ITUNIVERSITY.US
FROM HARD-CODED CONDITIONS TO RULES

- The benefits:
  - Allow decision makers to directly introduce their business rules with minimum translation involved
- You will learn about JBoss Drools and Guvnor GUI
  - With abbreviated example of Product Marketing
- Look for introduction to semantic enhancements helping to bridge business language and rules
  - With a brief exposure to Decision Tables and Business Architecture Sandbox for Enterprise (BASE)

JBoss Drools and Guvnor Introduction
From Code to Rule-based Development

```
if(person.getAge() > 18)
offers.add(creditCard)
```

```
<rule name="person age more than 18">
 <parameter identifier="person">
  <class>org.drools.examples.simple.Person</class>
 </parameter>

 <java:condition>person.getAge() > 18</java:condition>
 <java:consequence>
  offers.add(creditCard)
 </java:consequence>

</rule>
```

From a rule to a rule-set

```
<rule-set name="offer rules" xmlns="http://drools.org/rules"
xmlns:java="http://drools.org/semantics/java"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://drools.org/rules rules.xsd
http://drools.org/semantics/java java.xsd">

<rule name="person age more than 18">
  <parameter identifier="person">
    <class>org.drools.examples.simple.Person</class>
  </parameter>

 <java:condition>person.getAge() > 18</java:condition>
 <java:consequence>
    offers.add(creditCard)
 </java:consequence>

</rule>

</rule-set>
```

The Rule in a DRL File

```
import org.drools.examples.simple.Person
import org.drools.examples.simple.Offers
import org.drools.examples.simple.CreditCard

Rule: person age more than 18

  when
        person.age > 18

  then
        offers.add(creditCard);

  end
```

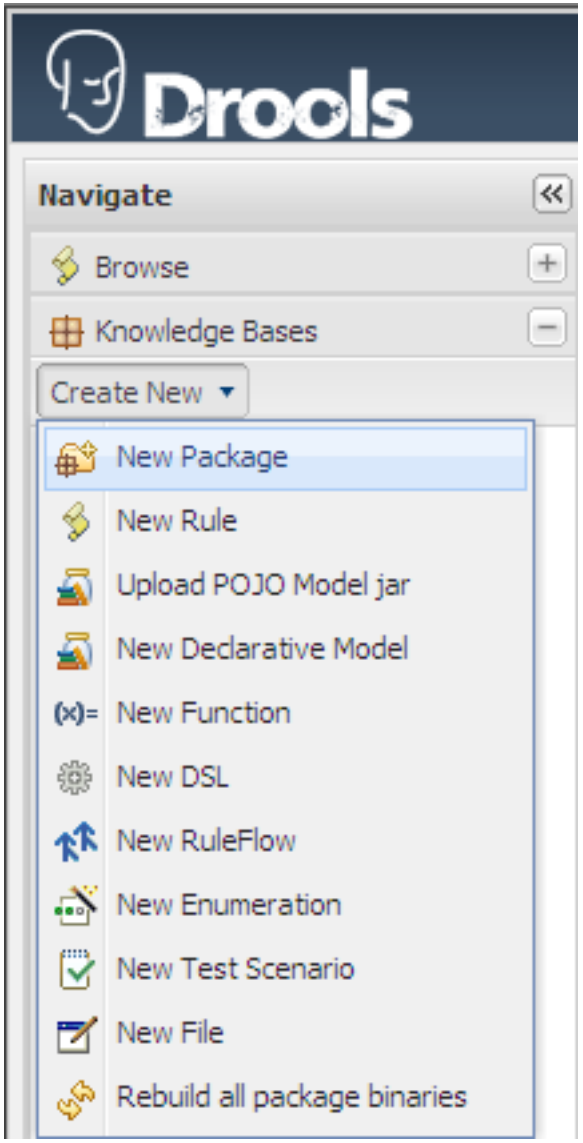
EXECUTE RULES IN JAVA PROGRAM

```
//load the rules from the rules.java.drl file

RuleBase ruleBase = RuleBaseLoader.loadFromStream(
this.getClass().getResourceAsStream( "rules.java.drl" ) );

WorkingMemory workingMemory =
ruleBase.newWorkingMemory( );

workingMemory.assertObject(person); // get data object

workingMemory.fireAllRules( );
```

GUVNOR (DROOLS ON THE WEB)
LOGIN, THEN CREATE CATEGORY AND PACKAGE

http://server:8080/drools-guvnor

Before creating any rules we need to establish a Category and a Package where we plan to create the rules.



