Scripting of Decisions

Mgr. Ivo Bek, Product Manager
Mgr. Marian Macik, Quality Engineer

April 2019

AGENDA

- ❏ Motivation
- ❏ Types of decisions
- ❏ DRL native rules versus DMN
- ❏ Uses of DMN
- ❏ DMN specification from OMG
- ❏ Demo

TYPES OF DECISIONS

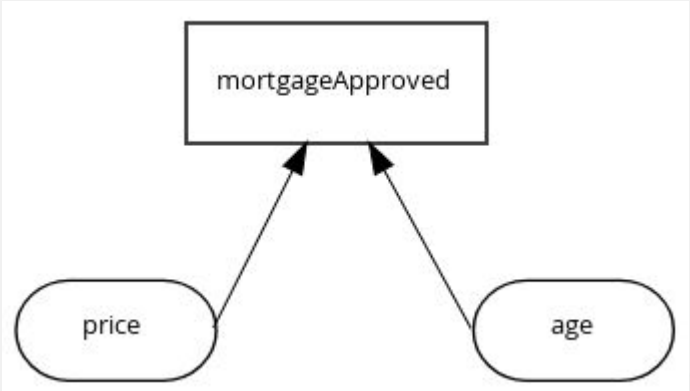
- Selection (Routing)
- Scoring
- Categorizing

- The decision doesn't take an action (no side effect), just determines a data value

DRL NATIVE RULES VERSUS DMN

- Rule scripting languages
- Generate rules
- Spreadsheets
- Decisions are tangible and more stable than individual rules

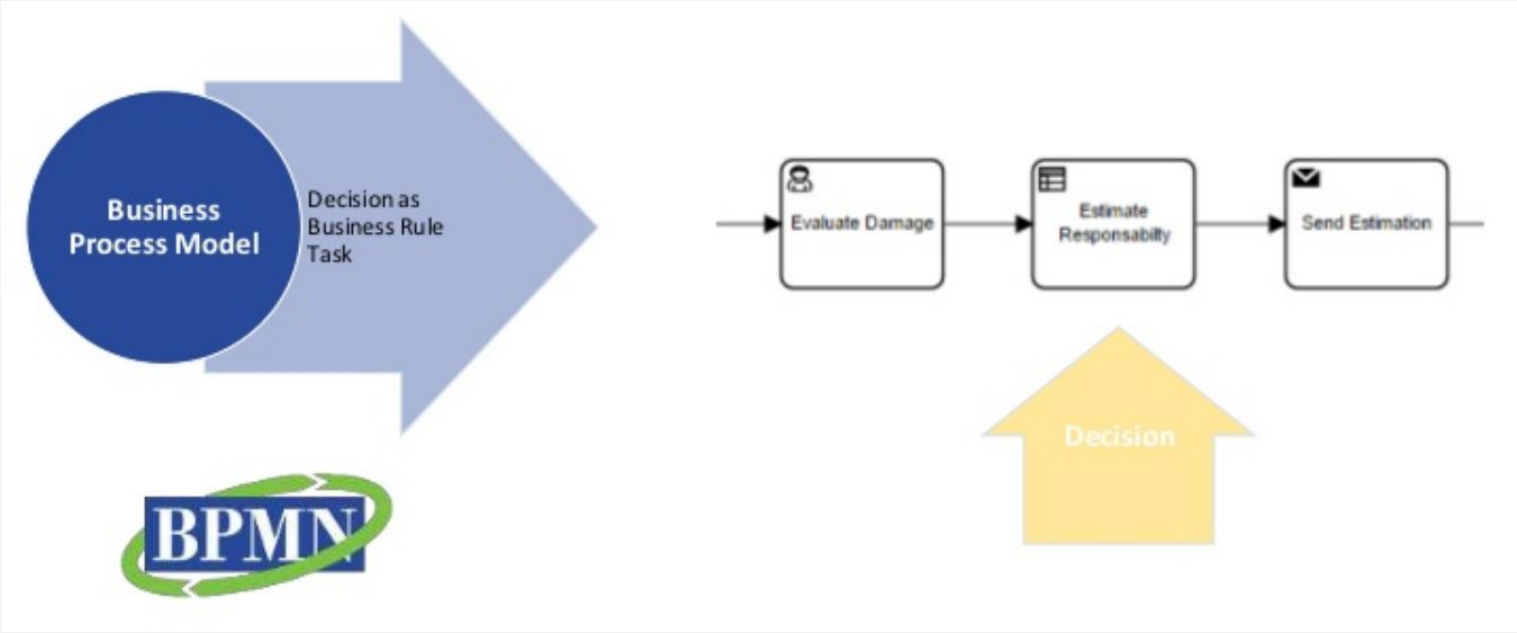
```
rule "Mortgage"
  when
    $price : Price(value < 10000)
    $person: Person(age > 25)
  then
    kcontext.setVariable(“mortgageApproved”, true);
end
```



