Meta-Modelling as a Concept:
The Conceptualisation of Modelling Methods

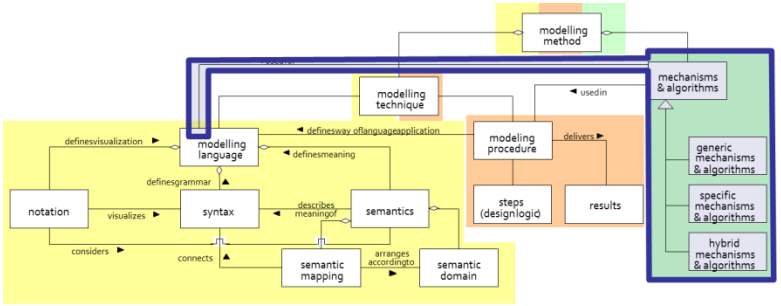
Nesat Efendioglu

Training Specific Scenarios

Selected Scenarios for Training specific Hands-On:

1. Realising a **Modelling Language**
   - Case: Entity Relationship Model

2. Implementing an **Algorithm**
   - Case: Structural Similarities of Business Processes



Analysis of Structural Similarities

2. SCENARIO: IMPLEMENTING AN ALGORITHM

Scenario Description

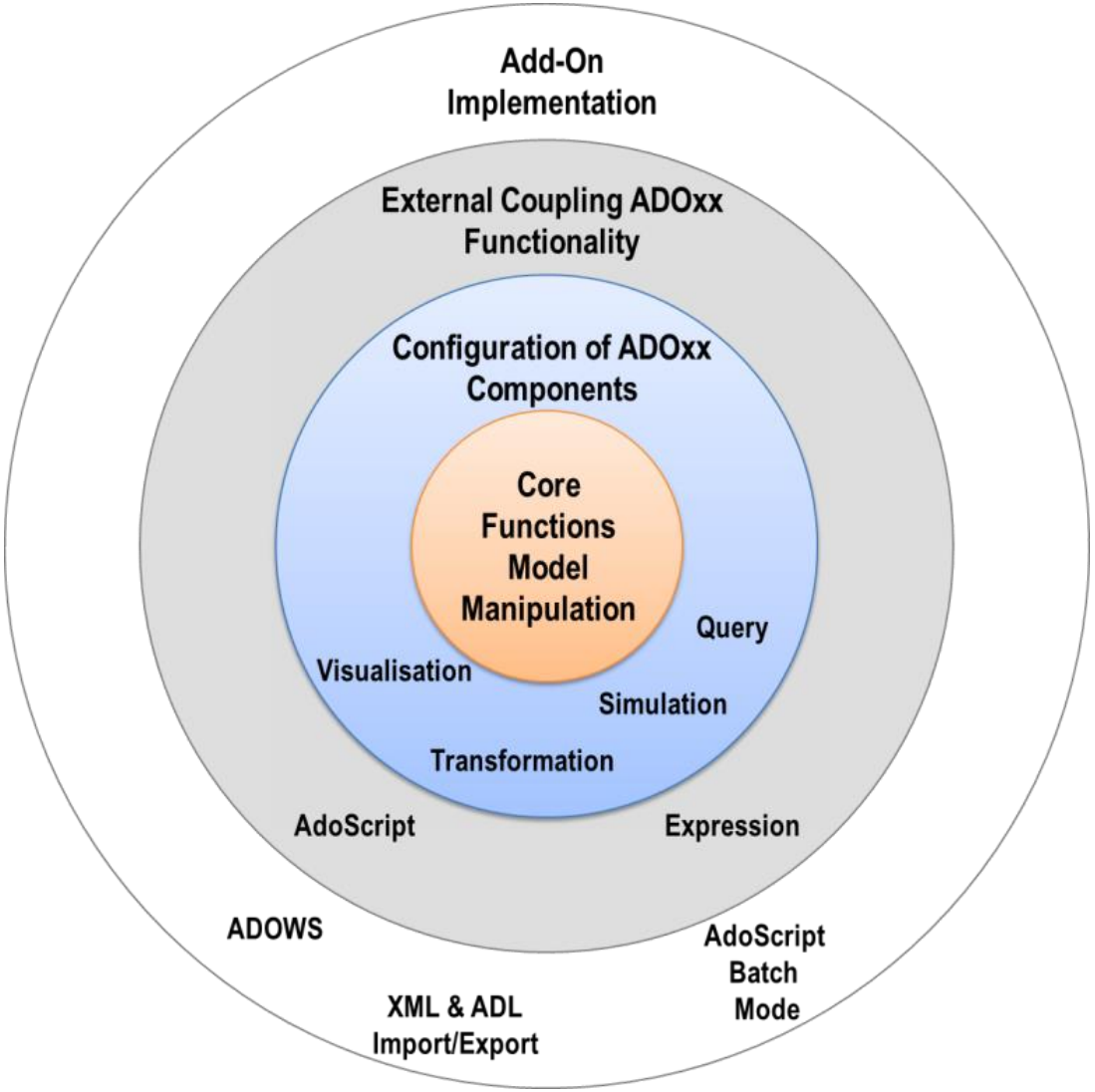
Case:

An algorithm for analysing structural similarities is implemented that queries business process models and creates a comparison matrix listing structural similarities.