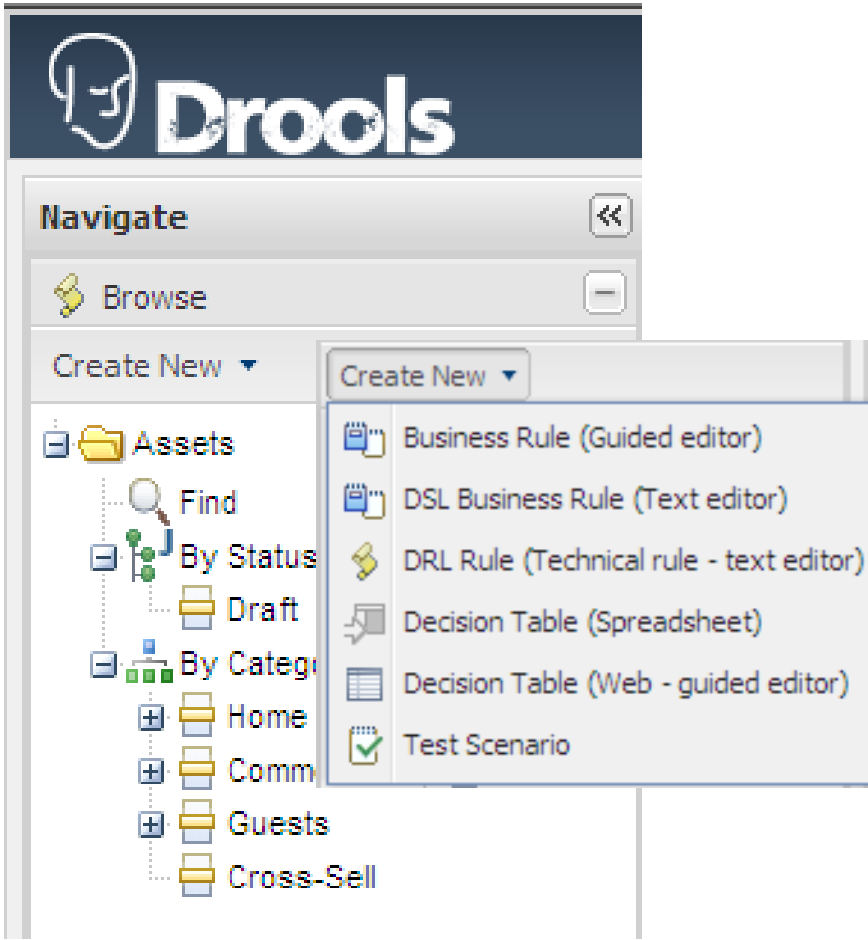
GUVNOR (DROOLS ON THE WEB)
CREATE A NEW RULE

