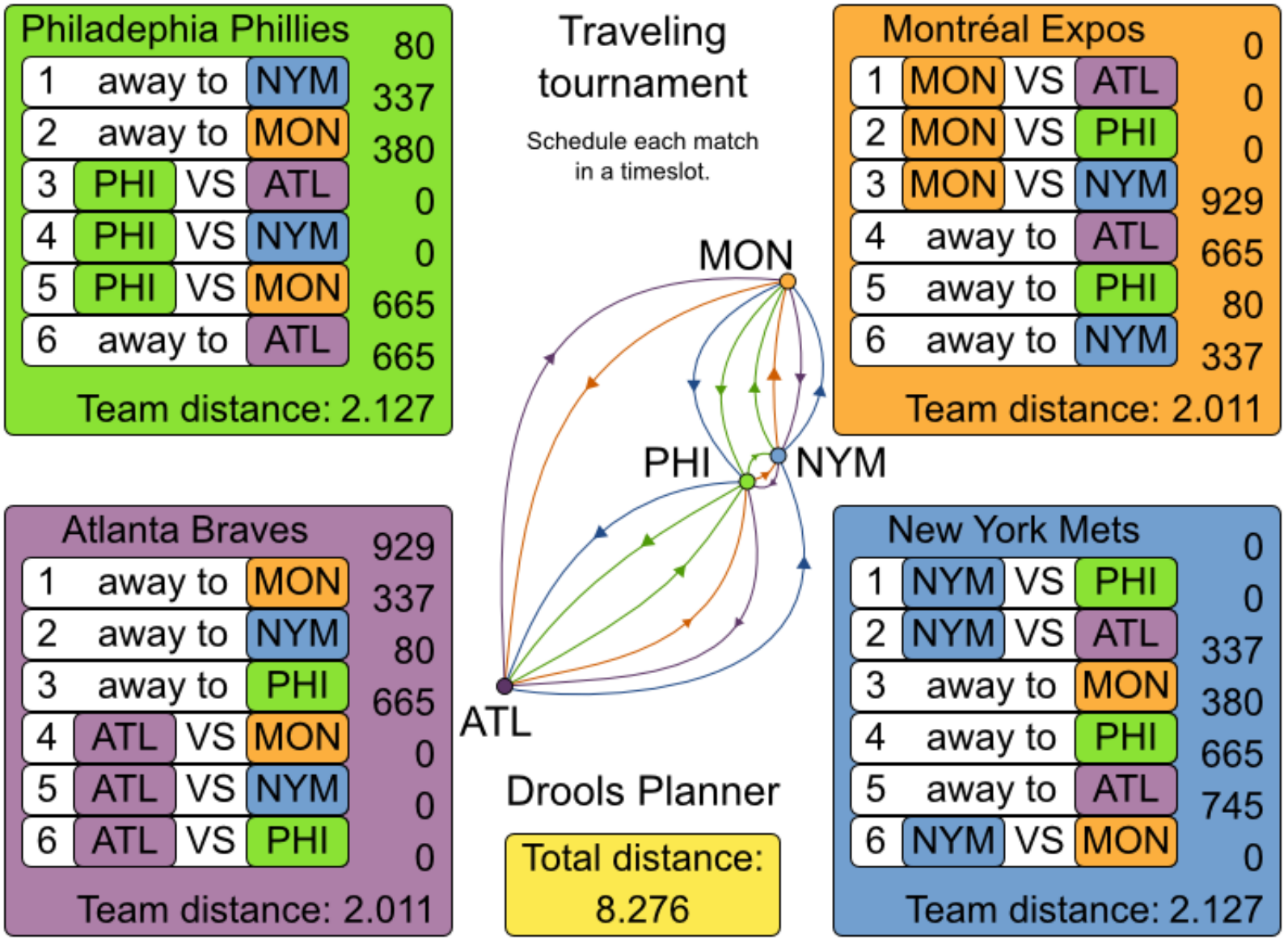
Examination timetabling

Assign each exam a period and a room.





Drools

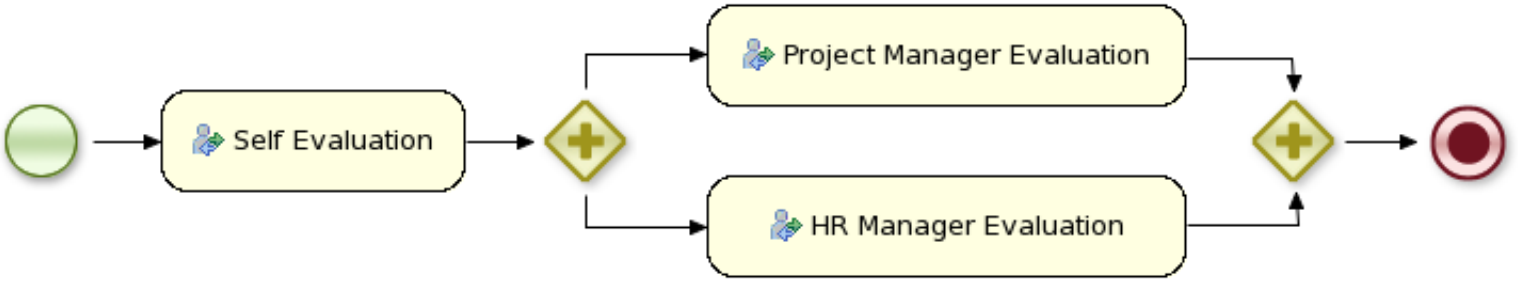

Philadephia Phillies
1 away to NYM 80
2 away to MON 337
3 PHI VS ATL 380
4 PHI VS NYM 0
5 PHI VS MON 0
6 away to ATL 665
Team distance: 2.127
Traveling tournament
Schedule each match in a timeslot.
Montréal Expos
1 MON VS ATL 0
2 MON VS PHI 0
3 MON VS NYM 0
4 away to ATL 929
5 away to PHI 665
6 away to NYM 80
Team distance: 2.011
MON
PHI
NYM
ATL
Atlanta Braves
1 away to MON 929
2 away to NYM 337
3 away to PHI 80
4 ATL VS MON 665
5 ATL VS NYM 0
6 ATL VS PHI 0
Team distance: 2.011
Drools Planner
Total distance: 8.276
New York Mets
1 NYM VS PHI 0
2 NYM VS ATL 0
3 away to MON 337
4 away to PHI 380
5 away to ATL 665
6 NYM VS MON 745
Team distance: 2.127

Drools Flow

Workflow engine

Drools Flow

A ***workflow*** is a process that describes the order in which a series of steps need to be executed, using a flow chart.



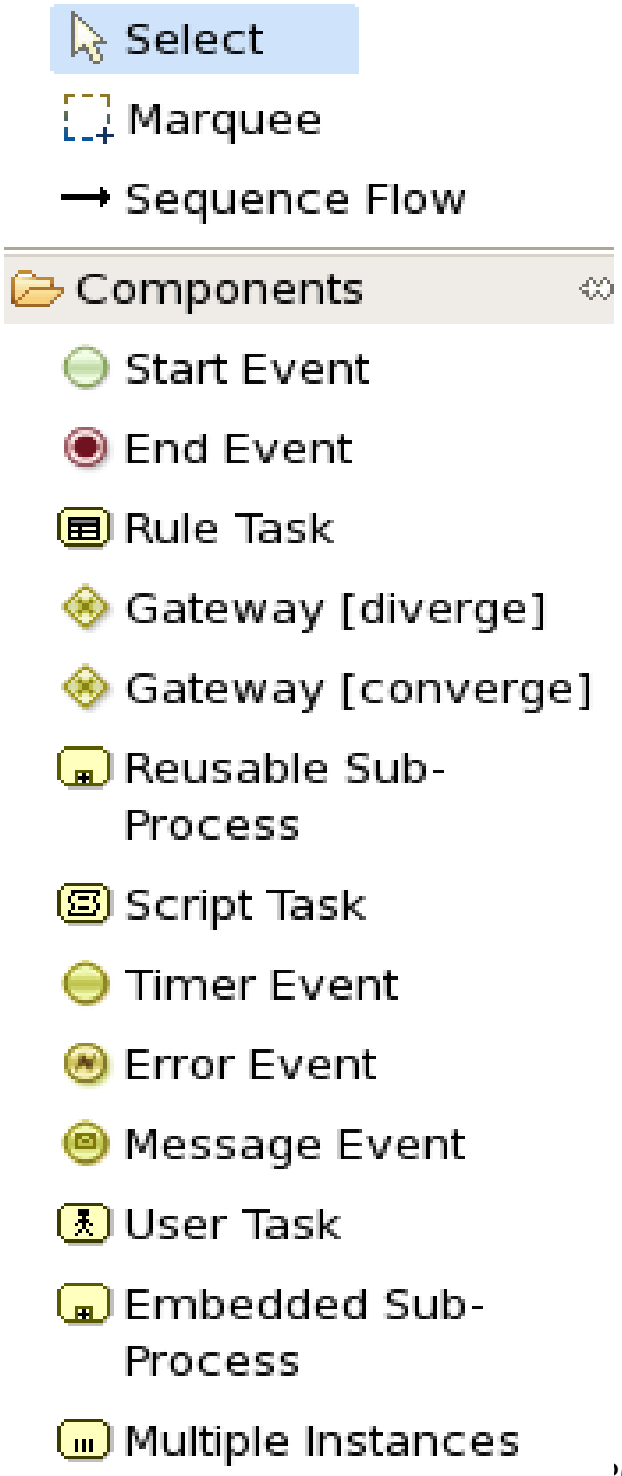
Advantages

- Transparency
- Automation
- Higher-level
- End-to-end process visibility and monitoring
- Increased agility (change process more easily)
- Lower cost of development
- Process analysis / continuous improvement

