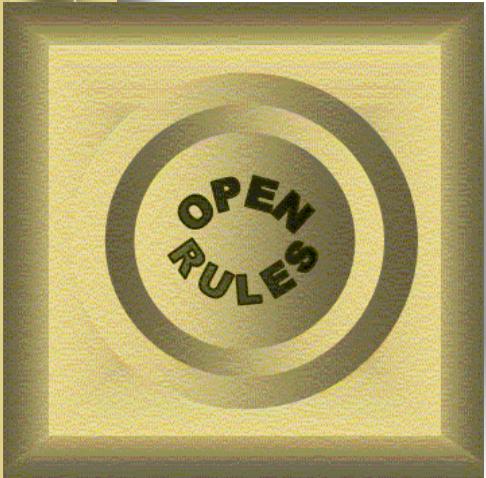


OMG Decision Modeling Information Day



OpenRules™

Creating, Testing, and Executing Decision Models with OpenRules-6

Presenter: Dr. Jacob Feldman
OpenRules, Inc., CTO

OPEN
RULES

Open Source BDMS

Search

SHARE



Open Rules

Business Rules - Time to Excel

Home • Product • Download • Services • Resources • Contact •

Decision Management System

OpenRules® is an Open Source Business Decision Management System oriented to Business Analysts. It allows subject matter experts (not developers) to create and maintain complex, rules-based, decision support applications.



Business Rules for Business Analysts

• • • • •

What Business Analyst Can Do:

Define Complex Business Decisions.
Represent Rules Families.
Define Business Glossaries.
Create Test Cases



What OpenRules Can Do:

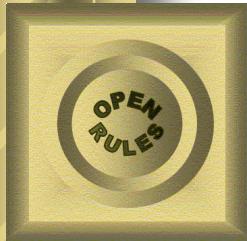
Efficient decision model execution.
Simple integration with Java and .NET.
Rules-based web applications.
Predictive analytics



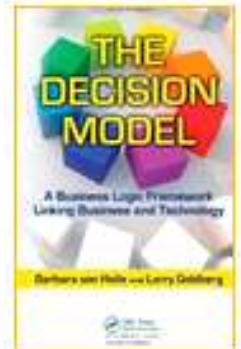
Key Differentiators:

Executable Decision Models.
No rule languages to learn.
No rules coding is required.
No IT involvement in representation





OpenRules 6 - BDMS



Major OpenRules® Functional Components



Rule Repository

Enterprise-class Repository
Maintained by Business Analysts
with Excel



Rule Engine

Executing Decision Models with
Inter-Related Rule Families



Rule Learner

Discovering Rules
Predictive Analytics



Rule Dialog

Developing dynamic rules-based Web
Questionnaires



Rule Solver

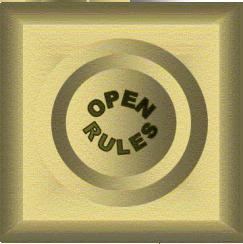
Solving Constraint Satisfaction
and Optimization Problems



Finite State Machines

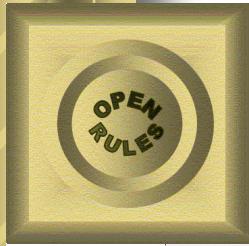
Event Processing
Connecting the Dots

● Business Rules for Business Analysts



Example “Hello Customer”

- Decide how to greet a particular customer during different times of the day (think IVR)
- Test:
 - Customer: Robinson is a married woman
 - Time of the day: 14:25 pm
 - Expected decision:
"Good Afternoon, Mrs. Robinson!"



Starting with a Decision

"Good Afternoon, Mrs. Robinson!"