Click on the **Browse** control

See **Assets**

Each Rule is an **Asset**

Select
Create New – Business Rule
or
Create New – DRL Rule



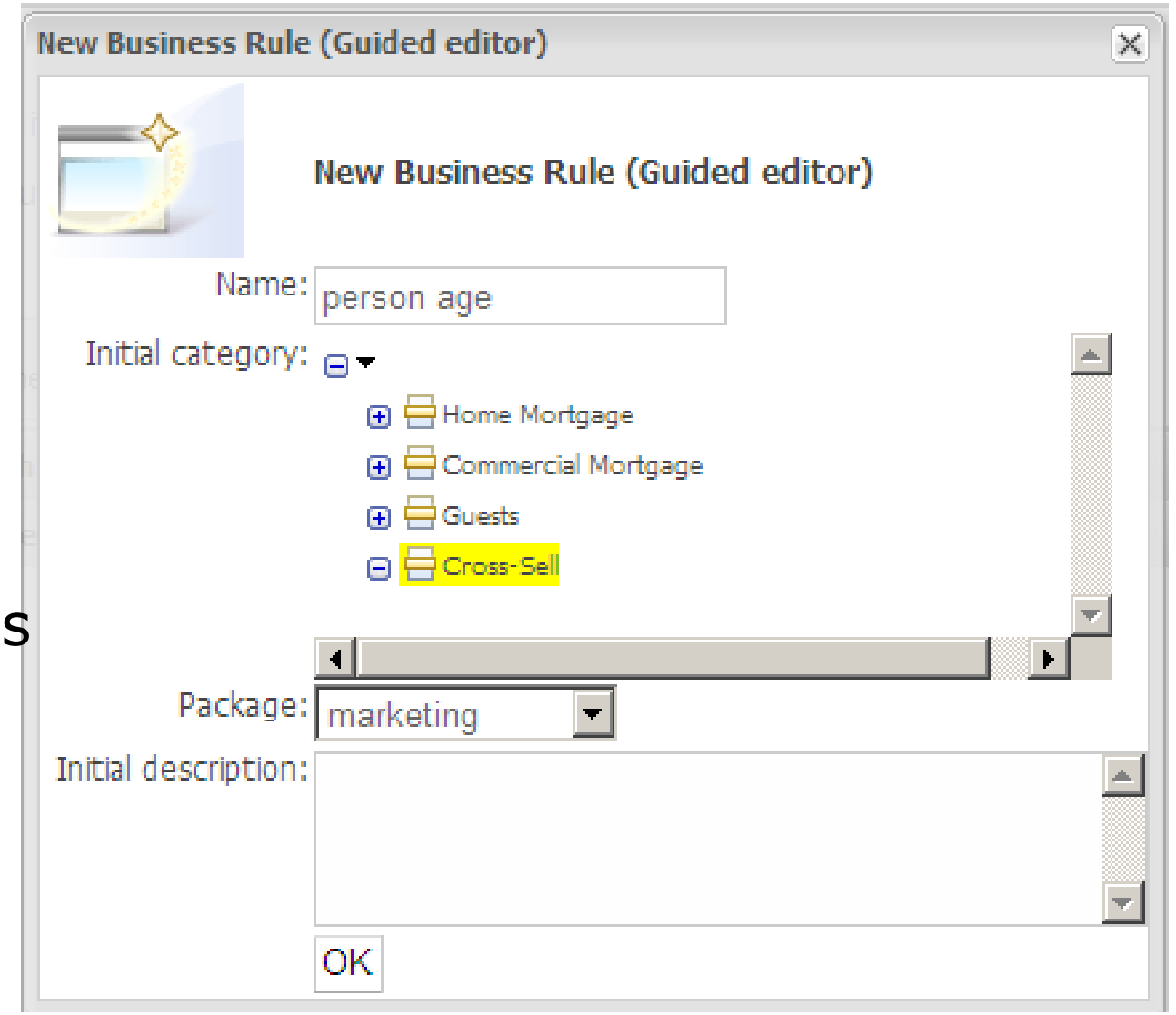
Guvnor (Drools on the Web)
New Business Rule (Guided Editor)