Drools Flow Palette

Predefined set of generic *node types*

- Start, end
- Gateway
- Script
- Sub-process
- Event
- ...



Characteristics

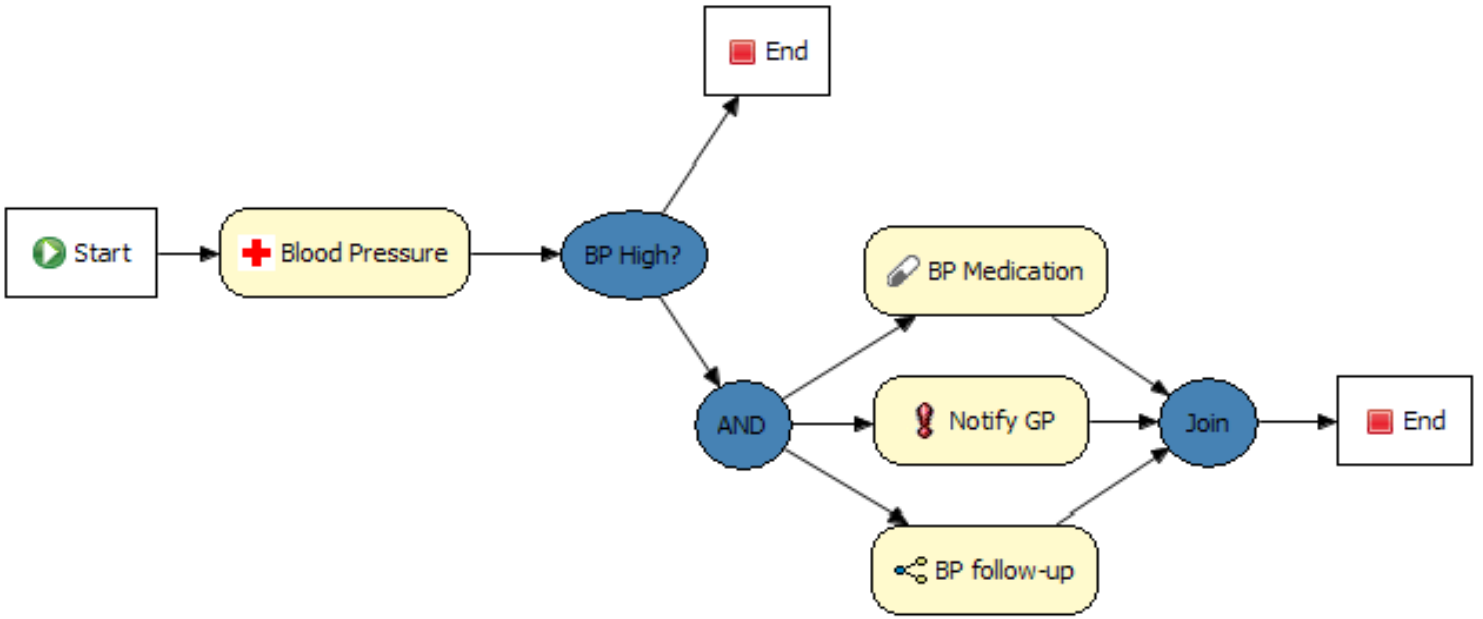
- Standard-based
  - BPMN 2.0 for process modeling
  - WS-HT for human tasks
- Advanced, declarative modeling
- Extensible, open-source process engine
  - Persistence, transactions, logging, etc.
- Domain-specific processes
- Integration and unification of processes, rules and event processing
- Full life cycle support

Code Example

```
// creating the knowledge base
KnowledgeBuilder builder
      = KnowledgeBuilderFactory.newKnowledgeBuilder();
  builder.add(ResourceFactory.newClassPathResource(
          "org/drools/examples/exampleProcess.bpmn"),
      ResourceType.BPMN);
KnowledgeBase kbase = builder.newKnowledgeBase();

// Starting a process
StatefulKnowledgeSession session
      = knowledgeBase.newStatefulKnowledgeSession();
session.startProcess("org.drools.examples.ExampleProcess");

// Signaling a process
session.signalEvent("myEvent", data);
```

Domain-specific Work Items

Extend palette with domain-specific, declarative work items



Processes vs. Rules

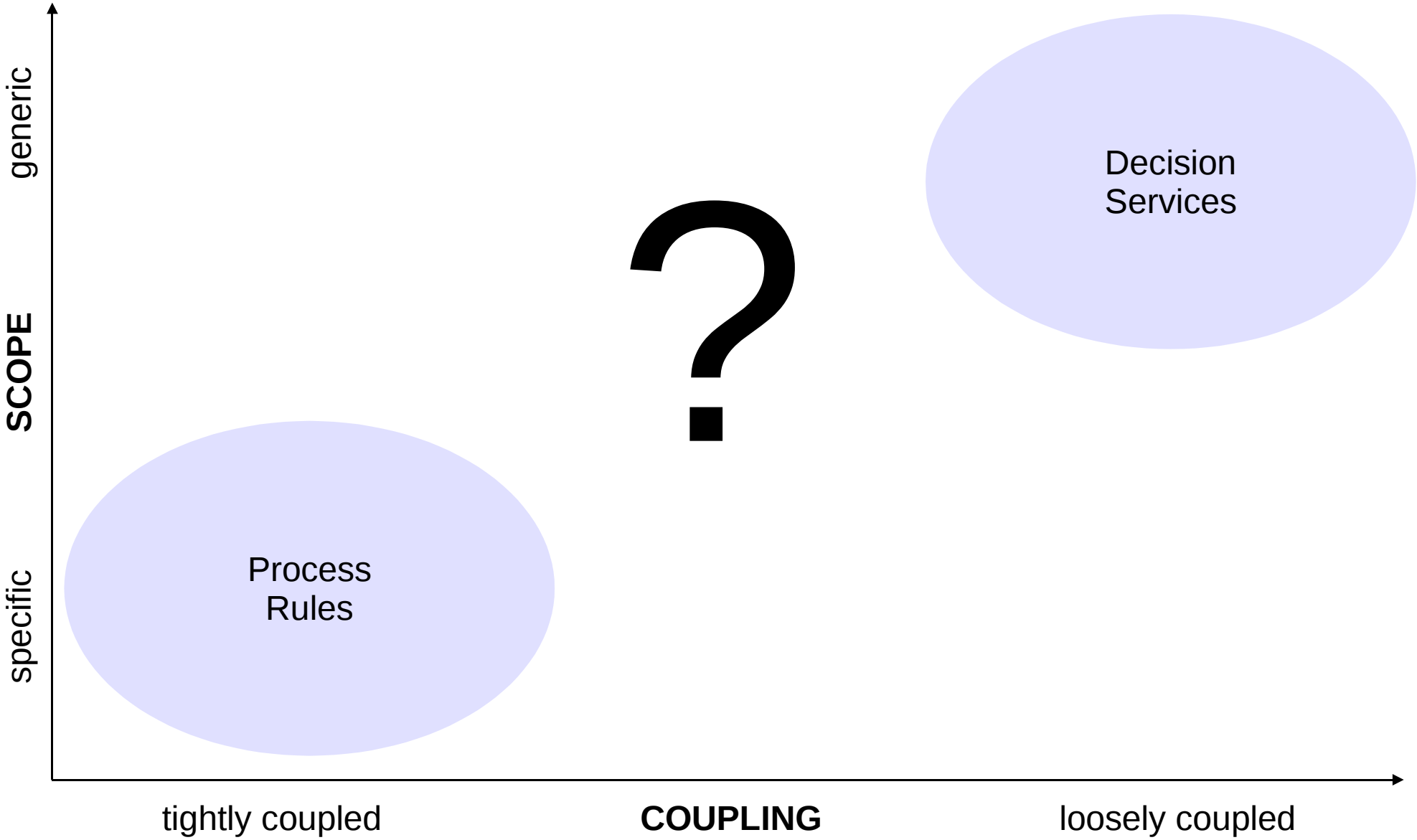
Process

- Focus on control flow
- Procedural
- Generic scope
- Long-living
- Stable
- Independent