Greeting

Salutation

Name

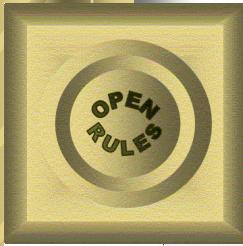
Decision DetermineCustomerGreeting

Decisions	Execute Rule Families
Define Greeting Word	<code>:= DefineGreeting()</code>
Define Salutation Word	<code>:= DefineSalutation()</code>



Rule Family “DefineGreeting”

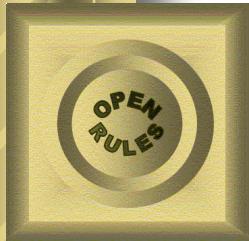
RuleFamily DefineGreeting					
Condition		Condition		Conclusion	
Hour of the Day		Hour of the Day		Greeting	
\geq	0	\leq	11	Is	Good Morning
>	11	\leq	17	Is	Good Afternoon
>	17	\leq	22	Is	Good Evening
>	22	\leq	24	Is	Good Night



Rule Family “DefineGreeting” (alternative representation)

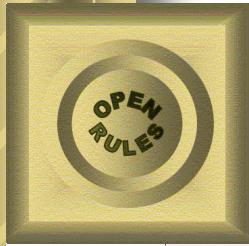
RuleFamily DefineGreeting	
If	Then
Hour of the Day	Greeting
0 - 11	Good Morning
11 - 17	Good Afternoon
17 - 22	Good Evening
22 - 24	Good Night

- We may turn on “The Decision Model Compliant” mode to prohibit this representation



Rule Family “DefineSalutation”

RuleFamily DefineSalutation					
Condition		Condition		Conclusion	
Gender		Marital Status		Salutation	
Is	Male			Is	Mr.
Is	Female	Is	Married	Is	Mrs.
Is	Female	Is	Single	Is	Ms.



Defining Business Glossary

Glossary glossary		
Fact Name	Business Concept	Attribute
Gender	Customer	gender
Marital Status		maritalStatus
Hour of the Day		hour
Greeting	Response	greeting
Salutation		salutation



Defining Test Data (in Excel)

Datatype Customer	
String	name
String	maritalStatus
String	gender
int	hour

Variable Customer customer			
name	maritalStatus	gender	hour
Customer Name	Marital Status	Gender	Hour
Robinson	Married	Female	14

Datatype Response	
String	greeting
String	salutation

Variable Response response	
greeting	salutation
Greeting	Salutation
?	?



Connecting Test Data to Glossary (Decision Objects)

DecisionObject decisionObjects

Business Concept	Business Object
Customer	<code>:= customer</code>
Response	<code>:= response</code>



Executing Decision Model

```
*** Decision: DetermineCustomerGreeting ***
Define Greeting Word
Conclusion: Greeting Is Good Afternoon
Define Salutation Word
Conclusion: Salutation Is Mrs.
```



DM Implementation Path

Decision DetermineCustomerGreeting

RuleFamily DefineGreeting

RuleFamily DefineSalutation

Glossary glossary

Fact Name	Business Concept	Attribute
-----------	------------------	-----------

Gender

*** Decision: DetermineCustomerGreeting ***

Define Greeting Word

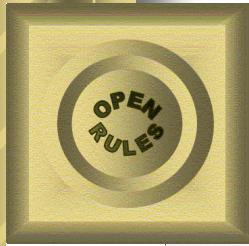
Conclusion: Greeting Is Good Afternoon

Custom Define Salutation Word

Conclusion: Salutation Is Mrs.

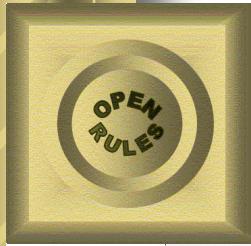
Customer := customer

Response := response



Integration with IT

- Let's assume that now our decision model is tested by business analysts
- How to integrate it with an actual IT application?
- No “mapping” to a target BRMS is required!