Select Cross-Sell **Category**

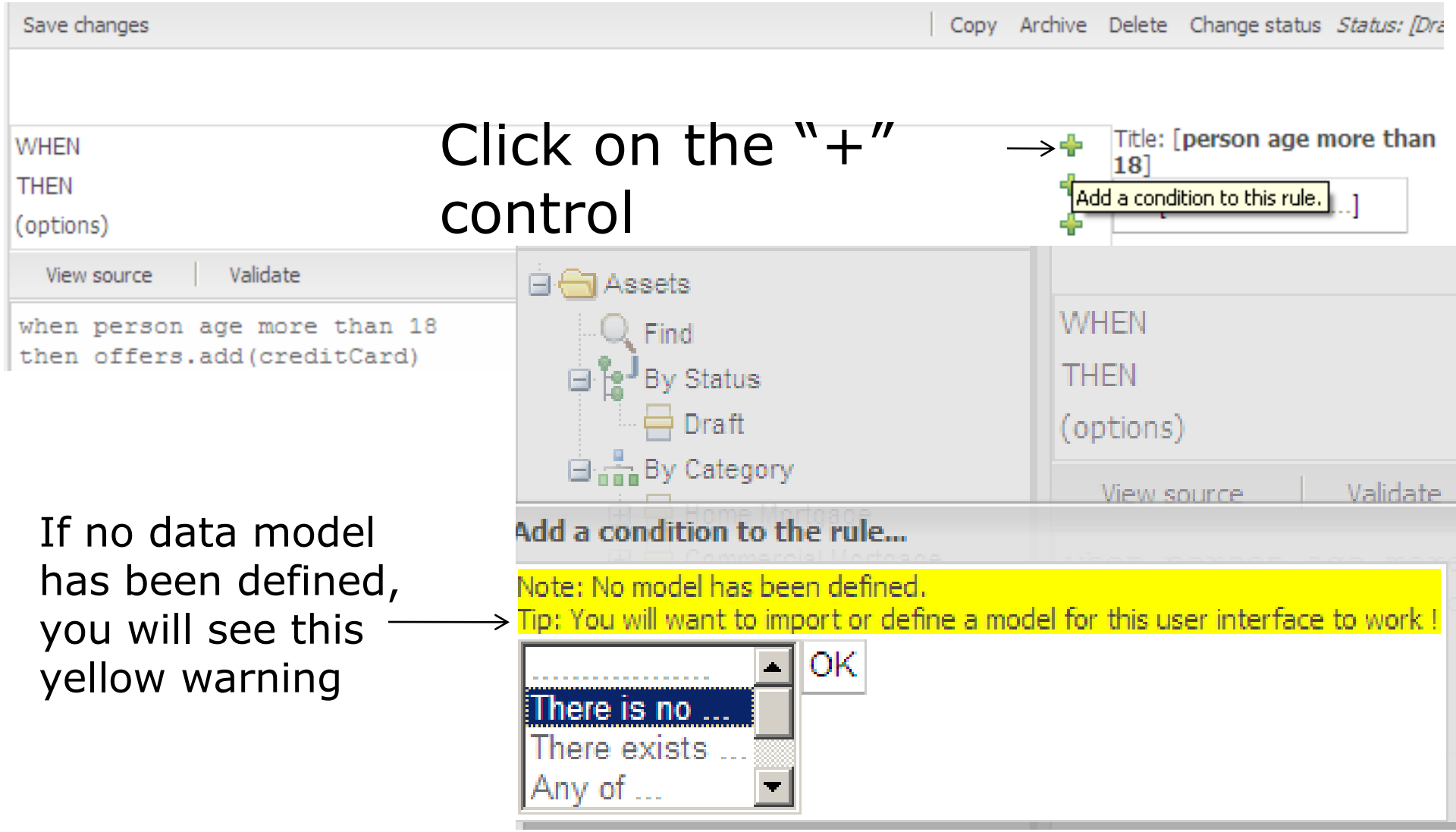
Select Marketing **Package**

Enter the Business Rule **Name**

Press OK



Guvnor (Drools on the Web)
Rule Condition and Consequence



If no data model has been defined, you will see this yellow warning

Rule-based Development
Product Marketing Example

A **user** with a **web browser** or iPhone accesses one of the web sites, which offers a set of products based on user's profile and Product rules

Integrated set of the **web sites** and IVR

Product Web Service With the embedded **Rules Engine**

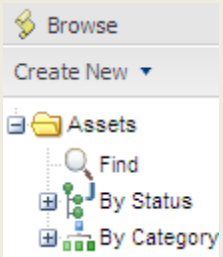
Rules

Production Data Warehouse

After successful testing a new or updated Rules file will be moved to production

Collaborative work of a business analyst (BA) and a rules SME while creating and updating rules with Guvnor GUI in BRP

Guvnor



Rules

Product Web Service With the embedded **Rules Engine**

Test/Business Rules Playground (BRP)

Then, the **BA** and/or a **tester** with a **web browser** or iPhone accesses one of the **test** sites, which offers a set of products based on user's profile and Product rules

Integrated set of the **test sites**

Data Structure is a Model.
Instances of the Structure are Facts.

- Some models come from an application (i.e. Person).
- They are loaded into java classes.
- Classes define a model imported as Java jar file.

- User may want to complement the application model with additional entities that are used mainly during the reasoning process (i.e. flexible rules).
- Users can define the model directly to the rules engine.
- You can create new types directly in DRL.

- Facts are asserted to run with specific data.
- JBoss + Spring frameworks connect rules with data.
- Data and a model must be available before firing rules!

(Drools Expert User Guide)