Rules

- Focus on data
- Declarative
- Specific case
- Immediate
- Dynamic
- Additive

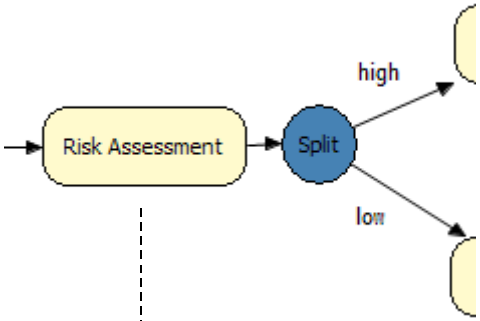
Rules and Processes



Decision Service

Extract decision logic from process definition using business rules

- Identify decision points
- Separate life cycle
- Permanent or temporary
- Other advantages
  - Reuse
  - Higher-level
  - Complexity



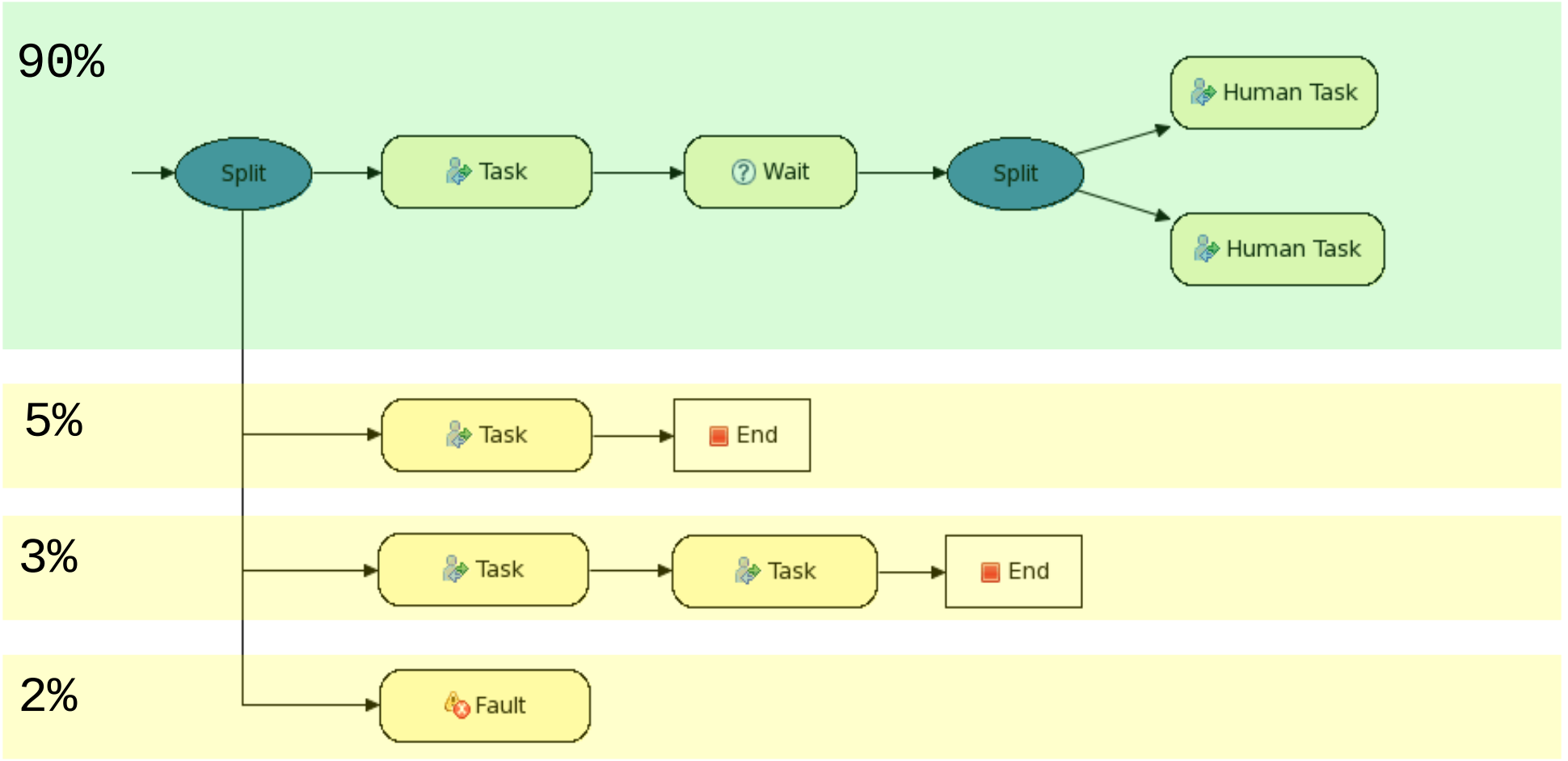

```
rule "High Risk if age < 21"
    ruleflow-group "RiskAssessment"
    when
        Person( age < 21 )
    then
        insert ( new RiskFactor(
            0.1, "Person is less than 21.") );
end
  end
    end
```

Process Rules

Using rules inside process ...

- for decisions (gateway, wait state)
- define
  - exceptional situations
  - escalation
  - cancellation
  - inclusion / exclusion criteria
  - assignment
- override default process logic
  - could be temporary
  - could be automatic (listen to own events)