Accepting Data From Java (1)

```
public class Customer {  
    String name;  
    String maritalStatus;  
    String gender;  
    int hour;  
  
    // getters and setters  
}
```

```
public class Response {  
  
    String greeting;  
    String salutation;  
  
    // getters and setters  
}
```



Accepting Data From Java (2)

```
public static void main(String[] args) {

    String fileName = "file:rules/Decision.xls";
    Decision decision = new Decision("DetermineCustomerGreeting",fileName);
    Customer customer = new Customer();
    customer.setName("Robinson");
    customer.setGender("Female");
    customer.setMaritalStatus("Married");
    customer.setHour(Calendar.getInstance().get(Calendar.HOUR_OF_DAY));
    Response response = new Response();
    decision.put("customer", customer);
    decision.put("response", response);
    decision.execute();
    out.println("Decision: "
                + response.getGreeting()
                + ", " + response.getSalutation()
                + customer.getName() + "!");
}
```



Accepting Data From Java (3)

- Simply replace

DecisionObject decisionObjects	
Business Concept	Business Object
Customer	<code>:= customer</code>
Response	<code>:= response</code>

- with

DecisionObject decisionObjects	
Business Concept	Business Object
Customer	<code>:= decision.get("customer")</code>
Response	<code>:= decision.get("response")</code>



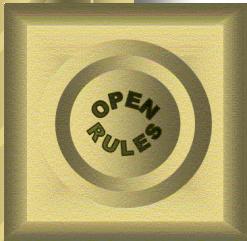
IT stays away of business logic

- So, the only integration points with IT were in these two tables:

Glossary glossary		
Fact Name	Business Concept	Attribute
Gender	Customer	gender
Marital Status		maritalStatus
Greeting	Response	greeting
Salutation		salutation
Current Hour		hour

DecisionObject decisionObjects	
Business Concept	Business Object
Customer	<code>= decision.get("customer")</code>
Response	<code>= decision.get("response")</code>

- Thus, IT never even looks at Rule Families!



You still may use Java...

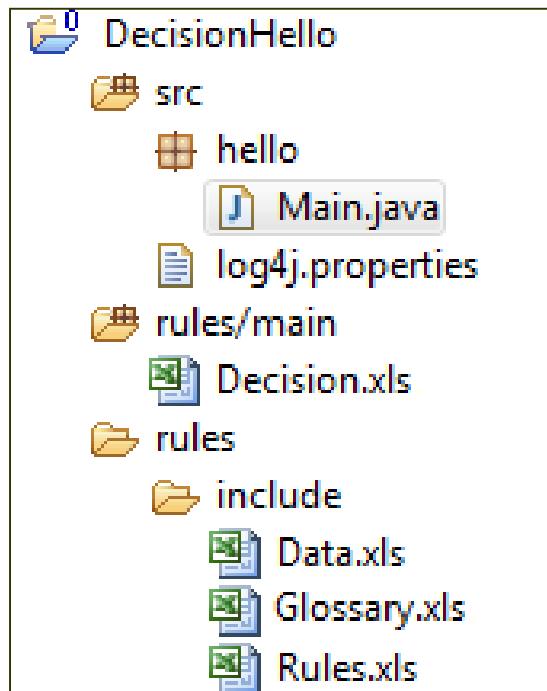
Decision DetermineCustomerGreeting

Decisions	Execute Rule Families
Define Greeting Word	<code>:= DefineGreeting()</code>
Define Salutation Word	<code>:= DefineSalutation()</code>
Display Decision	<code>{ System.out.println(getString("Greeting") + ", " + getString("Salutation") + customer.name + "!"); }</code>

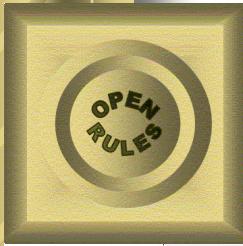
```
*** Decision: DetermineCustomerGreeting ***
Define Greeting Word
Conclusion: Greeting Is Good Morning
Define Salutation Word
Conclusion: Salutation Is Mrs.
Display Decision
Good Morning, Mrs.Robinson!
```