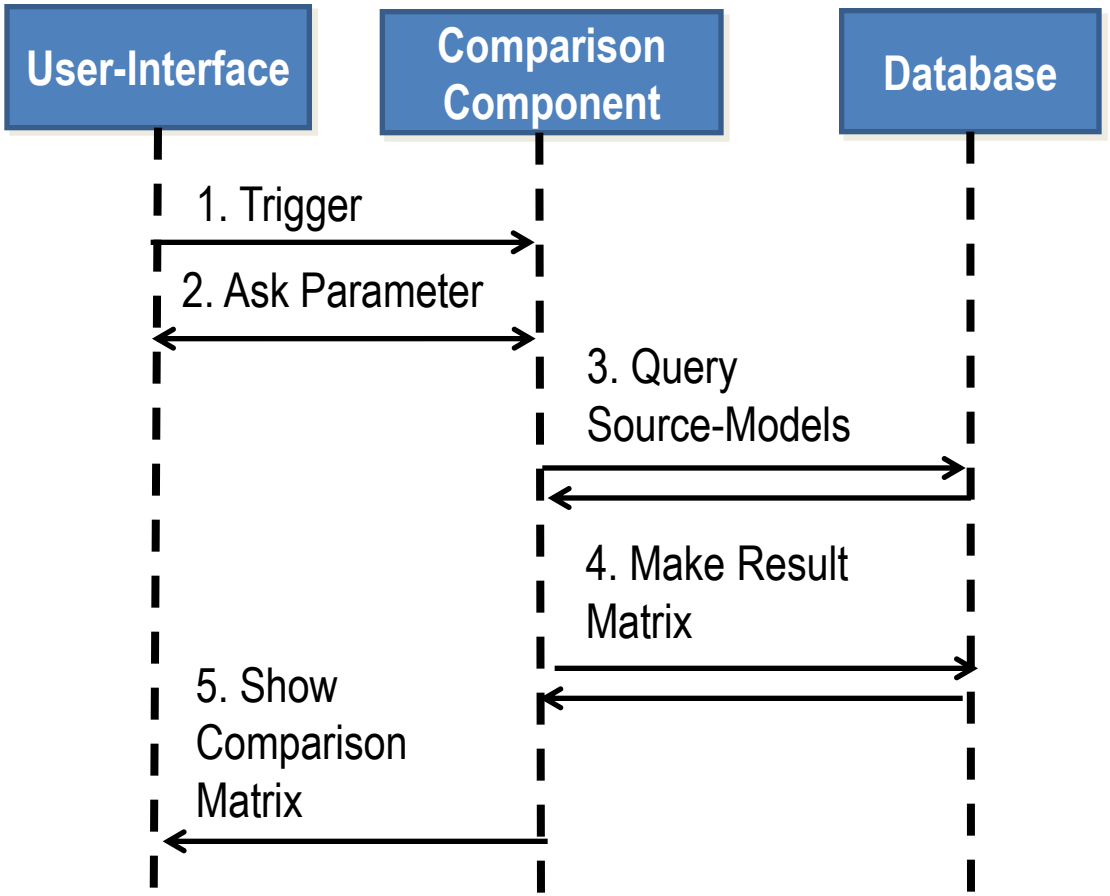
GOAL:

Demonstrate how models can be queried with AQL and ADOscripts, as well as indicate how to create and manipulate a model.