Exceptional Control Flow



Agility

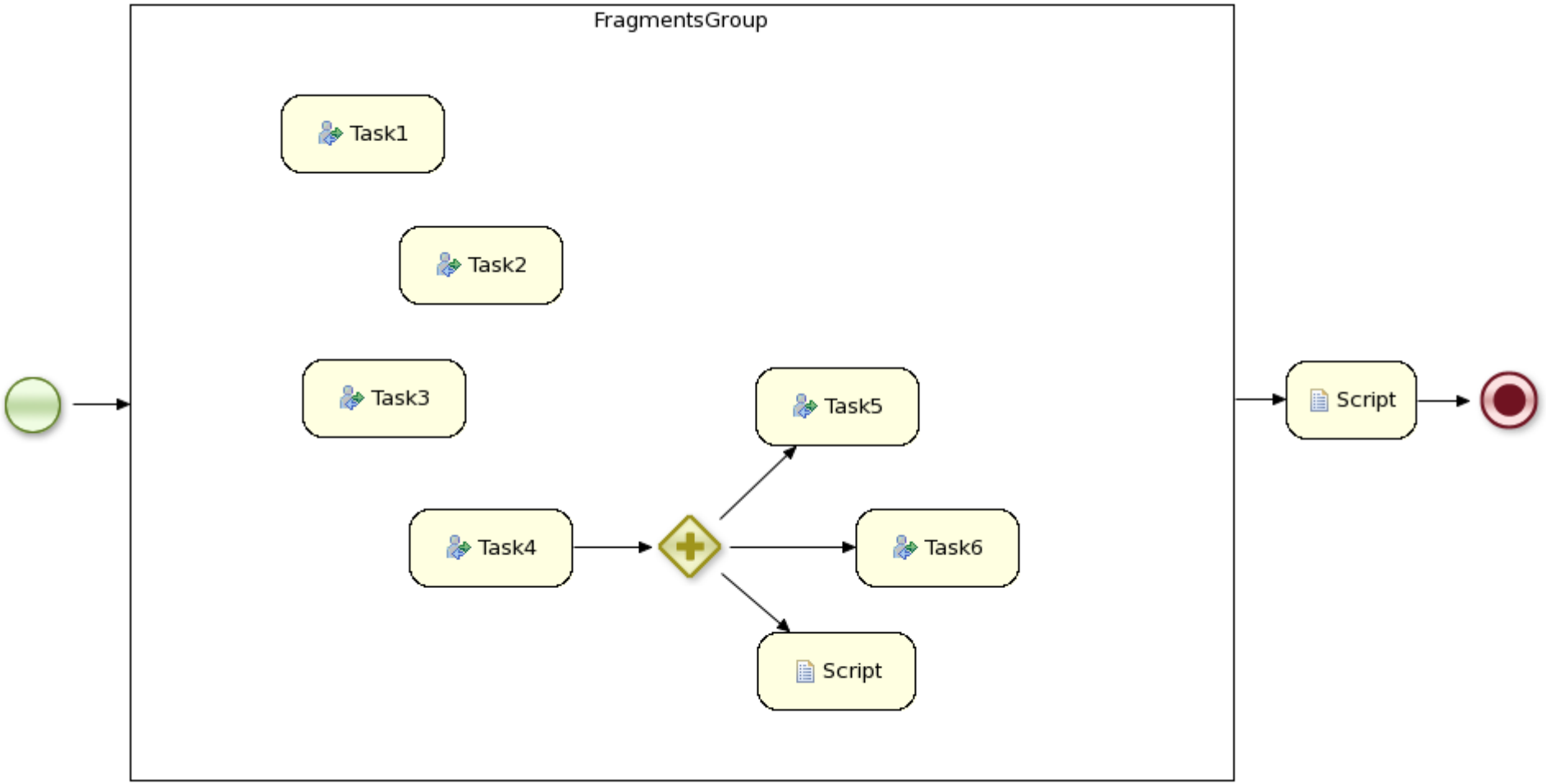
=

Variability

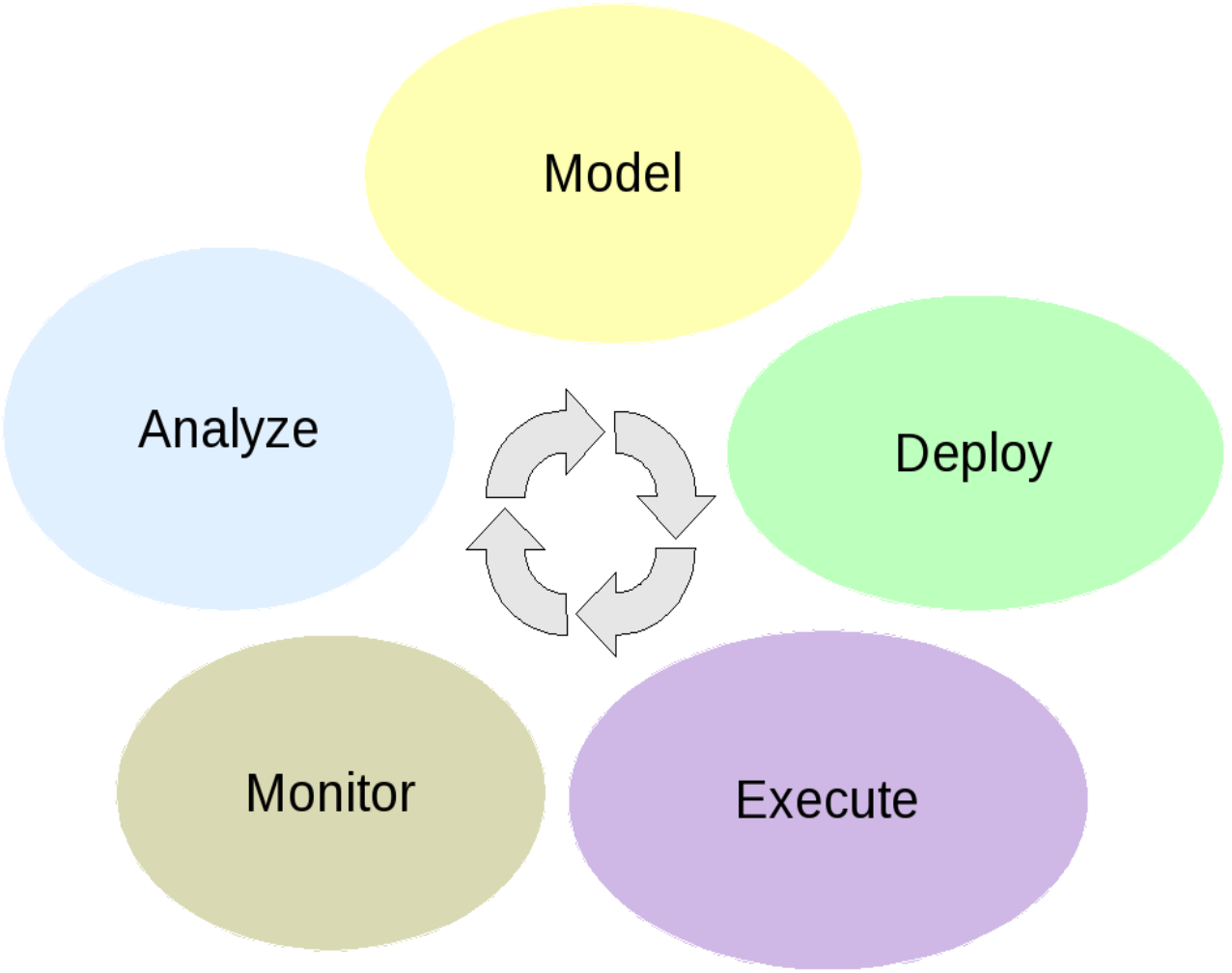
+

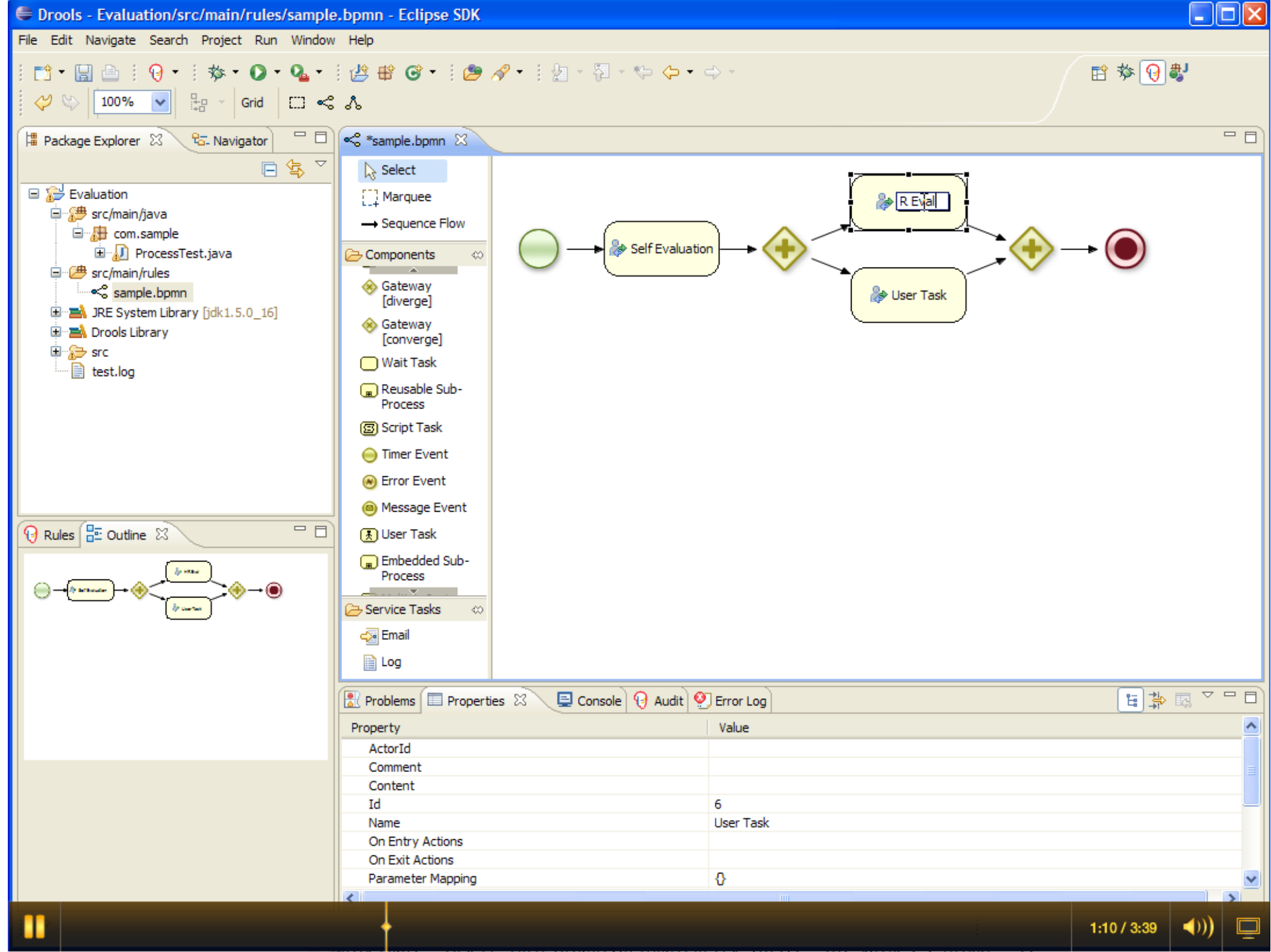
Change

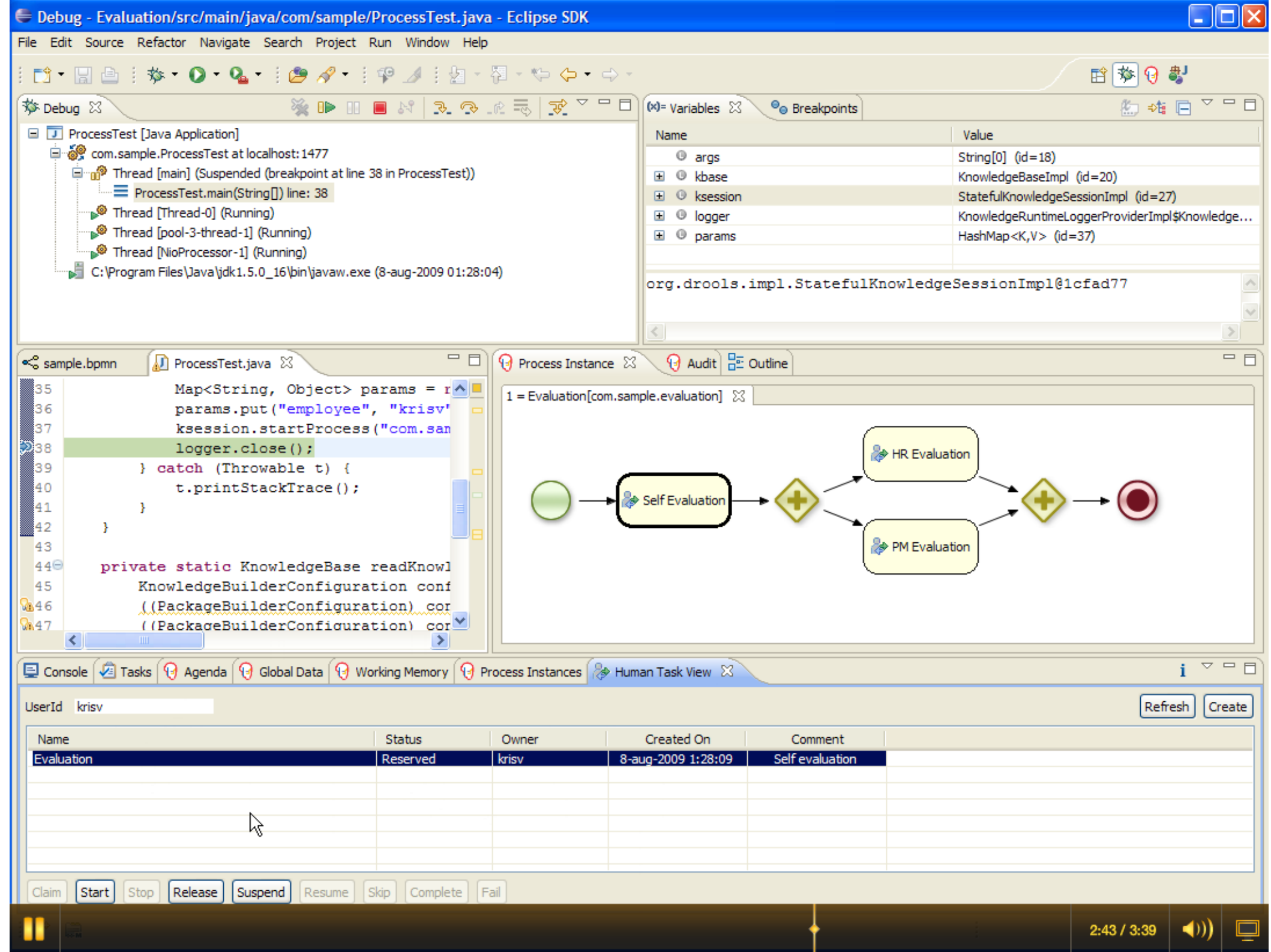
Non-lineair processes

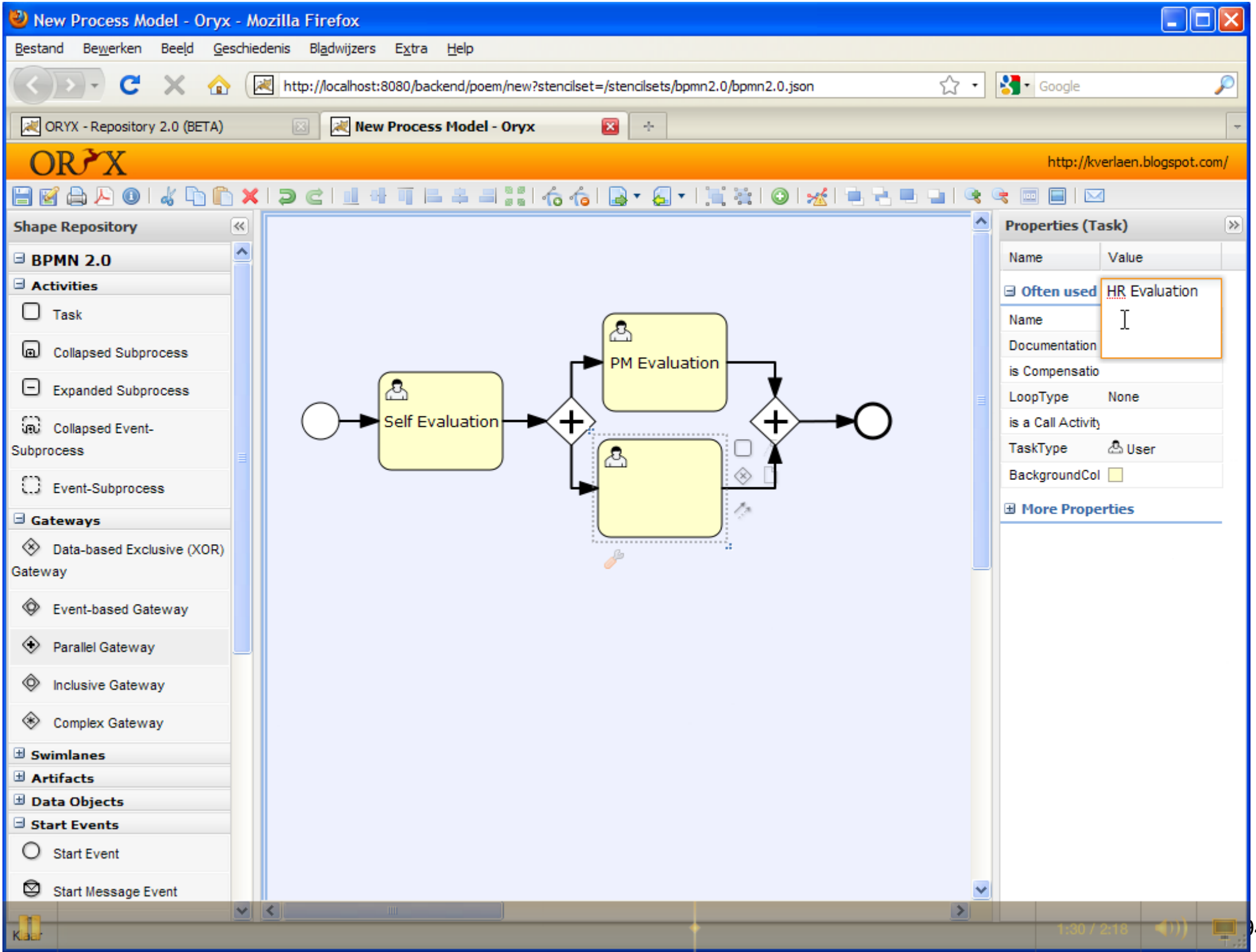


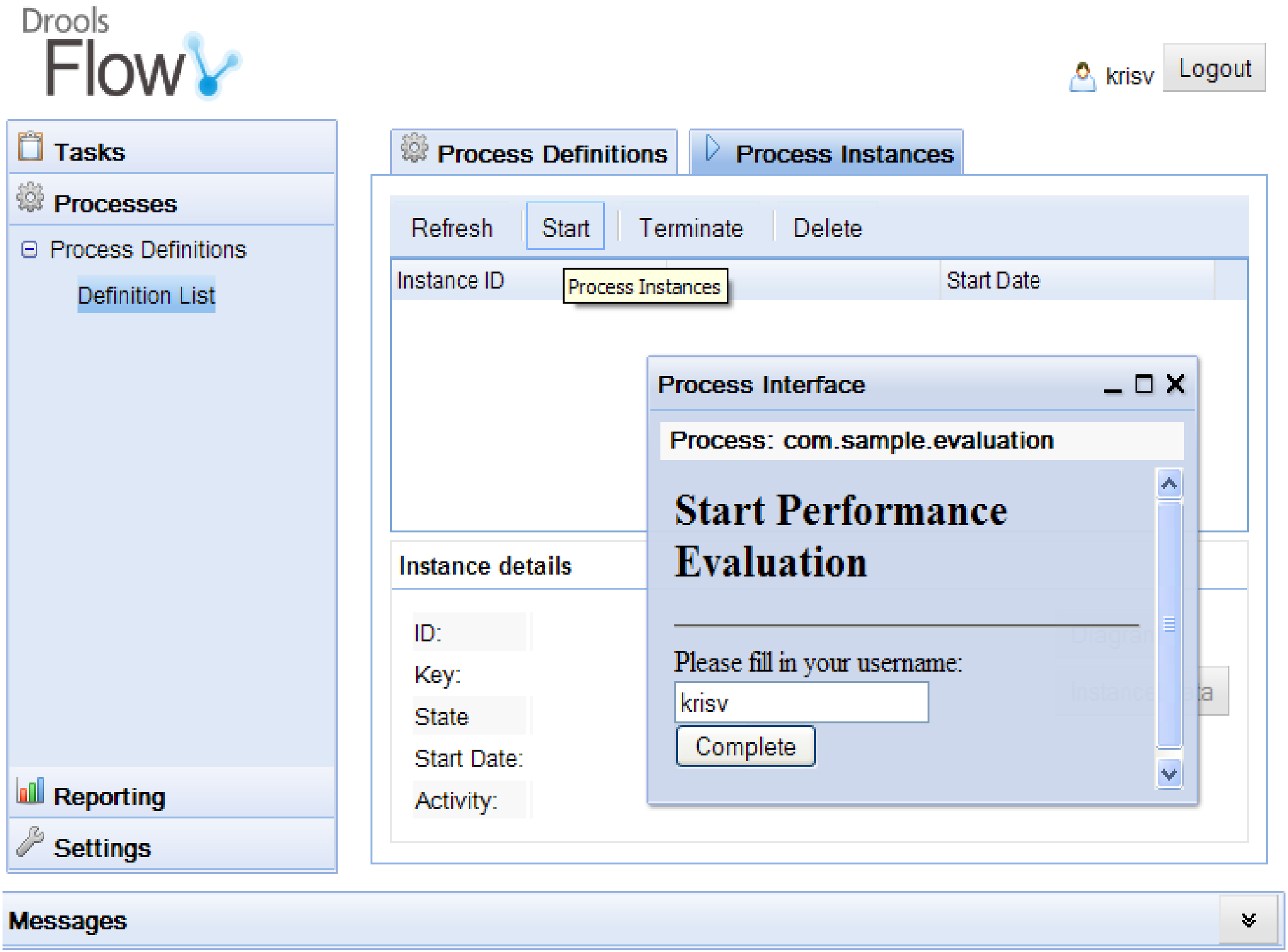
Knowledge Life Cycle

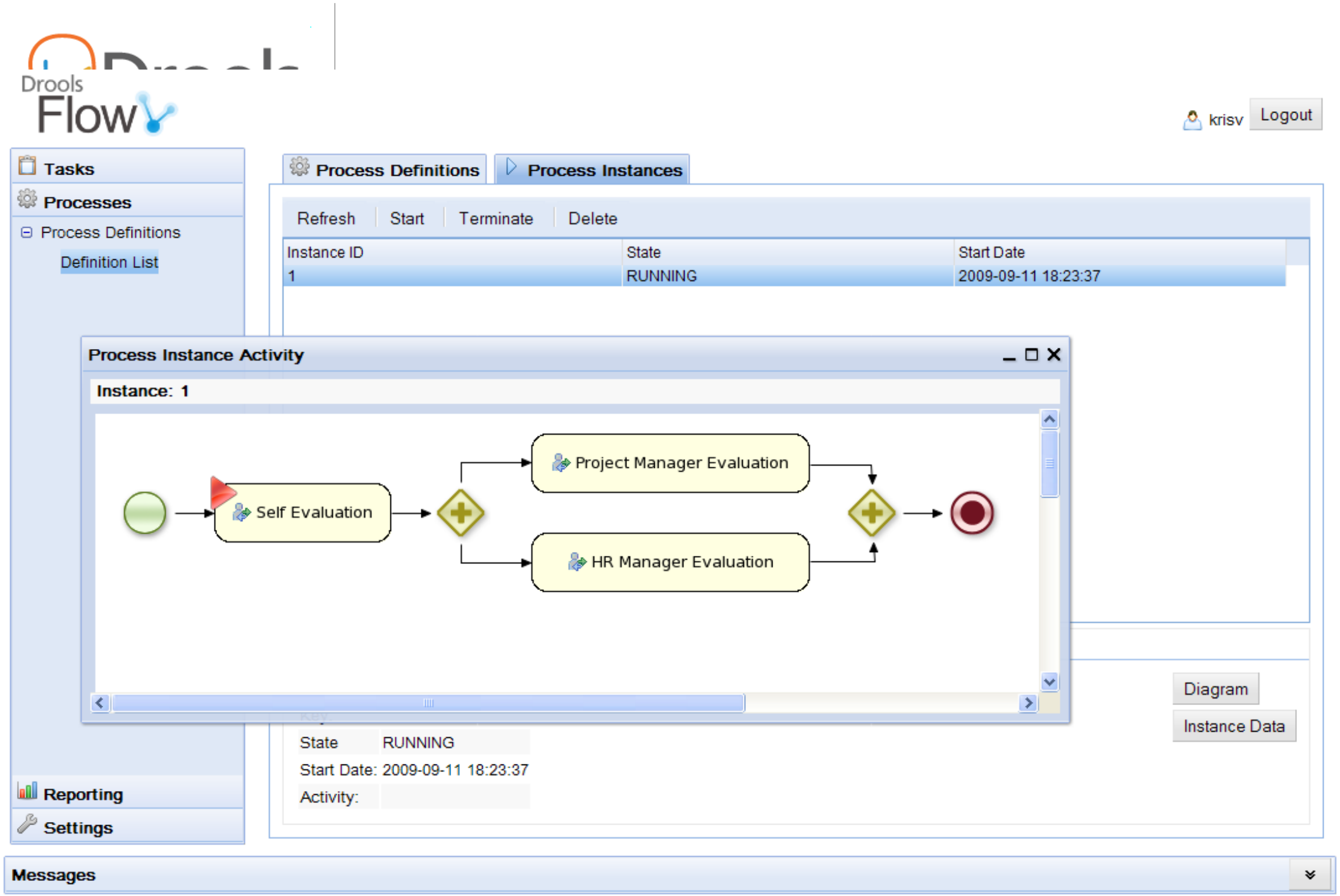


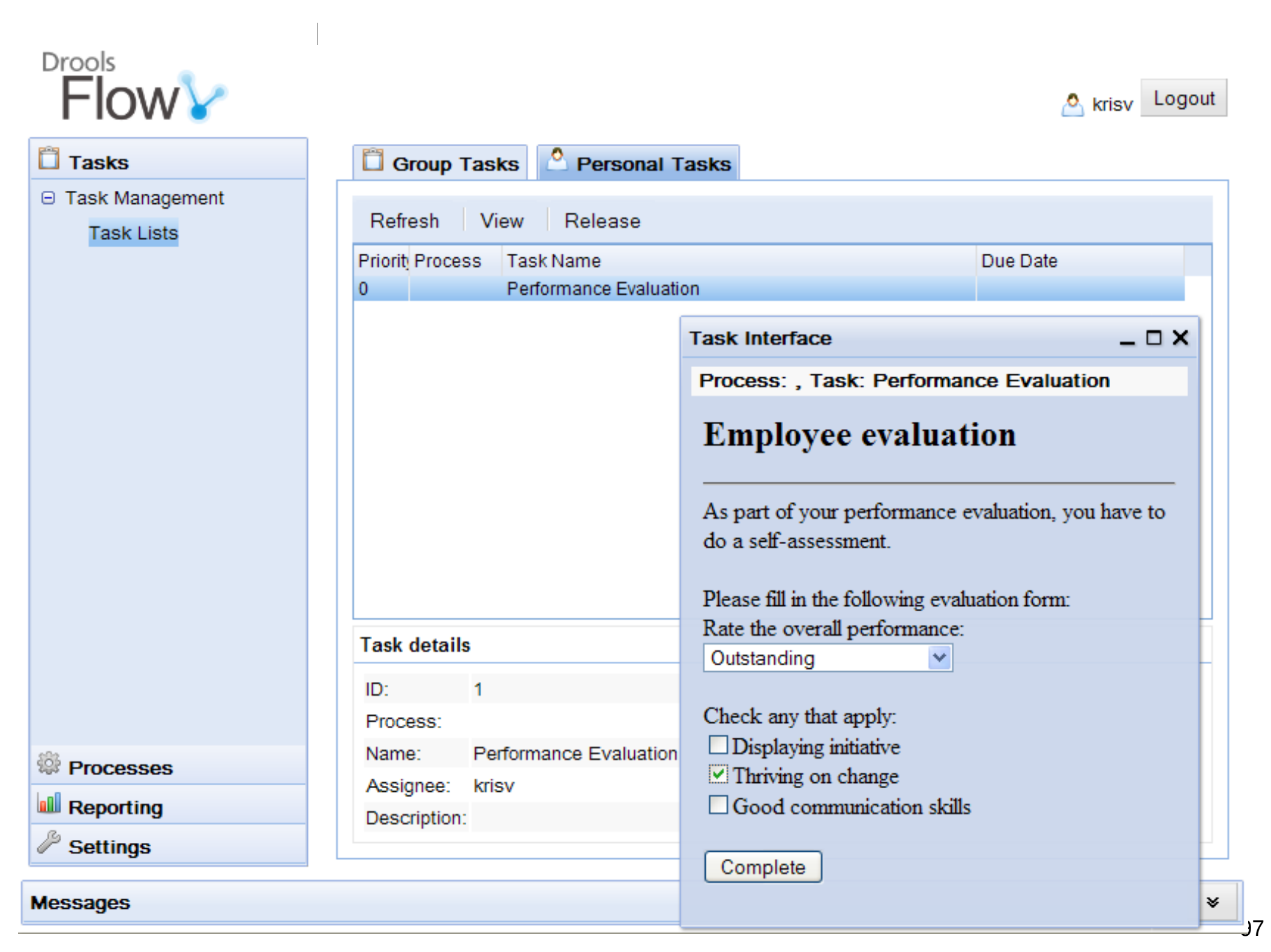

Drools - Evaluation/src/main/rules/sample.bpmn - Eclipse SDK
File Edit Navigate Search Project Run Window Help
100%
Grid
Package Explorer
Navigator
Evaluation
src/main/java
com.sample
ProcessTest.java
src/main/rules
sample.bpmn
JRE System Library [jdk1.5.0_16]
Drools Library
src
test.log
Rules
Outline
*sample.bpmn
Select
Marquee
Sequence Flow
Components
Gateway [diverge]
Gateway [converge]
Wait Task
Reusable Sub-Process
Script Task
Timer Event
Error Event
Message Event
User Task
Embedded Sub-Process
Service Tasks
Email
Log
Self Evaluation
R Eval
User Task
Problems
Properties
Console
Audit
Error Log
Property
Value
ActorId
Comment
Content
Id
6
Name
User Task
On Entry Actions
On Exit Actions
Parameter Mapping
{}
1:10 / 3:39