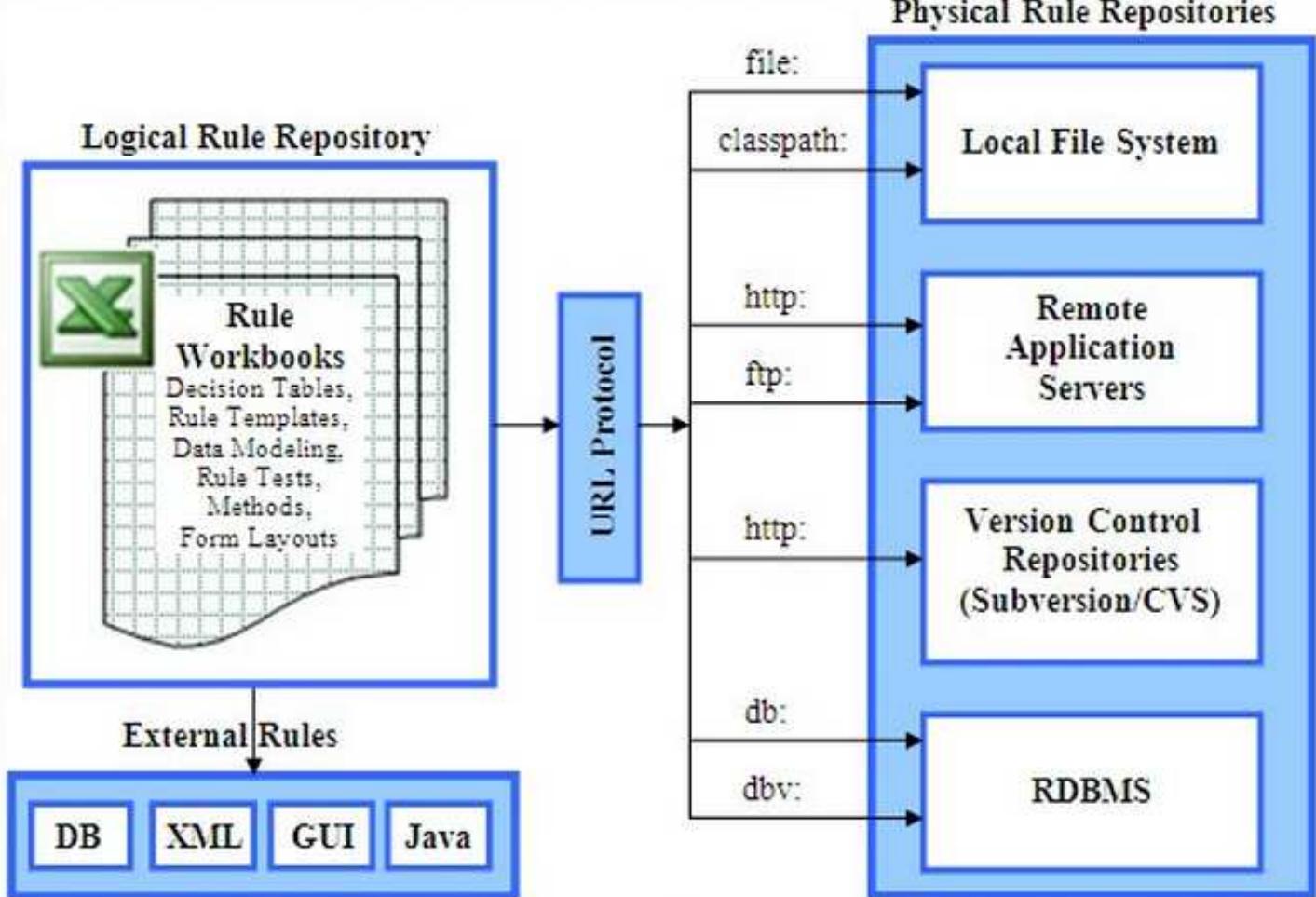
Rules Repository Structure

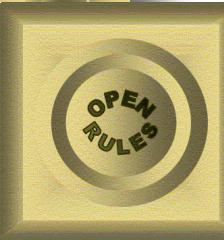


Environment	
include	./include/Rules.xls
	./include/Data.xls
	./include/Glossary.xls
	../../../../openrules.config/DecisionTemplates.xls