EXAMPLE: PRODUCT RULES

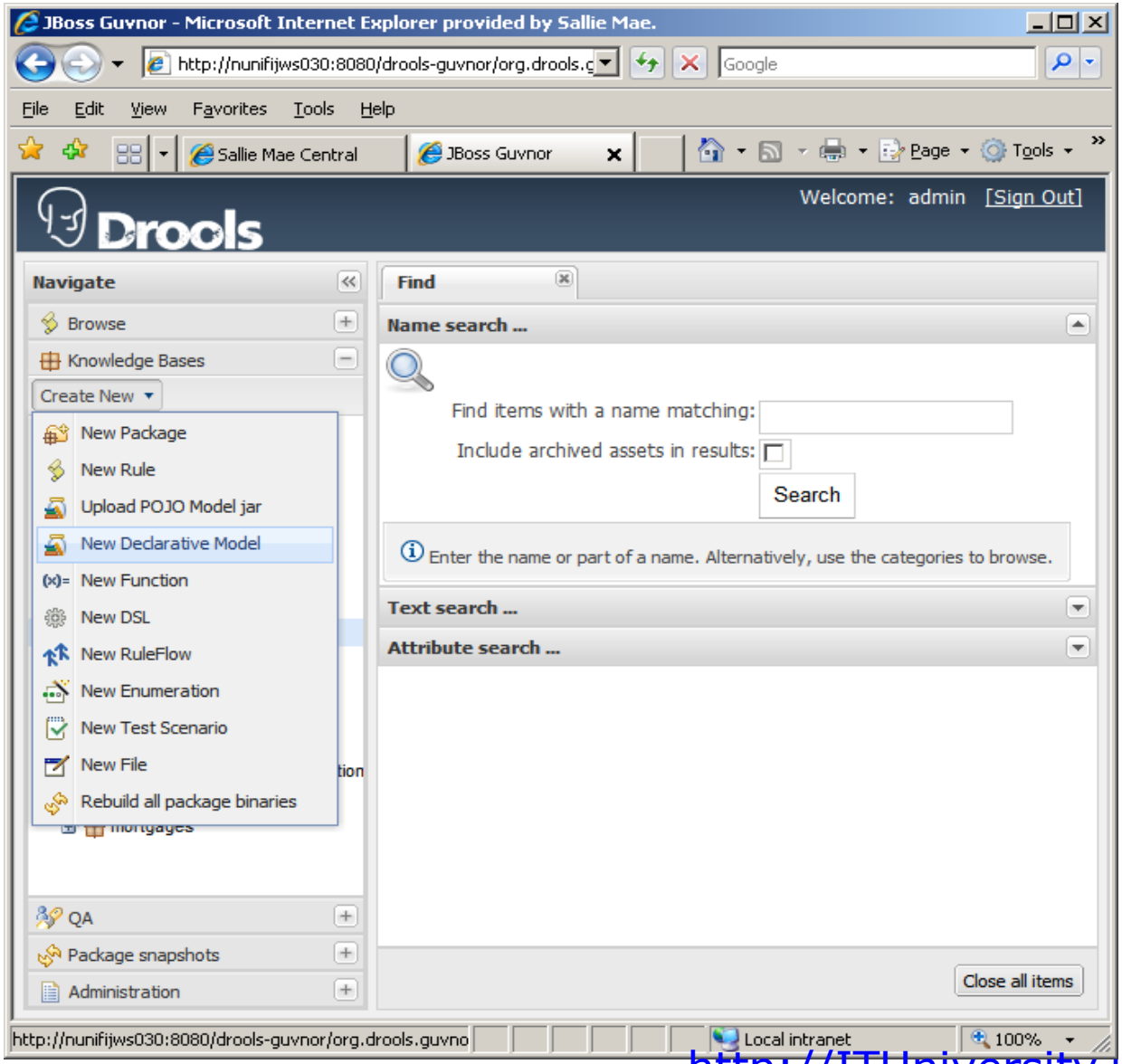
- When …
  - The person does have the product
  - The product is on the list
  - The ad type has not been supplied too recently
  - The ad size matches the web page
  - The web site is authorized to sell the product
  - (The person has not applied for the product)
- Then …
  - Offer the product with the highest score