Debug - Evaluation/src/main/java/com/sample/ProcessTest.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
ProcessTest [Java Application]
com.sample.ProcessTest at localhost:1477
Thread [main] (Suspended (breakpoint at line 38 in ProcessTest))
ProcessTest.main(String[]) line: 38
Thread [Thread-0] (Running)
Thread [pool-3-thread-1] (Running)
Thread [NioProcessor-1] (Running)
C:\Program Files\Java\jdk1.5.0_16\bin\javaw.exe (8-aug-2009 01:28:04)
Variables
Breakpoints
Name
Value
args
String[0] (id=18)
kbase
KnowledgeBaseImpl (id=20)
ksession
StatefulKnowledgeSessionImpl (id=27)
logger
KnowledgeRuntimeLoggerProviderImpl$Knowledge...
params
HashMap<K,V> (id=37)
org.drools.impl.StatefulKnowledgeSessionImpl@1cfad77
sample.bpmn
ProcessTest.java
Map<String, Object> params = r
params.put("employee", "krisv"
ksession.startProcess("com.san
logger.close();
} catch (Throwable t) {
t.printStackTrace();
}
}
private static KnowledgeBase readKnowl
KnowledgeBuilderConfiguration conf
((PackageBuilderConfiguration) cor
Process Instance
Audit
Outline
1 = Evaluation[com.sample.evaluation]
HR Evaluation
Self Evaluation
PM Evaluation
Console
Tasks
Agenda
Global Data
Working Memory
Process Instances
Human Task View
UserId krisv
Refresh