ADOxx Functionality on Meta Level



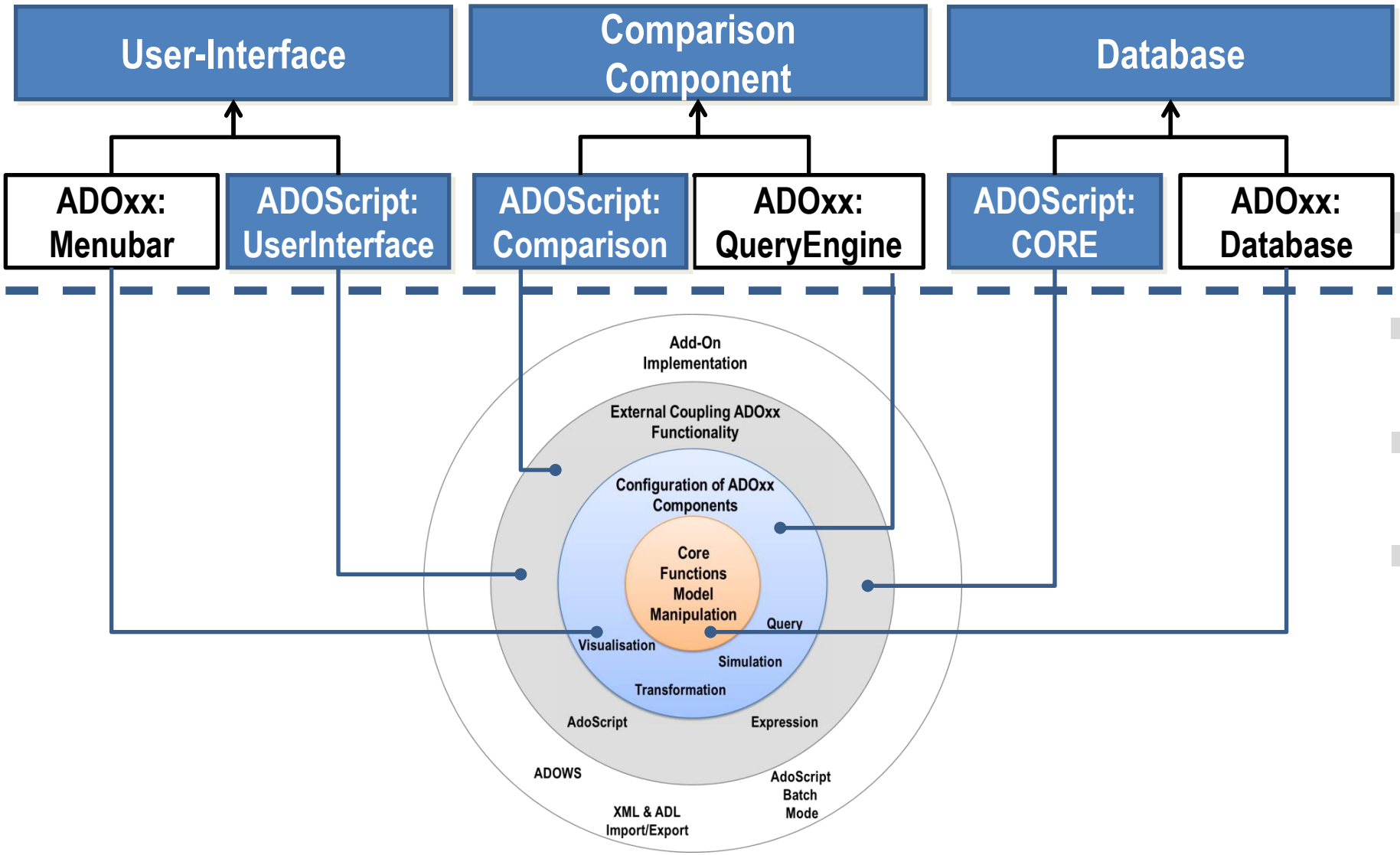
Description of Algorithm



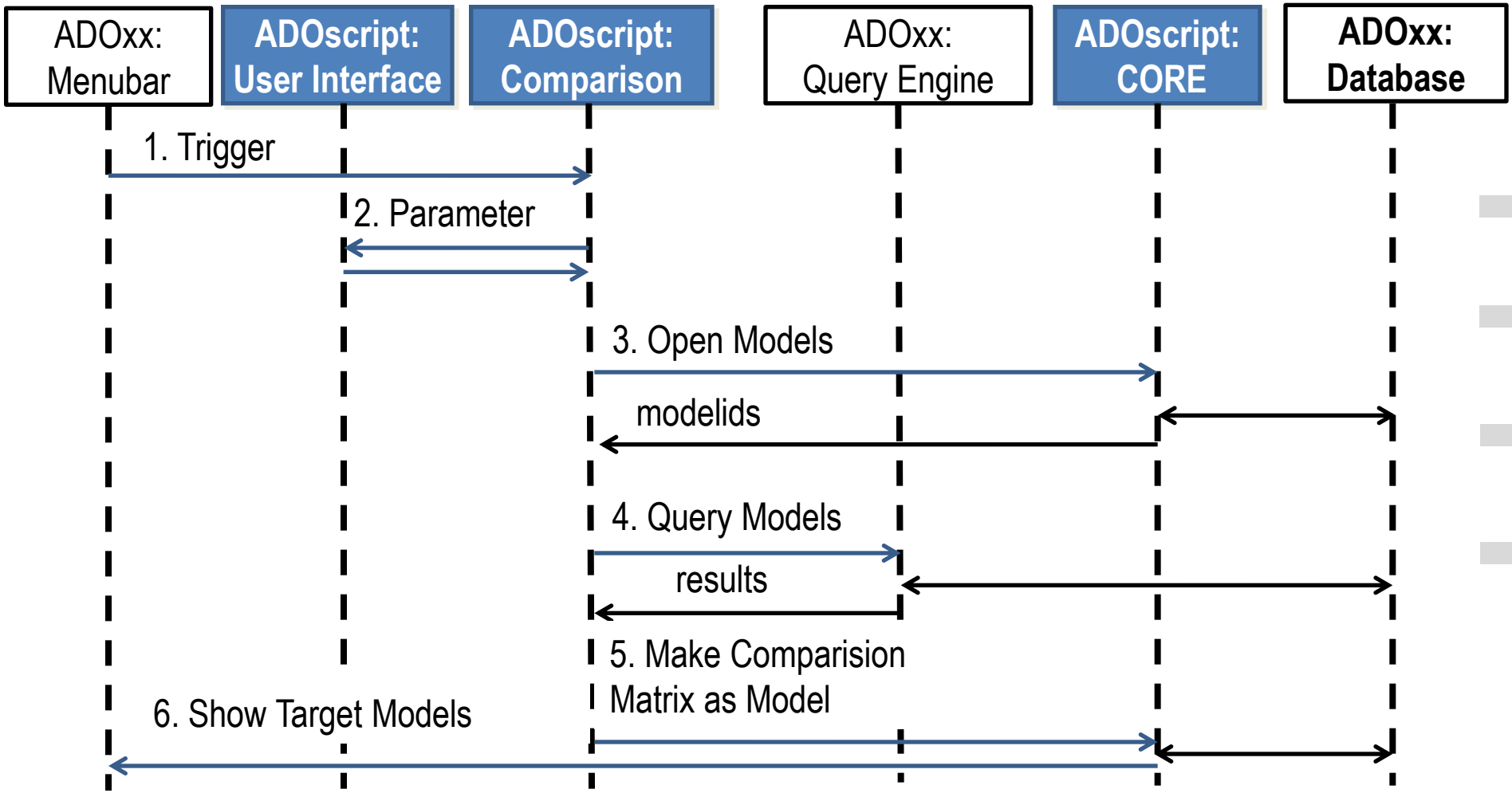
Additional Aspects:

- Implementation as plug-In to be used in other modelling languages.
- Comparison queries should be adaptable but start with comparing used objects.
- Migration from modelling language without plug-In to modelling language with plug-In has to be considered.