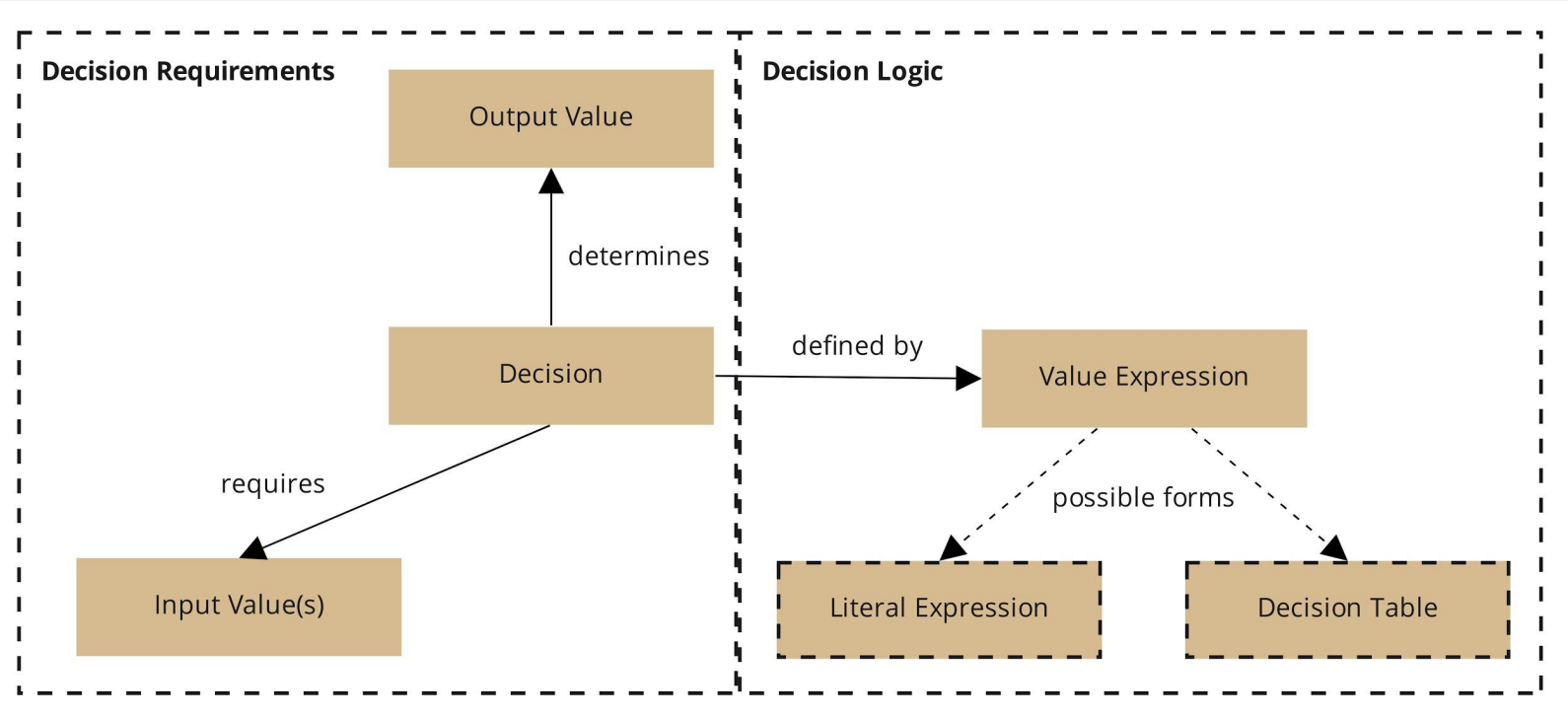
USES OF DMN

- Modeling human decision making
- Modeling requirements for decision making
- Executing decision logic

DECIDING AS AN ACTIVITY



FROM DECISION REQUIREMENT TO LOGIC



DECISION TABLE

Decision logic

- Rows = rules (most common)
- Hit policy determines what to do with rules that matches
  - Unique, First…

mortgageApproved *(Decision Table)*

F	price *(number)*	age *(number)*	mortgageApproved *(boolean)*	Description
1	<5000	-	true	Everybody can have 5000