INPUT...DECLARE A MODEL

Processing ->

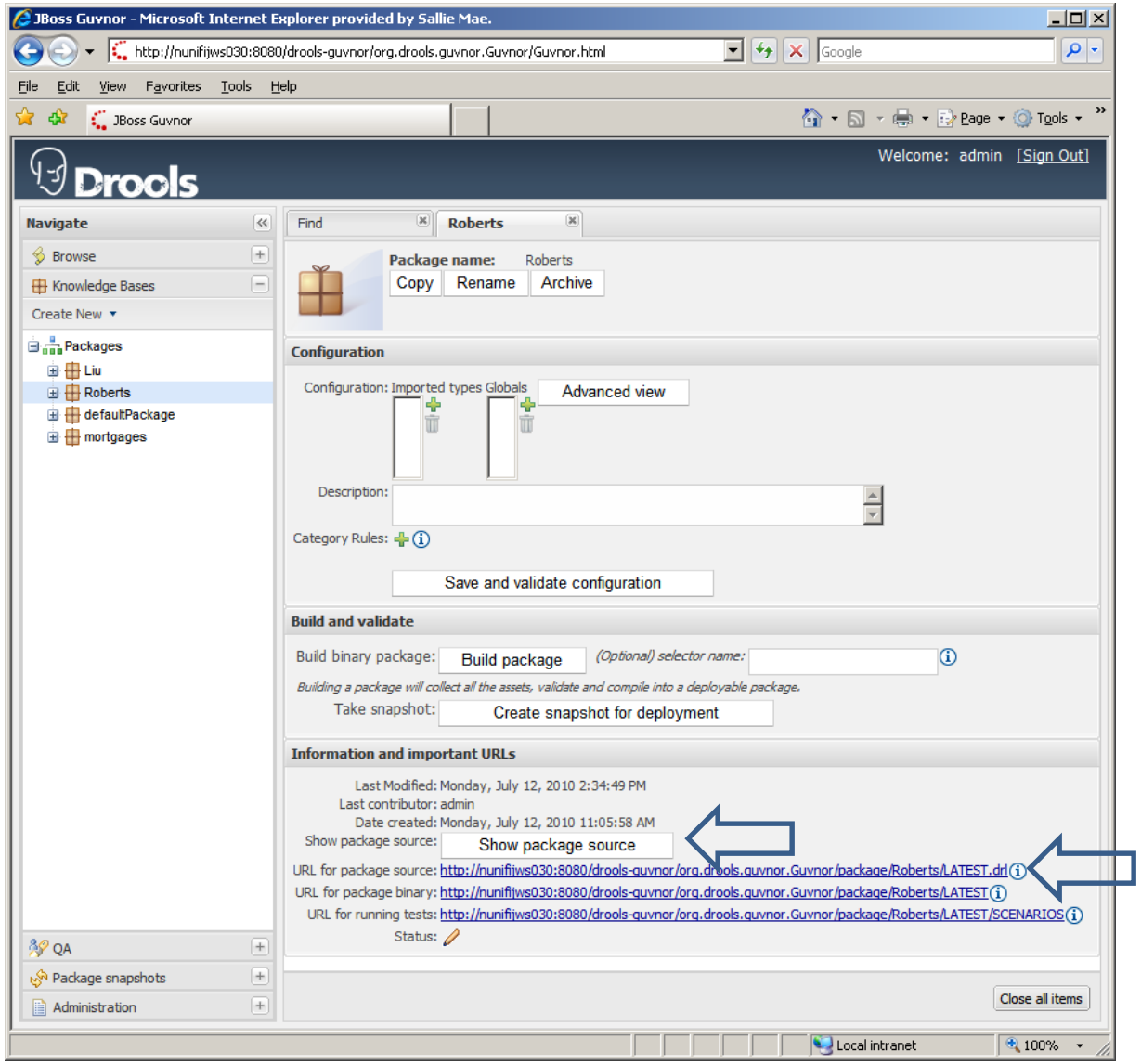
Input ->

Output ->



SAVE DRL FILE

1. Select your package under Knowledge Bases
2. Click "Show package source" to view DRL
3. Click on "URL for package source" to save



Objectives –Java Code Example with The Rules

- Two ways to create Rules
  - Modeling Rules
  - Writing Rules
- Working sample for “The Rules”
- Data Discussion

RULES: JAVA METHOD EXAMPLE

```
public void applySimpleRule(ProductRequest req, ArrayList<Product> marketingProducts){
    StatefulKnowledgeSession session;
    try {
        KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
        kbuilder.add( ResourceFactory.newClassPathResource ("SampleRule.drl",
    getClass()),ResourceType.DRL);

        KnowledgeBuilderErrors buildErrors = kbuilder.getErrors();
        if(buildErrors.size() > 1){
            log.error("Error encountered when building the rules... ");
        }

        KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
        kbase.addKnowledgePackages( kbuilder.getKnowledgePackages() );
        session = kbase.newStatefulKnowledgeSession();
        session.insert(req);
        session.insert(marketingProducts);
        session.insert(log);
        session.fireAllRules();
        session.dispose();
    } catch ( Exception e ) {
        e.printStackTrace();
        throw new RuntimeException( e );
    }
}
```

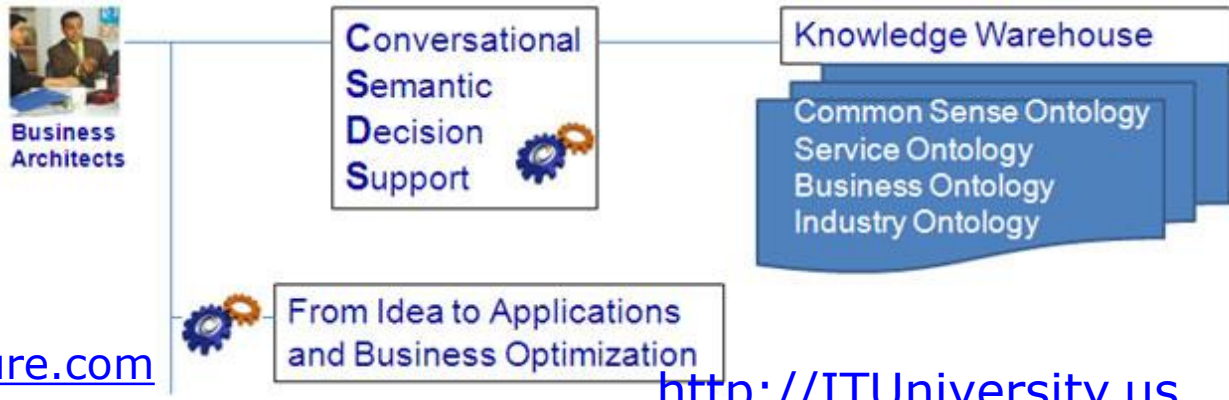
SUMMARY

- Why Rule-based development?
- What are the benefits of Code/Rules separation?
- What are the steps to create a new rule and what are the challenges?
- How Code/Rules are connected?
- How Data/Rules are connected?

