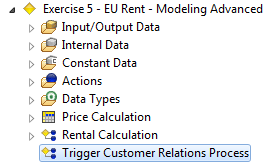
Exercise 5: Ruleflow

**New requirements:**

The current rule model should to be restructured. The performing steps for customer relations are supposed to be separated from the remaining logic. So, the decision if the customer gets a loyalty present or not is to be outsourced to a new rule. Additionally, the logic for identifying the best customer should be calculated differently.

**Steps:**

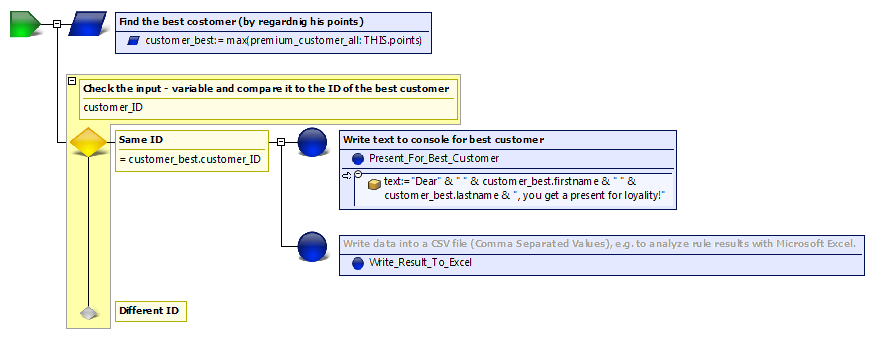
1. Create a new flow rule Trigger Customer Relations Process.
2. Remove all modeling elements from rule Rental Calculation referring to the customer relations process (element identifying the loyalty present for the best customer) and add these elements to the newly created rule Trigger Customer Relations Process. Be reminded, that input-/output data elements, internal data elements, constant data elements, action data elements and data structures have to be moved to the level of the newly created rule, as well (add these elements to the just mentioned level for a first approach). Be aware of appearing errors referring to the structural changes and try to resolve them carefully.



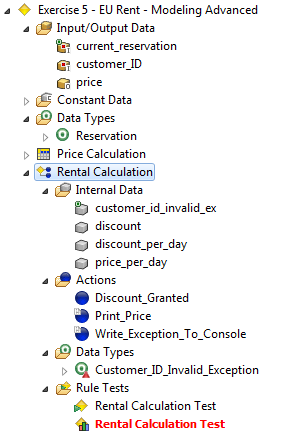
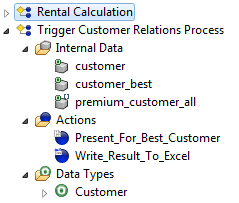
1. Now, try to get the data elements well structured within your rule model: not all data elements need to be on the highest level of the rule project. Decide which of the elements will be needed for both rules or will just be needed locally for one rule (always referring to the “best practices”) and move them accordingly.
2. Call rule Trigger Customer Relations Process out of rule Rental Calculation.
3. Execute the rule and analyze its result.

Result: Modeling of EU RENT

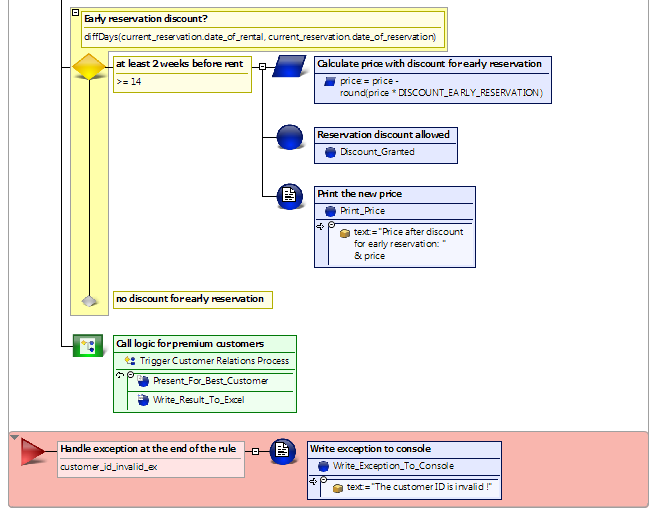
New rule Trigger Customer Relations Process:



Business model:

Rule Rental Calculation:



Questions:

1. Which of the following statements are suiting for element “Call Flow Rule”?

|  |  |  |
| --- | --- | --- |
| a | Calling other rules is possible |  |
| b | Calling other rule models is possible |  |
| c | Calling rules of other rule models is possible |  |
| d | Allows project – structuring |  |
| e | Allows reuse of rules |  |

2. Which of the following statements are suiting for element “Done & Return”?

|  |  |  |
| --- | --- | --- |
| a | Always finishes the execution of all rules |  |
| b | Always finishes the execution of the actual rule |  |
| c | Finishes the execution of all rules (in case that the actual rule hasn’t been called out of another rule) |  |
| d | Restarts execution of rule code going out of the calling rule |  |

3. Name two different repeating – methods, which are possible for element “Repeat”.

Results:

1. Which of the following statements are suiting for element “Call Flow Rule”?

|  |  |  |
| --- | --- | --- |
| a | Calling other rules is possible | x |
| b | Calling other rule models is possible |  |
| c | Calling rules of other rule models is possible | x |
| d | Allows project – structuring | x |
| e | Allows reuse of rules | x |

2. Which of the following statements are suiting for element “Done & Return”?

|  |  |  |
| --- | --- | --- |
| a | Always finishes the execution of all rules |  |
| b | Always finishes the execution of the actual rule | x |
| c | Finishes the execution of all rules (in case that the actual rule hasn’t been called out of another rule) | x |
| d | Restarts execution of rule code going out of the calling rule | x |

3. Name two different repeating – methods, which are possible for element “Repeat”.

* Repeat for each of multiple elements
* Repeat while a condition holds
* Repeat with counter