Create
Name
Status
Owner
Created On
Comment
Evaluation
Reserved
krisv
8-aug-2009 1:28:09
Self evaluation
Claim
Start
Stop
Release
Suspend
Resume
Skip
Complete
Fail
2:43 / 3:39


New Process Model - Oryx - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
http://localhost:8080/backend/poem/new?stencilset=/stencilsets/bpmn2.0/bpmn2.0.json
Google
ORYX - Repository 2.0 (BETA)
New Process Model - Oryx
ORYX
http://kverlaen.blogspot.com/
Shape Repository
BPMN 2.0
Activities
Task
Collapsed Subprocess
Expanded Subprocess
Collapsed Event-Subprocess
Event-Subprocess
Gateways
Data-based Exclusive (XOR) Gateway
Event-based Gateway
Parallel Gateway
Inclusive Gateway
Complex Gateway
Swimlanes
Artifacts
Data Objects
Start Events
Start Event
Start Message Event
Self Evaluation
PM Evaluation
Properties (Task)
Name
Value
Often used
HR Evaluation
Name
Documentation
is Compensatio
LoopType
None
is a Call Activity
TaskType
User
BackgroundCol
More Properties
Klaar
1:30 / 2:18


Drools
Flow
krisv
Logout
Tasks
Processes
Process Definitions
Definition List
Reporting
Settings
Process Definitions
Process Instances
Refresh
Start
Terminate
Delete
Instance ID
Process Instances
Start Date
Process Interface