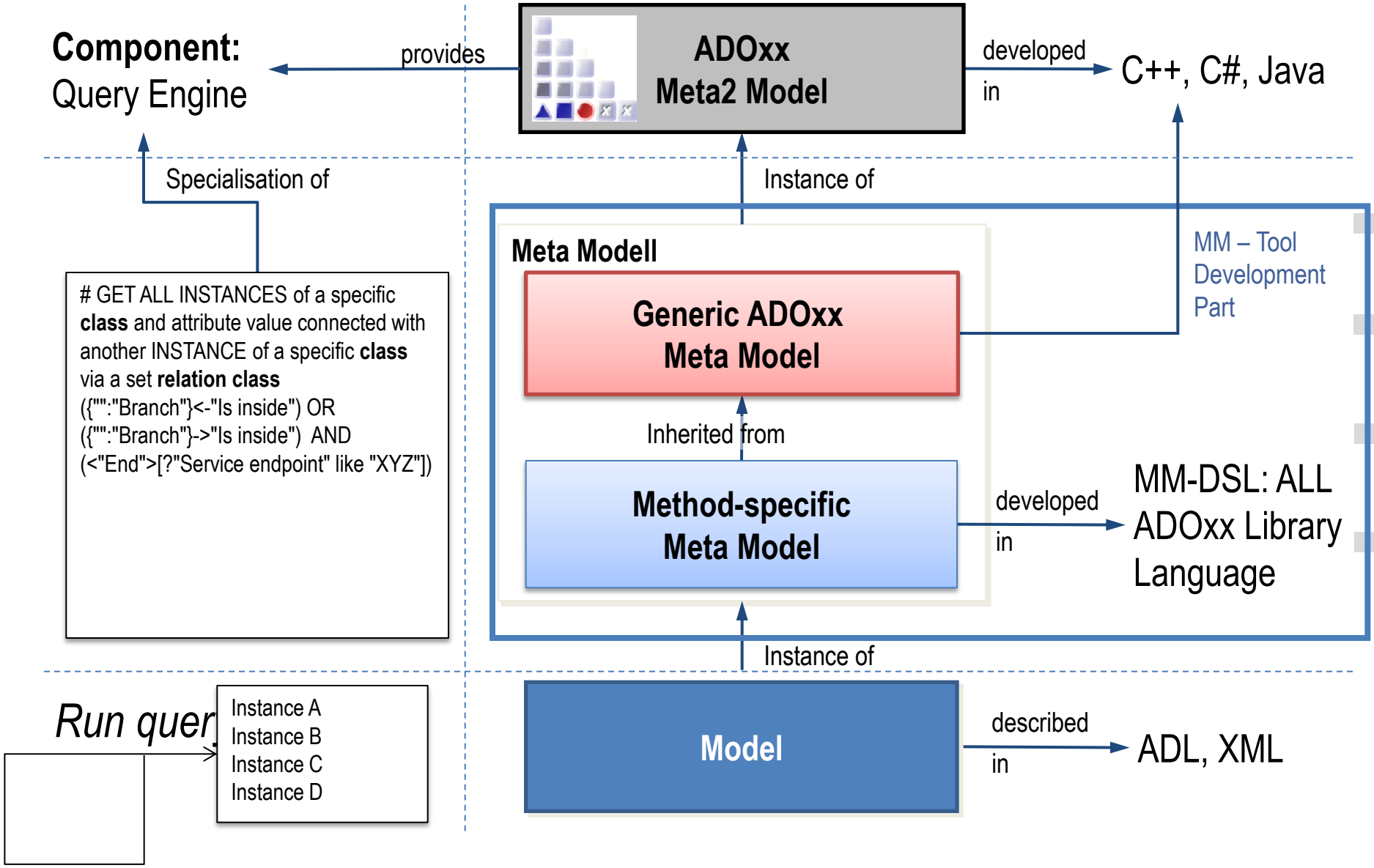
Mapping ADOxx Functionality



ADOxx Realisation Approach



Meta Modelling Layer: Implementing and Algorithm



Added Value of Metamodelling Platform

Used meta-modelling functionality for realisation of the scenario:

- **ADOScrip:** ADOscript can generate a new model “Comparison Model” to present the results of the business process comparison. This technique can also be used for graph-rewriting.

- **AQL: ADOxx Query Language**
  - **ADOxx query engine** is provided by the platform and can analyze business process models.
  - **ADOScripts** can invoke the query engine and hence compare in a stepwise approach business processes

- **Hyperlinks and INTERREF:** Similar to the first scenario, the resulting model can use **INTERREFS** and **Hyperlink** for better navigation from the resulting “Comparison Matrix” to the originally compared business processes.

ADOxx Realisation Hands-On

1. **Modelling Language Extensions to enable this algorithms**
    1. New model type “Comparison Model”
    2. New class “Box”, “Row Name” for Comparison Matrix Element

2. **Configure ADOxx**
    1. Configure Menubar
    2. Write AQL statements for query engine

3. **Implement Algorithm with ADOscript**
    1. ADOscript User Interface
    2. Invoking query engine with ADOscript
    3. Create target model “Comparison Matrix and place matrix elements according the results of the query.

Used ADOxx Functionality: Implementing an Algorithm

- Introduction
- Setup of Implementation Environment

- Modelling Language Implementation
  - Classes
  - Relations
  - Class Attributes and Attributes
    - GRAPHREP
    - ATTRREP
    - CLASS Cardinality
    - CONVERSION
    - Model Pointer
  - Attribute Facets
  - Model Types

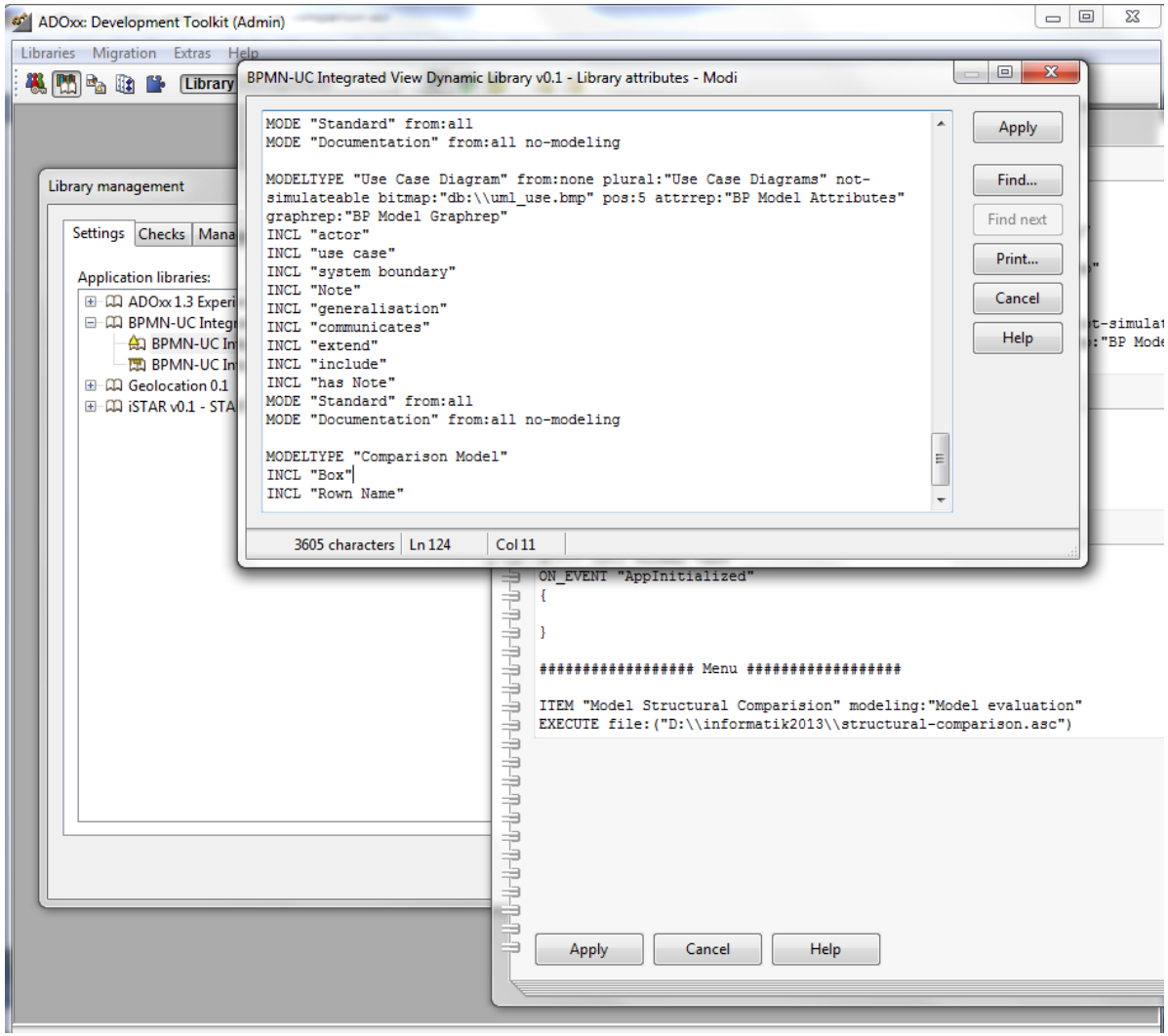
- Mechanisms & Algorithms Implementation
  - Core Functions for Model Manipulation
    - Database
    - Visualisation
    - **Query**
    - Transformation
  - Configuration of ADOxx Components
    - Visualisation
    - Query
  - **External Coupling ADOxx Functionality**
    - **ADOscript Triggers**
    - **ADOscript Language Constructs**
    - Visualisation ADOscript
    - Visualisation Expression
    - **Query ADOscript**
    - Transformation ADOscript
  - ADD-ON Implementation
    - ADOxx Web-Service
    - XML / ADL Import – Export
    - ADOscriptBatch Mode