2	<10000	> 25	true	Only age 26 and up can have up to 10000
3	-	-	false	

LITERAL EXPRESSION

Decision logic

- FEEL
- DMN FEEL Cheat Sheet

DEMO
Decision-driven Business Process

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

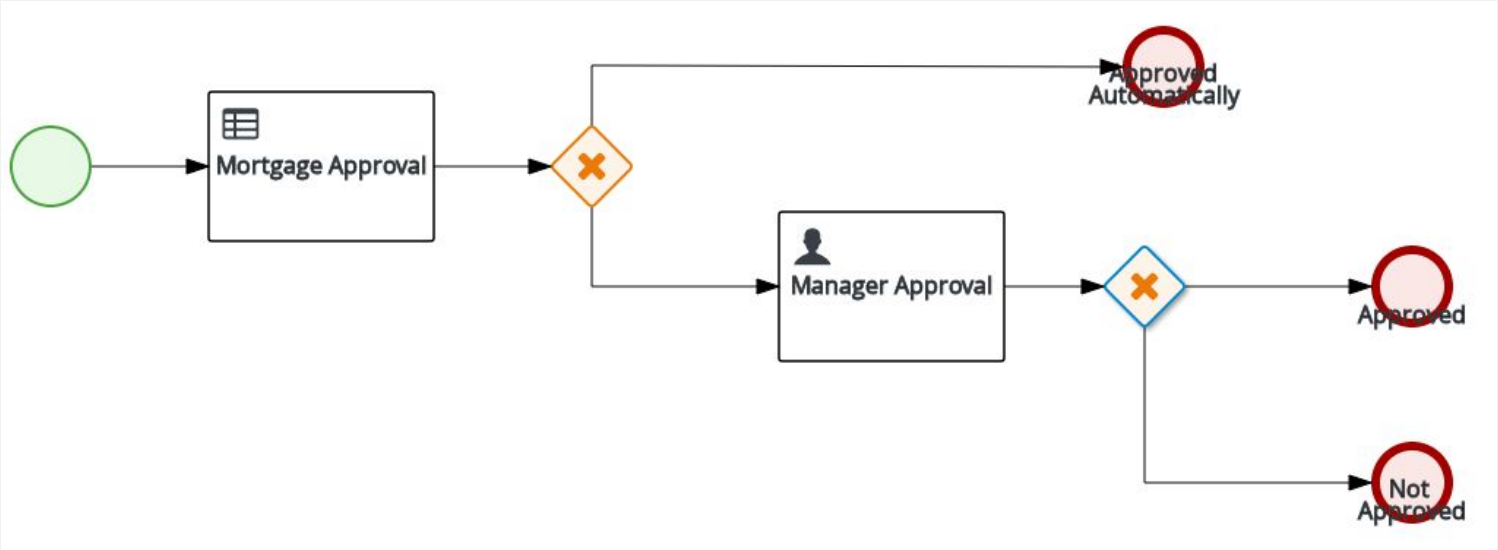
twitter.com/RedHatNews

youtube.com/user/RedHatVideos

LABS
DMN inside Business Process Forms

The Process

- In a new project, design the following process:



Data Object Person

- Create this type using Data Object asset.
- Add field **age** of type **Integer** to it.