Process: com.sample.evaluation
Start Performance Evaluation
Please fill in your username:
krisv
Complete
Instance details
ID:
Key:
State
Start Date:
Activity:
Messages


Drools Flow
krisv
Logout
Tasks
Processes
Process Definitions
Definition List
Reporting
Settings
Process Definitions
Process Instances
Refresh
Start
Terminate
Delete
Instance ID
State
Start Date
1
RUNNING
2009-09-11 18:23:37
Process Instance Activity
Instance: 1
Project Manager Evaluation
Self Evaluation
HR Manager Evaluation
Diagram
Instance Data
State
RUNNING
Start Date:
2009-09-11 18:23:37
Activity:
Messages


Drools
Flow
krisv
Logout
Tasks
Task Management
Task Lists
Processes
Reporting
Settings
Group Tasks
Personal Tasks
Refresh
View
Release
Priorit
Process
Task Name
Due Date
0
Performance Evaluation
Task Interface
Process: , Task: Performance Evaluation
Employee evaluation
As part of your performance evaluation, you have to do a self-assessment.
Please fill in the following evaluation form:
Rate the overall performance:
Outstanding
Check any that apply:
Displaying initiative
Thriving on change
Good communication skills
Complete
Task details
ID:
1
Process:
Name:
Performance Evaluation
Assignee:
krisv
Description:
Messages