HANDS-ON

Analysis of Structural Similarities

2. SCENARIO: IMPLEMENTING AN ALGORITHM

Define new Modeltype „Comparison Matrix“

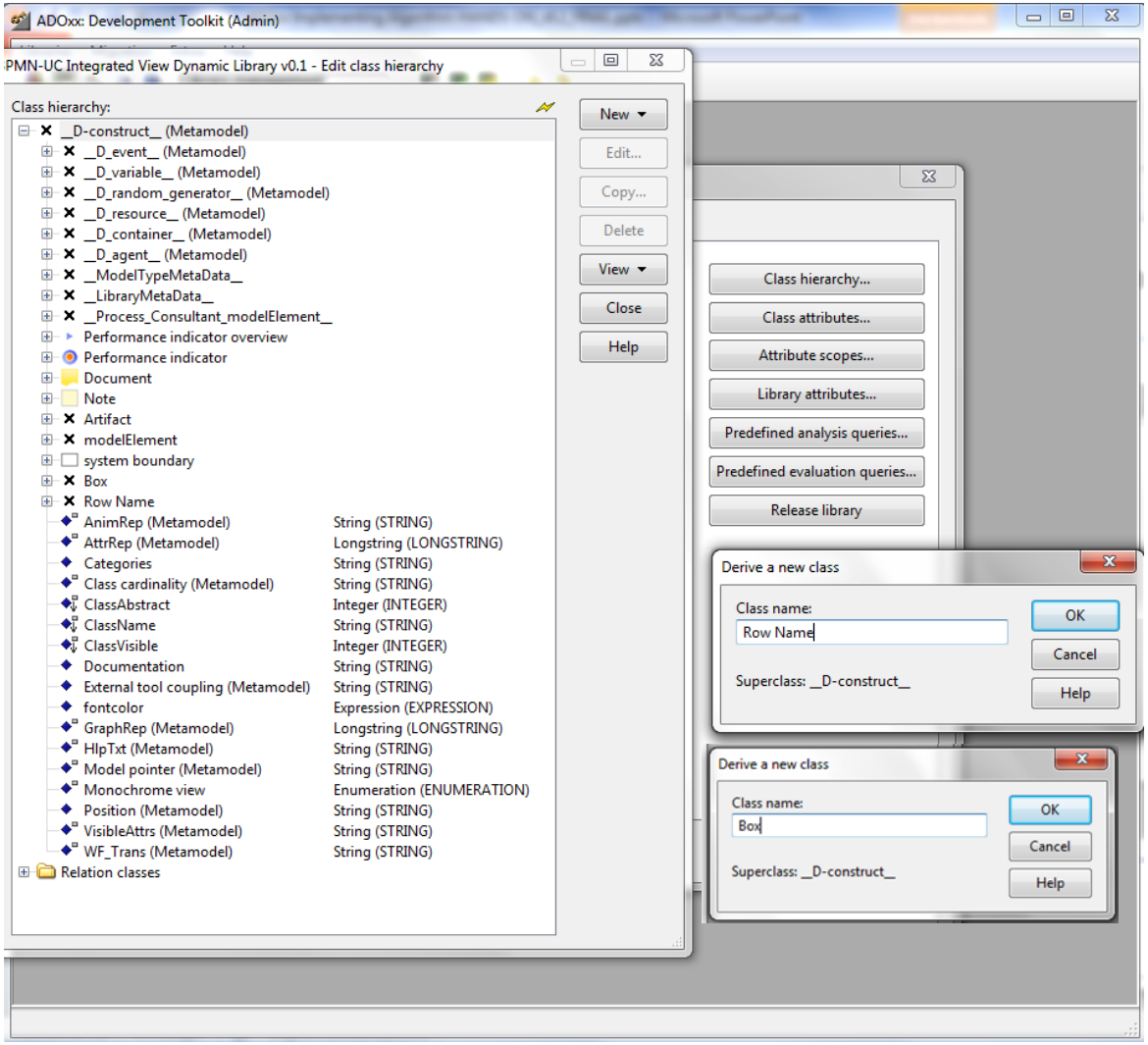


New Modeltype:

- Select “BPMN-UC Integrated View Dynamic Library” and open Library attributes.
- Got to Add Ons
- Add the Modeltype “Comparison Model” in the Modi attribute
- When the classes are defined, you need to INCLUDE “Box” and “Row name”