Applying the Conversational, Semantic Approach to Rules

- While technology speaks XML and Web Services, business people prefer natural language (NL)
- A conversational semantic decision support can bridge these two worlds and provide mapping between NL and services

- For example, a Business Analyst (BA) writes a line of requirements: "application starts with login."
- The program would reply "Do you mean the Authentication Service?"



Reference: http://ITofTheFuture.com

HOMEWORK

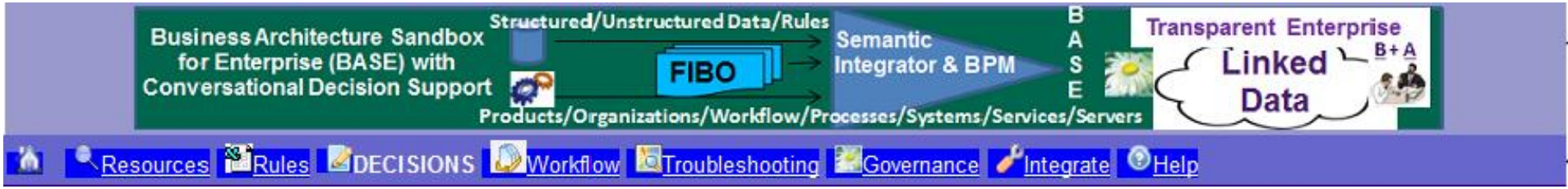
1. Answer Summary Questions

2. Download and Install BRMS from
   https://docs.jboss.org/drools/release/

BUSINESS ARCHITECTURE SANDBOX FOR ENTERPRISE (BASE)

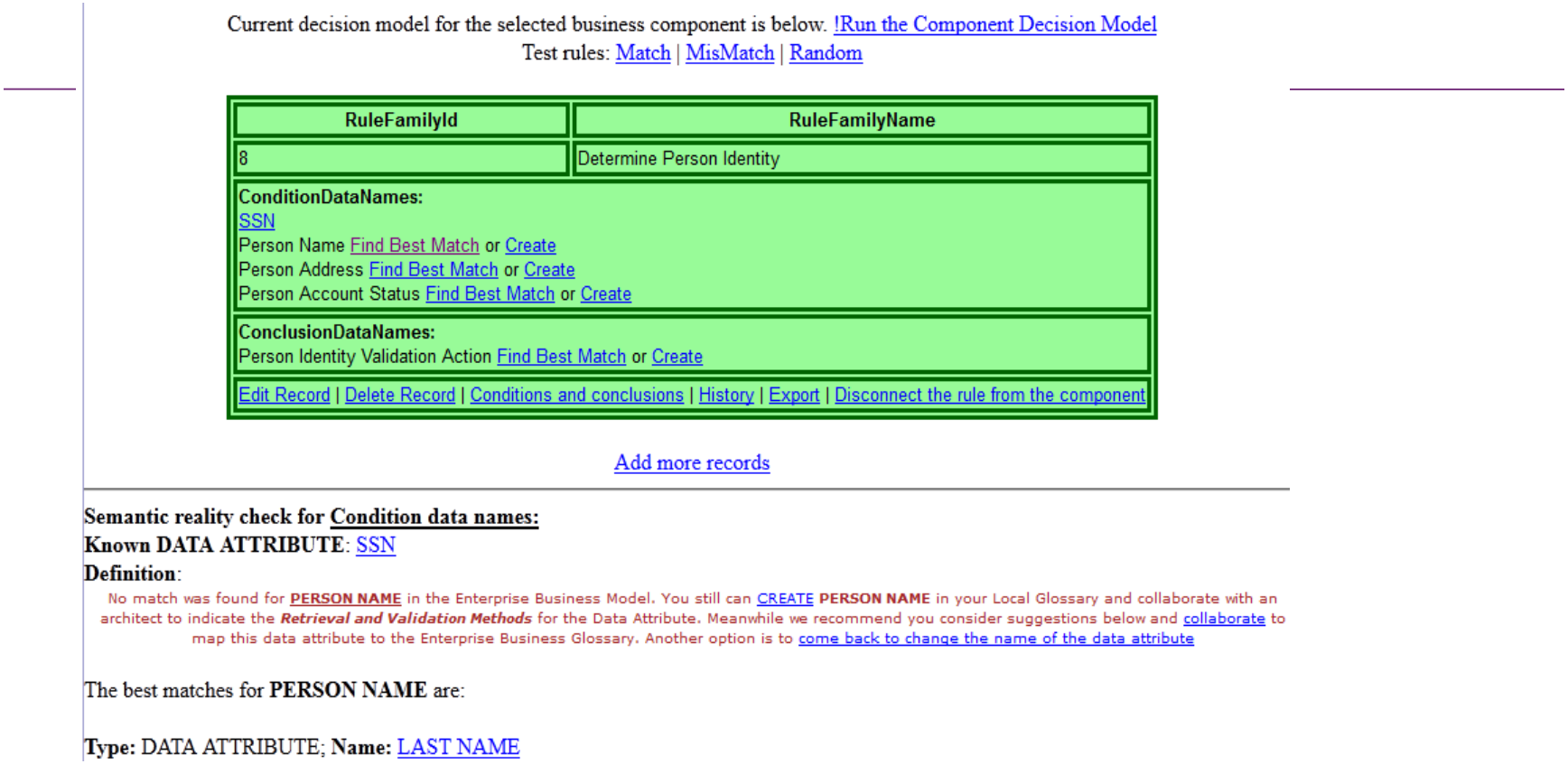
- BASE* offers to business an easy entrance and a playground to collaborate with IT.