Process variables

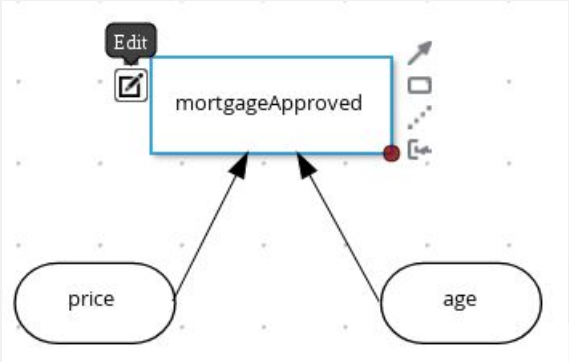
- Create 3 process variables:
  - **price** of type **Integer**
  - **approved** of type **Boolean**
  - **person** of type **Person**

DMN

mortgageApproved *(Decision Table)*

F	price *(number)*	age *(number)*	mortgageApproved *(boolean)*	Description
1	<5000	-	true	Everybody can have 5000
2	<10000	> 25	true	Only age 26 and up can have up to 10000
3	-	-	false	

- Create a DMN asset.
- Add 2 DMN Input Data nodes:
  - **price** of type **number**
  - **age** of type **number**
- Add **mortgageApproved** DMN Decision node of type boolean and click on a small edit icon near the node.
- Click on **Select expression** and choose Decision Table. Configure it like on the picture.
- Save your DMN.