MODELTYPE "Comparison Model"
INCL Box
INCL Row name

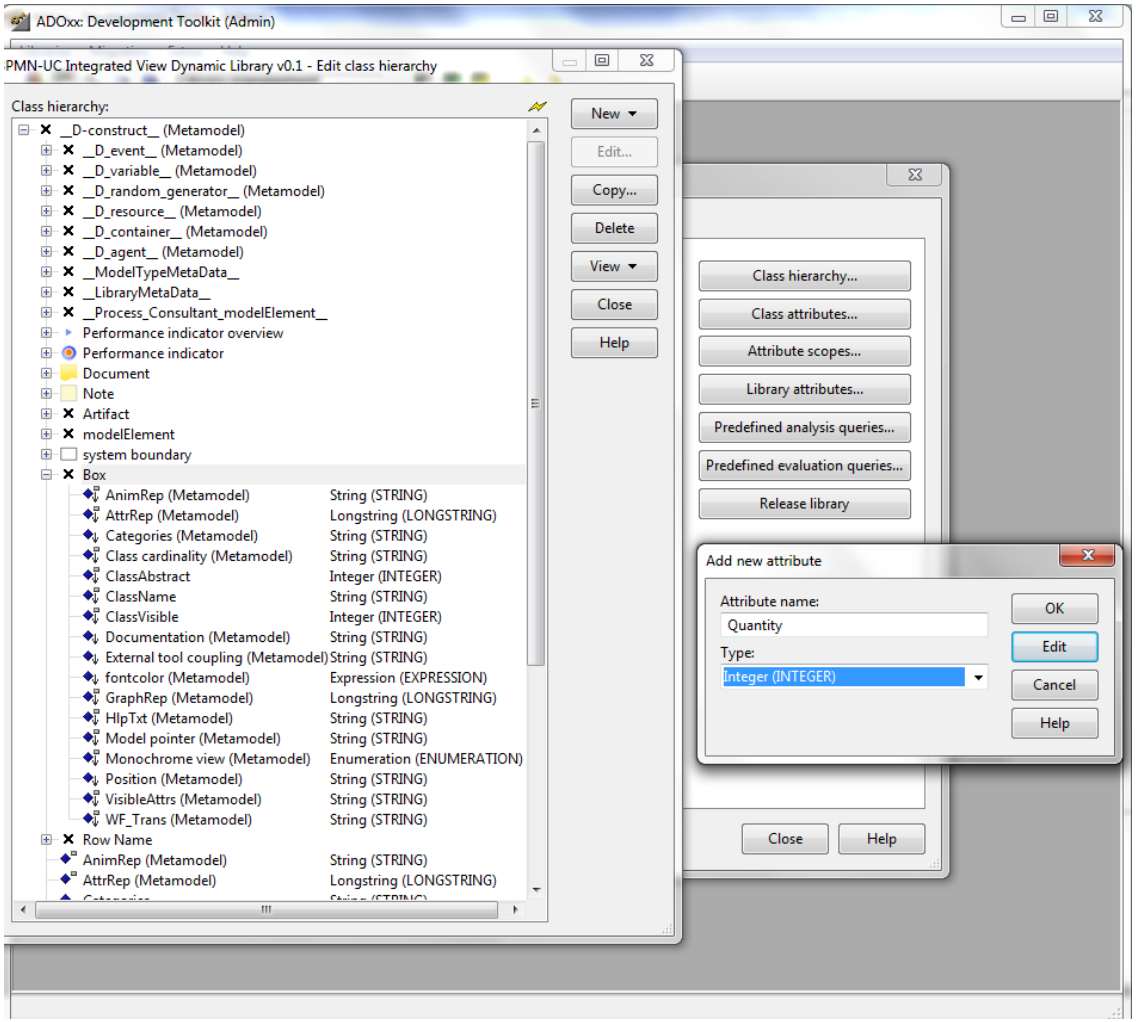
Make New Modeltype



New Modeltype:

- Select “BPMN-UC Integrated View Dynamic Library” and open Library attributes.
- Open Class hierarchy, view “Metamodel” and “Class hierarchy” in the View button, select __D-construct__ and click new class.
- Name new classes: “Box” and “Row Name”
- Box and Row Name are now sub-classes of __D-construct__

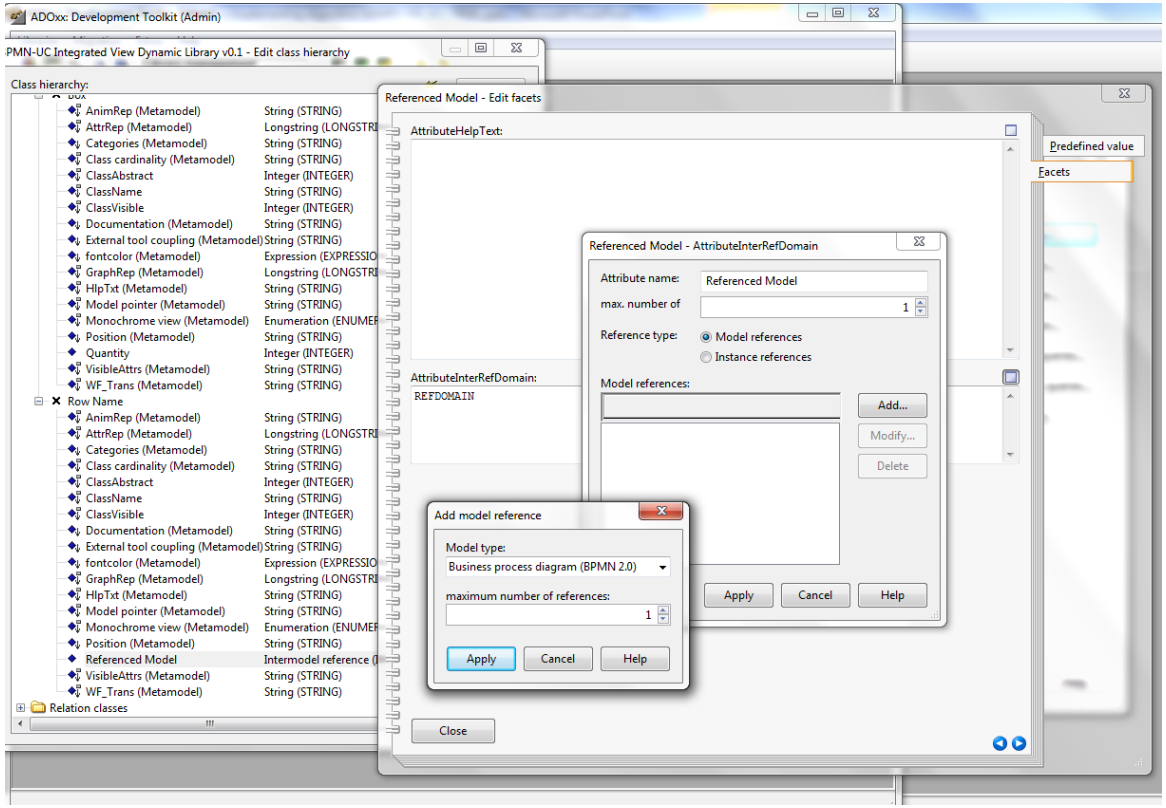
Add Attributes for Classes „Box“ and „Row Name“



Add Attributes

- Select “Box” and click New, attribute.
- Make “Quantity” as type INTEGER.
- Select “Row Name” and click New, attribute.
- Make “Referenced model” an INTERREF to target modeltype “BPMN”
- Make “Row name” a STRING.