- BASE helps placing the seeds of semantic technology in the current business ground and helps transitioning to Semantic Cloud Architecture.



Reference: http://ITofTheFuture.com

CREATING A RULE WITH SEMANTIC CHECK



Current decision model for the selected business component is below. !Run the Component Decision Model
Test rules: Match | MisMatch | Random

RuleFamilyId	RuleFamilyName
8	Determine Person Identity

ConditionDataNames:
SSN
Person Name Find Best Match or Create
Person Address Find Best Match or Create
Person Account Status Find Best Match or Create

ConclusionDataNames:
Person Identity Validation Action Find Best Match or Create

Edit Record | Delete Record | Conditions and conclusions | History | Export | Disconnect the rule from the component

Add more records

Semantic reality check for Condition data names:
Known DATA ATTRIBUTE: SSN
Definition:
No match was found for **PERSON NAME** in the Enterprise Business Model. You still can CREATE **PERSON NAME** in your Local Glossary and collaborate with an architect to indicate the ***Retrieval and Validation Methods*** for the Data Attribute. Meanwhile we recommend you consider suggestions below and collaborate to map this data attribute to the Enterprise Business Glossary. Another option is to come back to change the name of the data attribute

The best matches for **PERSON NAME** are:

Type: DATA ATTRIBUTE; **Name:** LAST NAME

The resulting screen displays this rule family and automatically produces the links for running and testing the model. The program provides the semantic check for Condition Data Names.

Some data attributes, like SSN, are already in the system, and some are not. The program gives recommendations on mapping the data names to similar data attributes, existing in the system, or creating new attributes on-the-fly.

DECISION TABLE BASED RULES ENGINE (RE)

List all rules | List all rule-based components | Selected Rule Family is Determine Person Identity (id#8)

The rule family is used by the following components: ENROLLMENT FOR WEB SERVICES WORKFLOW | STORE CUSTOMER PROFILE FROM THE WEB

Conditions				Conclusions	
SSN Edit/Delete Data Name	Person Name Edit/Delete Data Name	Person Address Edit/Delete Data Name	Person Account Status Edit/Delete Data Name	Person Identity Validation Action Edit/Delete Data Name	
Existing Value true Edit Condition	**Existing Value** true Edit Condition	**Existing Value** false Edit Condition	**Valid Value** In Good Standing Edit Condition	Message: Existing account is valid for practical purposes Edit Conclusion	Delete Rule
Existing Value true Edit Condition	**Existing Value** true Edit Condition	**Existing Value** false Edit Condition	**Valid Value** Not Valid Edit Condition	Message: accept new customer profile instead of exisisting one. Edit Conclusion	Delete Rule
Add Condition	Add Condition	Add Condition	Add Condition	Add Conclusion	

The rules are present as the rows and columns in the **decision table**. Each row is a separate rule, which includes several **conditions** and a **conclusion.**

BASE uses semantic approach to connect the rules and data (a common RE problem). Read "data know how to handle data" here: http://ITofTheFuture.com