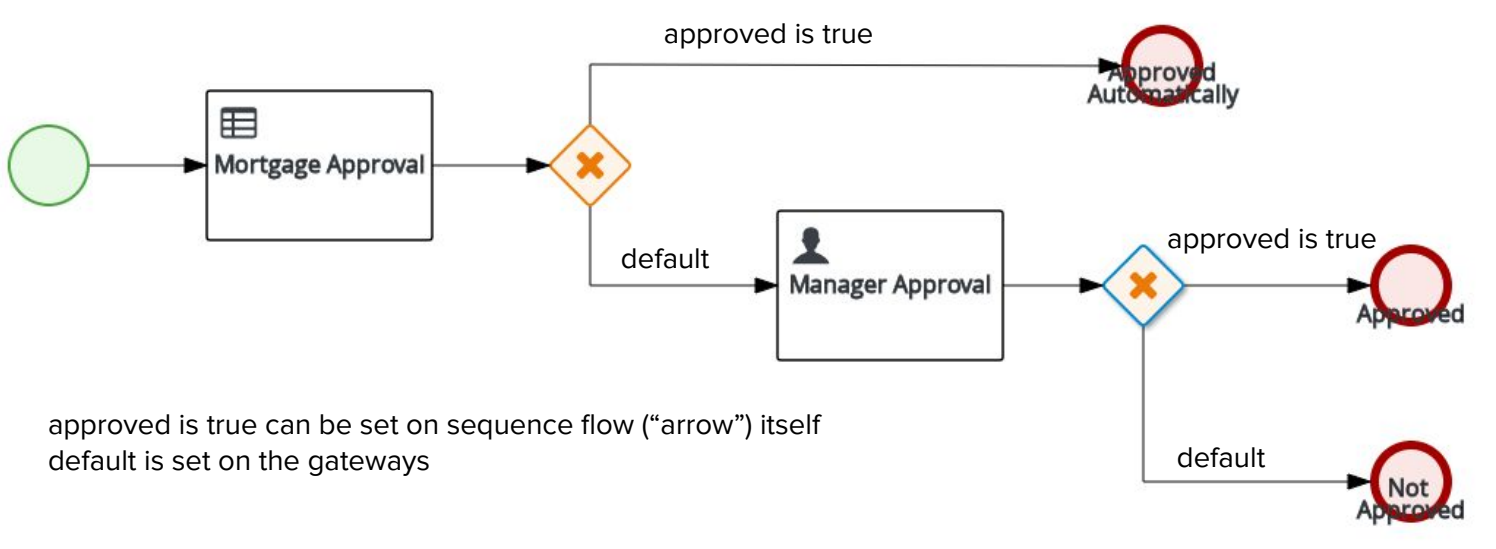


Mortgage Approval Business Rule Task

- Select **DMN** as **Rule Language**.
- Fill in the namespace, it can be found when clicking on DMN canvas and then on Diagram properties.
- Decision Name is **mortgageApproved**.
- DMN Model Name is the name of your DMN file (without .dmn extension).
- Configure data assignments:
  - Inputs:
    - **price** of type **Integer** from source **price**.
    - **age** of type **Integer** from source **#{person.age}**.
  - Outputs:
    - **mortgageApproved** of type **Boolean** with target **approved**.

Gateways

- Configure the gateways as on the picture:



Manager Approval Human Task

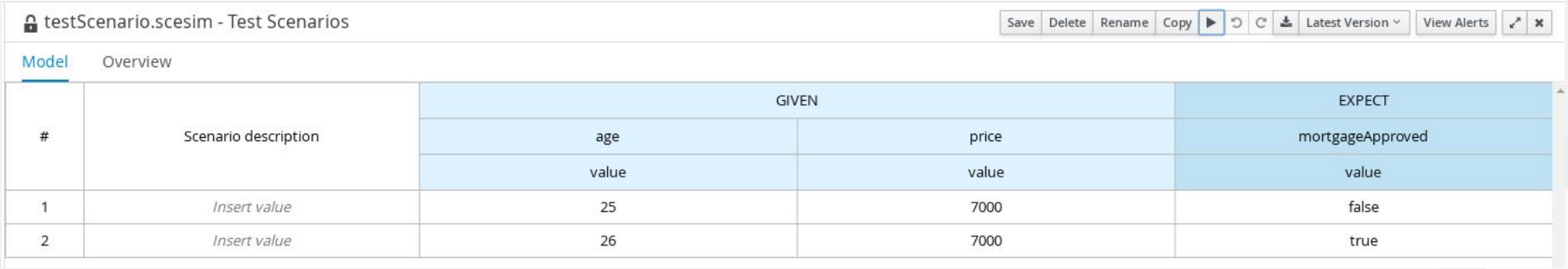
- Configure data assignments:
  - Inputs:
    - **price** of type **Integer** from source **price**.
    - **person** of type **Person** from source **person**.
  - Outputs:
    - **approved** of type **Boolean** with target **approved**.
- Assign the task to the actor **wbadmin**.

Run the Process

- Congratulations! You can now run your process and verify that it works.
- If you have spare time, you can try to fine-tune the forms:
  - Process form shouldn't have **approved** field visible.
  - **price** and **age** should be required fields.
  - Manager Approval task can have **price** and **age** as input (data assignment) which would mean they will be available for manager to review when completing the task. After you changed the assignment, regenerate forms to propagate all changes you made. Be sure that these two fields are read-only as a manager can only change the **approved** field.

Test Scenario

- If you want, you can try creating **Test Scenario** asset.
  - Add Asset of type **Test Scenario**.
  - Select **Source type** of DMN.
  - Choose your DMN asset.
- Fill your test data and expected results.
- Run your test scenario using the play button.



#	Scenario description	GIVEN		EXPECT
		age	price	mortgageApproved
		value	value	value
1	*Insert value*	25	7000	false
2	*Insert value*	26	7000	true