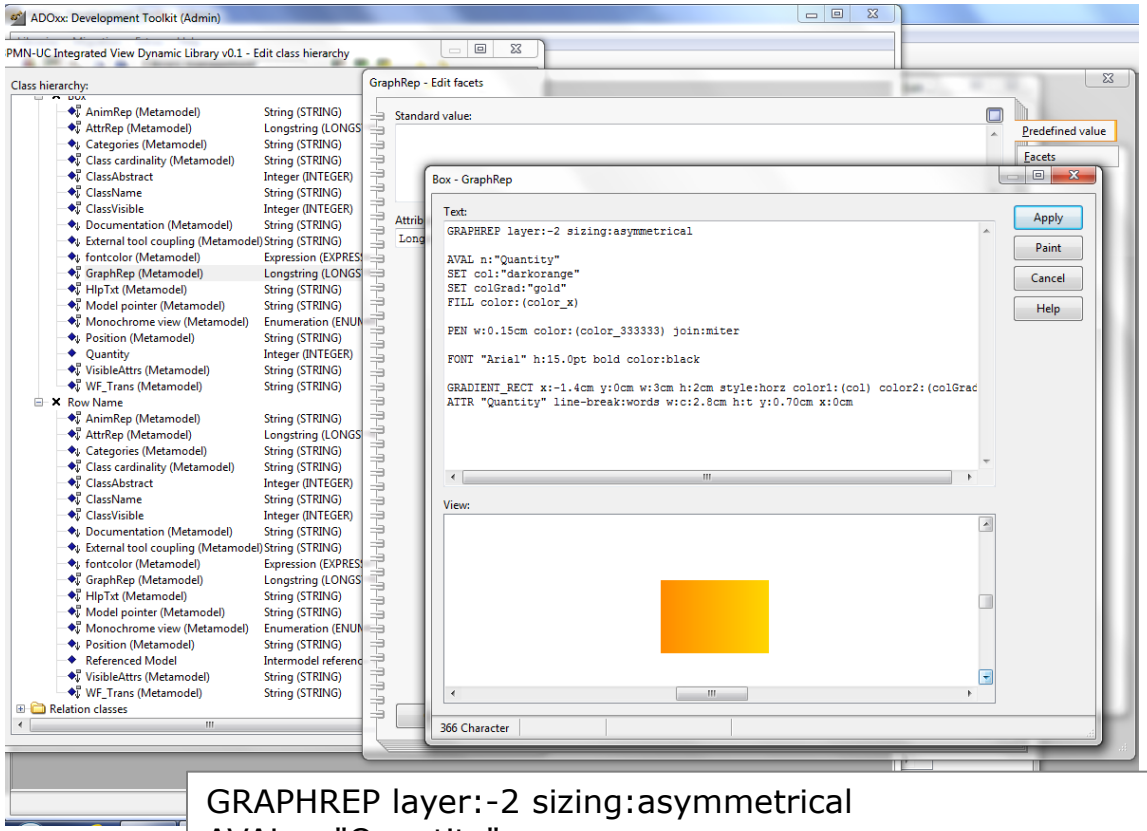
Edit INTERREF



Specification of INTERREF

- EDIT Facet
- Select AttributeInterrefDomain
- Select “Model reference”
- Max number of references is 1
- Select Business Process Diagram
- Max number of references is 1

Add GRAPHREP



Specification of GRAPHREP

- Select “Box”
- Click on Attribute “GraphRep”
- Open the GraphRep Editor
- Enter text, paint it and apply.

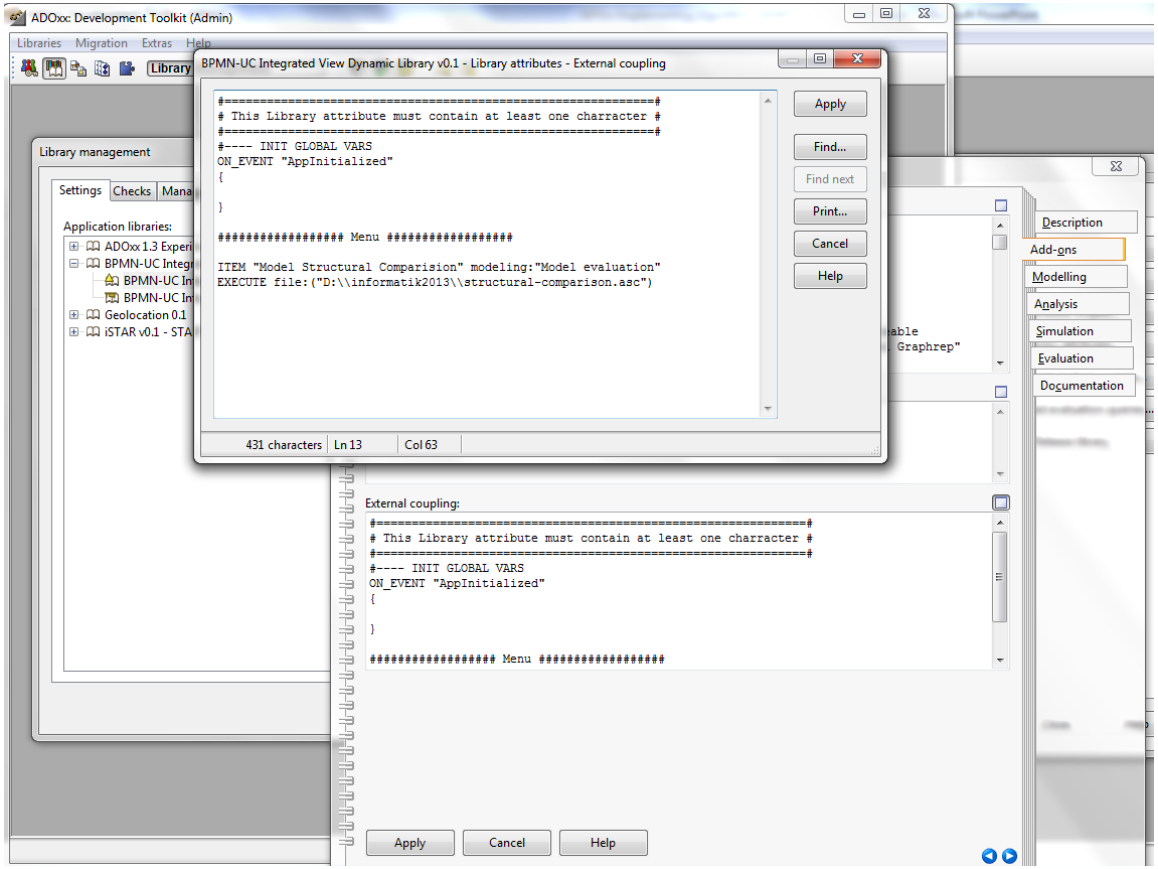
```
GRAPHREP layer:-2 sizing:asymmetrical
AVAL n:"Quantity"
SET col:"darkorange"
SET colGrad:"gold"
FILL color:(color_x)
PEN w:0.15cm color:(color_333333) join:miter
FONT "Arial" h:15.0pt bold color:black
GRADIENT_RECT x:-1.4cm y:0cm w:3cm h:2cm style:horz color1:(col) color2:(colGrad)
ATTR "Quantity" line-break:words w:c:2.8cm h:t y:0.70cm x:0cm
```