Rules Repository





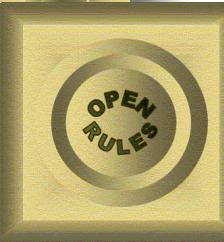
More Rule Families (Banking Example)

RuleFamily DefineCustomerProfile			
Condition		Conclusion	
Combined Balance		Customer Profile	
Within	0-500	Is	New
Within	500-2000	Is	Bronze
Within	2000-5000	Is	Silver
Within	5000-15000	Is	Gold
Within	15000-10000000	Is	Platinum



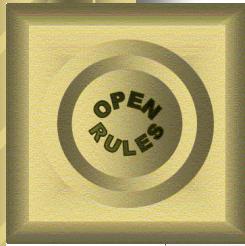
More Complex Rule Families

RuleFamily DefineUpSellProducts								
Condition		Condition		Condition		Conclusion		Message
Customer Profile		Customer Products		Customer Products		Offered Products		Set Comment
Is One Of	New,Bronze,Silver	Include	Checking Account	Do Not Include	Saving Account	Are	Saving Account, Debit/ATM Card, Web Banking	
Is One Of	New,Bronze,Silver	Include	Checking Account, Overdraft Protection	Do Not Include	CD with 25 basis point increase, Money Market Mutual Fund, Credit Card	Are	CD with 25 basis point increase, Money Market Mutual Fund, Credit Card	
Is One Of	New,Bronze,Silver	Include	Checking Account, Saving Account	Do Not Include	CD with 25 basis point increase, Money Market Mutual Fund, Credit Card	Are	CD with 50 basis point increase, Money Market Mutual Fund, Credit Card, Debit/ATM Card, Web Banking	
Is One Of	Gold	Include	Checking Account	Do Not Include	CD with 25 basis point increase, Money Market Mutual Fund, Web Banking	Are	CD with 50 basis point increase, Money Market Mutual Fund, Credit Card, Debit/ATM Card, Web Banking, Brokerage Account	Gold Package
Is One Of	Platinum	Include	Checking Account, Saving Account	Do Not Include	CD with 25 basis point increase, Money Market Mutual Fund, Web Banking	Are	CD with 50 basis point increase, Money Market Mutual Fund, Credit Card with no annual fee, Debit/ATM Card, Web Banking with no charge, Brokerage Account	Platinum Package



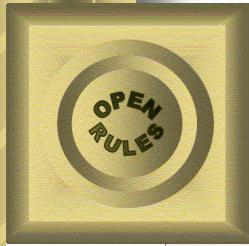
More Rule Families (Calculate Vacation Days)

RuleFamily DefineVacationDays		
If	If	Then
Age in Years	Years of Service	Vacation Days
<18		$::= (22 + 5)$
[18;45)	<15	$::= 22$
[18;45)	[15;30)	$::= (22 + 2)$
[18;45)	≥ 30	$::= (22 + 5 + 3)$
[45;60)	<15	$::= (22 + 2)$
[45;60)	[15,30)	$::= (22 + 2)$
[45;60)	≥ 30	$::= (22 + 5 + 3)$
60+		$::= (22 + 5 + 3)$



More Rule Families (Calculate Vacation Days)