Drools Flow

krisv Logout

- Tasks
- Processes
- Reporting
  - Available Reports
    - Processes
- Settings

Process Reports

Available Reports

Drools

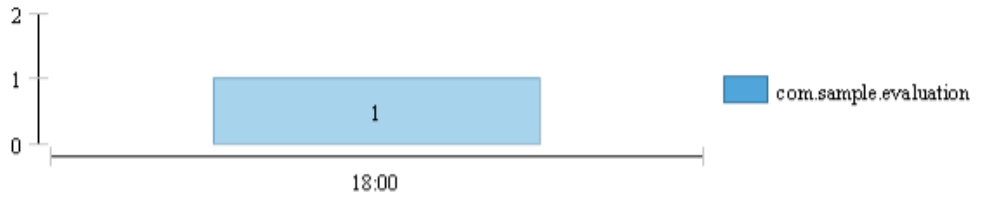
11 september 2009

Business Activity Monitoring

Process com.sample.evaluation

Process Definition Id:	com.sample.evaluation
Total number of instances:	1
Number of instances last 24h:	1
Number of active instances:	1

Start Process Instances



Id	Start	End
1	11 september 2009 18:23:37	

Messages

Drools Fusion

Complex Event Processing (CEP)

Terminology

Event

a significant change of state at a particular point in time

Complex event

an abstraction of other events called its members

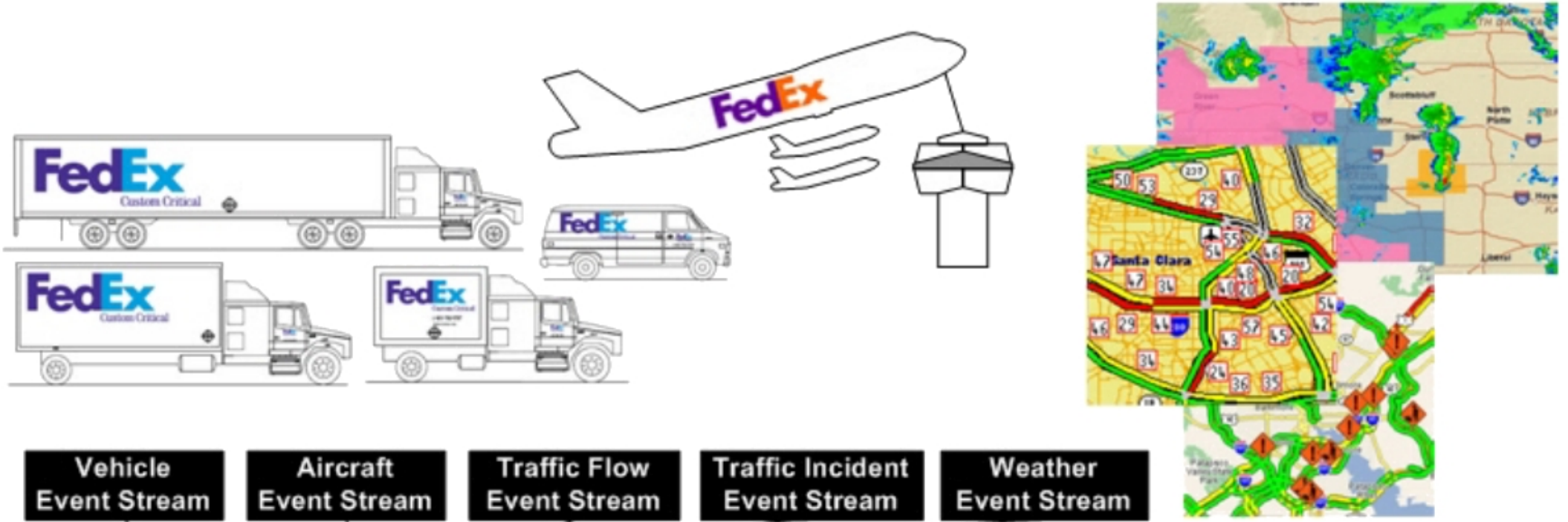
Complex event processing

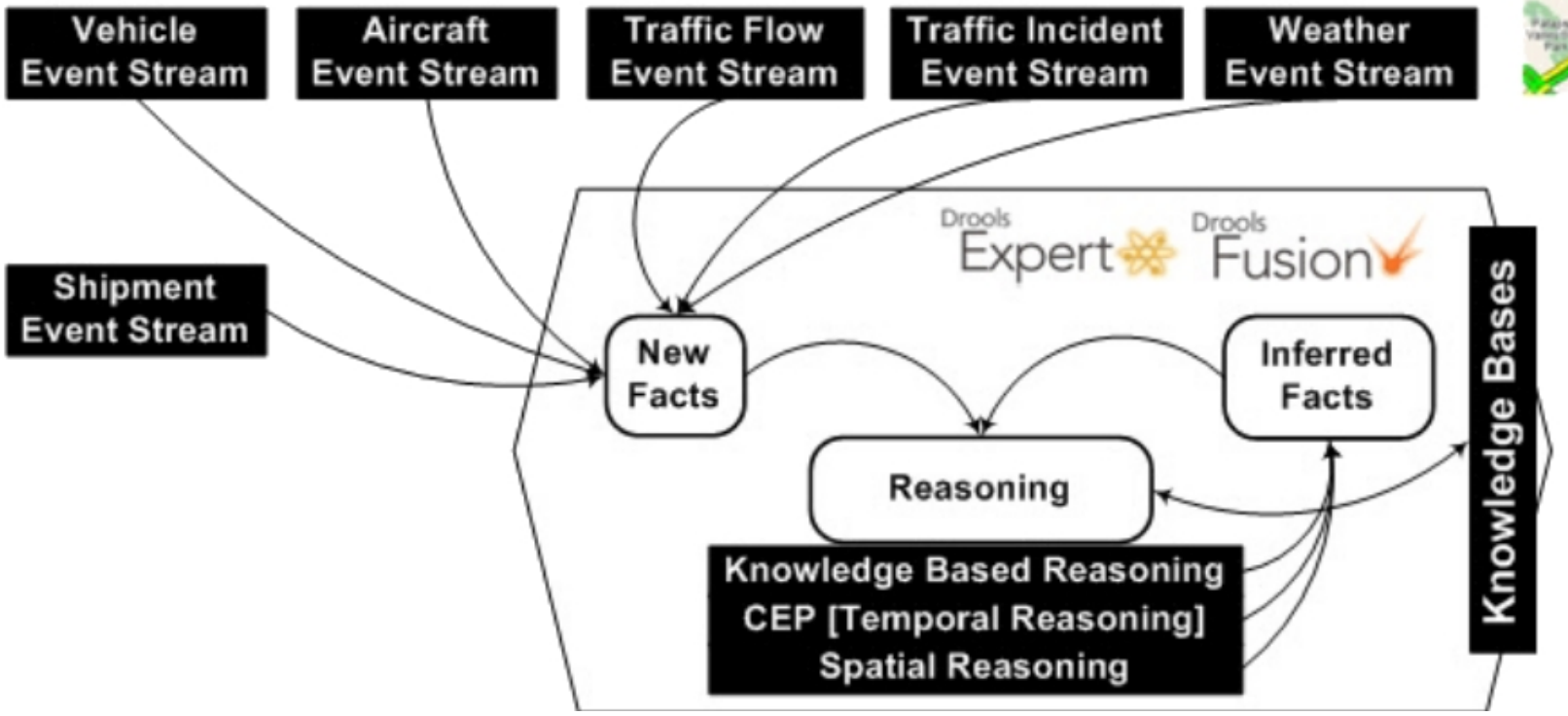
processing multiple events with the goal of identifying the meaningful events within the event cloud

Characteristics

- Huge volume of events, but only a few of real interest
- Usually events are immutable
- Usually queries/rules have to run in reactive mode
- Strong temporal relationships between events
- Individual events are usually not important
- The composition and aggregation of events is important

A “Simple” Example





Features

- Event Detection
  - Events are facts
  - Time + duration
  - Event streams + pipelines
- Event Correlation
  - 13 temporal operators
  - Sliding windows
    - Accumulate ... over window:time(12h)
  - Temporal dimension support
- Event abstraction
  - Compose complex events

Code Example

```
declare StockTick

  @role( event )

  symbol : String
  price : double

end
```

```
rule “Shipment not picked up in time”
when
   Shipment( $pickupTime : scheduledPickupTime )
   not ShipmentPickup( this before $pickupTime )
then
   // shipment not picked up, action required.
end
```

Summary

Summary

- 3 different paradigms
  - Business Rules
  - Business Processes
  - Complex Event Processing
- Unification and integration
  - Life cycle
  - Tooling
  - API

Q & A

Questions?

Useful links

- Website
  - http://www.jboss.org/drools/
- Reference manual
  - http://www.jboss.org/drools/documentation.html
- Blog
  - http://blog.athico.com/
- Mailing lists (forum through nabble)
  - http://www.jboss.org/drools/lists.html