GRAPHREP – Box & Row name

```
GRAPHREP layer:-2 sizing:asymmetrical

AVAL n:"Quantity"
SET col:"darkorange"
SET colGrad:"gold„

FILL color:(color_x)

PEN w:0.15cm color:(color_333333) join:miter
FONT "Arial" h:15.0pt bold color:black

GRADIENT_RECT x:-1.4cm y:0cm w:3cm h:2cm style:horz color1:(col) color2:(colGrad)
ATTR "Quantity" line-break:words w:c:2.8cm h:t y:0.70cm x:0cm
```

```
GRAPHREP
FONT "Arial" h:10pt bold color:black

AVAL reference:"Referenced model"
AVAL rowname:"Row name"

IF(LEN reference > 0)
    ATTR "Referenced model" line-break:words x:-1.4cm y:0.75cm w:c:2.8cm h:c:1.5cm format:"%m"
ELSIF (LEN rowname > 0)
    ATTR "Row name" line-break:words x:-1.4cm y:0.75cm w:c:2.8cm h:c:1.4cm
ELSE
    ATTR "Name" line-break:words x:-1.4cm y:0.75cm w:c:2.8cm h:c:1.4cm
ENDIF
```

Add Menubar



Add Menubar

- Select Dynamic Library.
- Open Library Attributes
- Select Add-On
- Open External Coupling
- Add Menubar in External Coupling

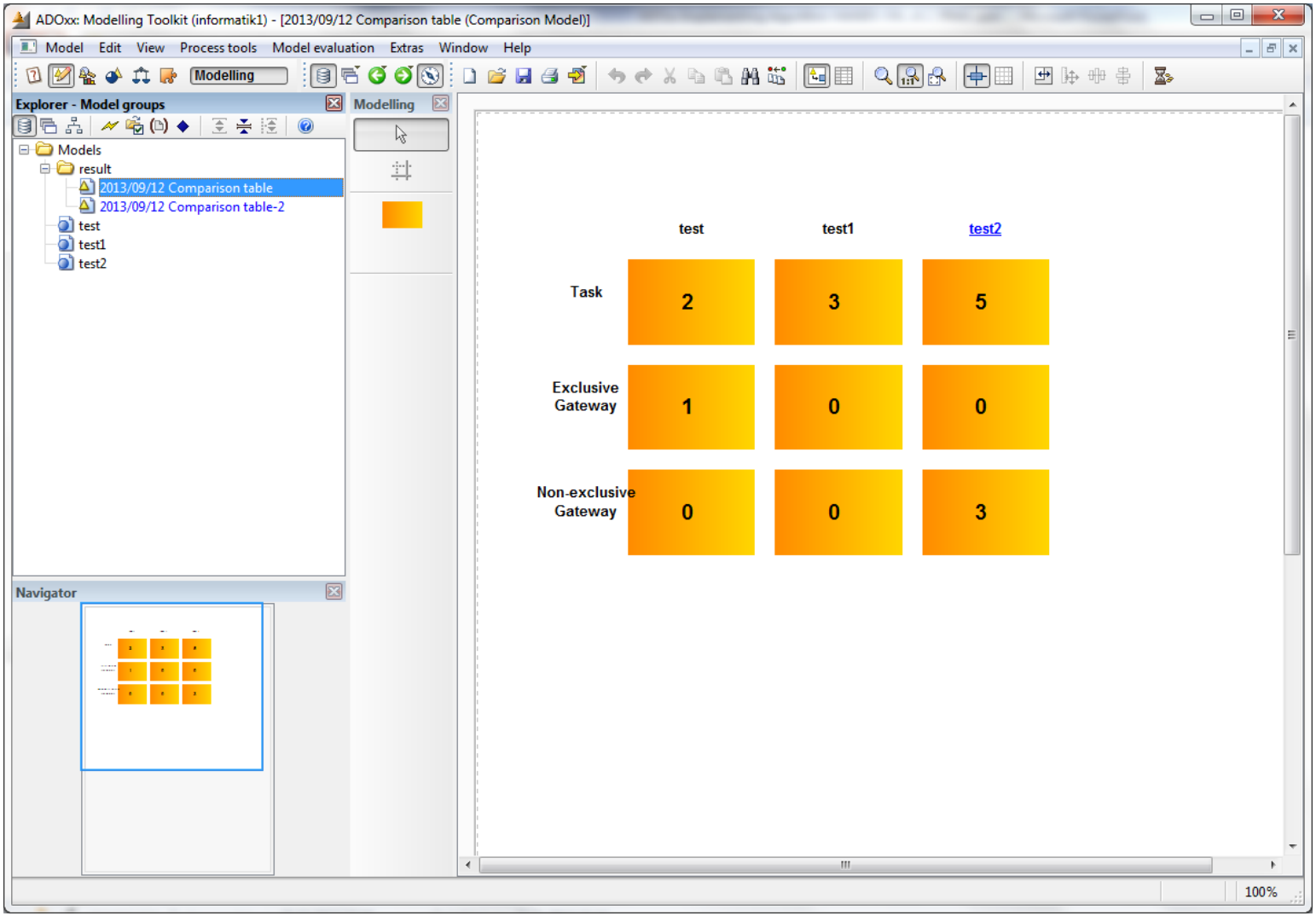
################## Menu ##################

ITEM "Model Structural Comparision" **modeling**:"Model evaluation"
EXECUTE file:("D:\\informatik2013\\structural-comparison.asc")

Copy and Configure ADOscript

```
##########################################
# Structural Comparision                 #
##########################################

#----------------------------------------
# Parameter setup
#----------------------------------------

SETL strtkn_element:"Task,Exclusive Gateway,Non-exclusive Gateway,X"
SETL aqltkn_statements:"(<\"Task\">)@(<\"Exclusive Gateway\">)@(<\"Non-exclusive Gateway\">)"
SETL int_cnt_elements:(tokcnt((strtkn_element),","))

SETL str_modeltype-1:"Business process diagram (BPMN 2.0)"
SETL str_modeltype_name:"Comparison Model"

#----------------------------------------
# Source Model and Target Model selection
#----------------------------------------

…
```

Result


ADOxx: Modelling Toolkit (informatik1) - [2013/09/12 Comparison table (Comparison Model)]
Model Edit View Process tools Model evaluation Extras Window Help
Modelling
Explorer - Model groups
Models
result
2013/09/12 Comparison table
2013/09/12 Comparison table-2
test
test1
test2
Modelling
Navigator
test
test1
test2
Task
2
3
5
Exclusive Gateway
1
0
0
Non-exclusive Gateway
0
0
3
100%

Any Questions?