Rules void DecisionTable1(Test t)									
C1	Age >=			18	45	60			
C2 Rules void DecisionTable2(Test t)									
C3 # Rules void DecisionTable3(Test t)									
C4 C1	Age	<18	[18;45]	[18;45]	[18;45]	[45;60]	[45;60]	[45;60]	60+
C2 A1	Service	<15	[15;30)	>=30	<15	[15,30)	30+		
A1 A2	Assign 22 days	X	X	X	X	X	X	X	X
A2 A3	5 extra days	X			X			X	X
A3 A4	2 extra days			X		X	X		
A4	3 extra days				X			X	X



Currently available operators

- <, Is Less, Is Less Than
- <=, Is Less Than or Equal To
- =, ==, Is
- !=, Is Not, Is Not Equal To, Not, Not Equal, Not Equal To
- >, Is More, Is Greater, Is More Than, Is Greater Than
- >=, Is More Or Equal To, Is Greater Or Equal To, ...
- Within, Interval
- Is One Of, Is One, Is One Of Many, Is Among, Among
- Include, Include All
- Exclude, Do Not Include, Exclude One Of
- Are

*Spaces and Upper/Lower cases are not important



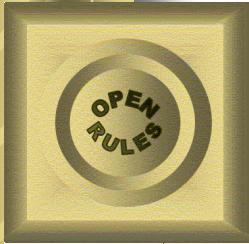
Examples of valid expressions

- > 0
- 5
- [5;10)
- [5;10]
- [5'000;10'000'000)
- [5,000;10,000,000)
- (5;100,000,000]
- [-100,-60)
- 5-10 // contains 5 and 10
- 5- 10
- -5 - 2
- -5 - -2
- from 5 to 20
- != 0
- <> 0
- less 5
- less than 5
- less or equals 5
- less or equals to 5
- smaller than 5
- more 10
- more than 10
- 10+
- >10
- >=10
- between 5 and 10
- no less than 10
- no more than 5
- equals to 5

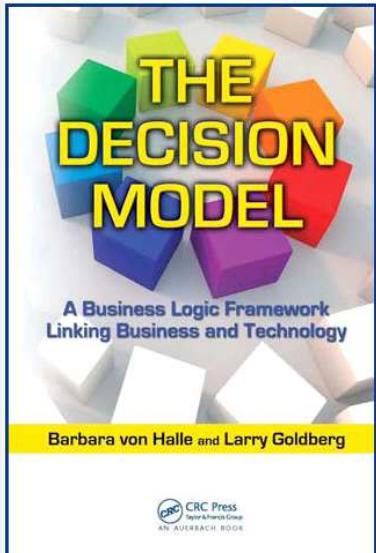


Examples of valid expressions

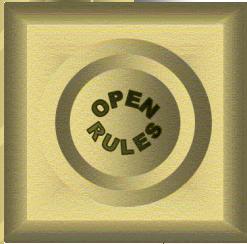
- greater or equal than 5 and less than 10
- not 10
- more than 5 less or equal than 10
- contains 10
- more than 5,111,111 less or equal than 10,222,222
- everything between 5,111,111 and 10,222,222
- Allowed combinations:
 - any words <number> any words <number>
 - Both integer and real numbers allowed
 - Integers can contain commas
 - Real numbers can include commas and digital points and a scientific form like 3.14e-5
- Arrays like “a, B, car, boat”
- and more



Executable Decision Models



- The Decision Model Primer
 - OpenRules-6 installation includes KPI's "The Decision Model Primer" that can be executed without coding
- Custom Decision Models
 - Completely configurable OpenRules templates allow a user to define what allowed and to add new custom features
- Built-in Integration with:
 - Rule Learner: Predictive Analytics
 - Rule Solver: Optimization
 - Rule Dialog: Questionnaire Builder
 - Rule Compressor



Key Differentiators

- Executable and Customizable Decision Models
- No Rule Languages to learn
 - no coding is required
- No proprietary GUI for rules management
 - business analysts create and execute decision models within Excel
- No IT involvement in business logic
- Easy to use and to integrate with Java/.NET