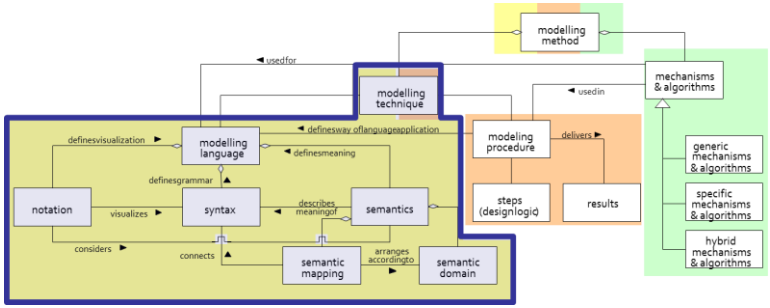


CASE: BIVEE Virtual Enterprise Model

1. SCENARIO: REALISING A MODELLING LANGUAGE

Scenario Description

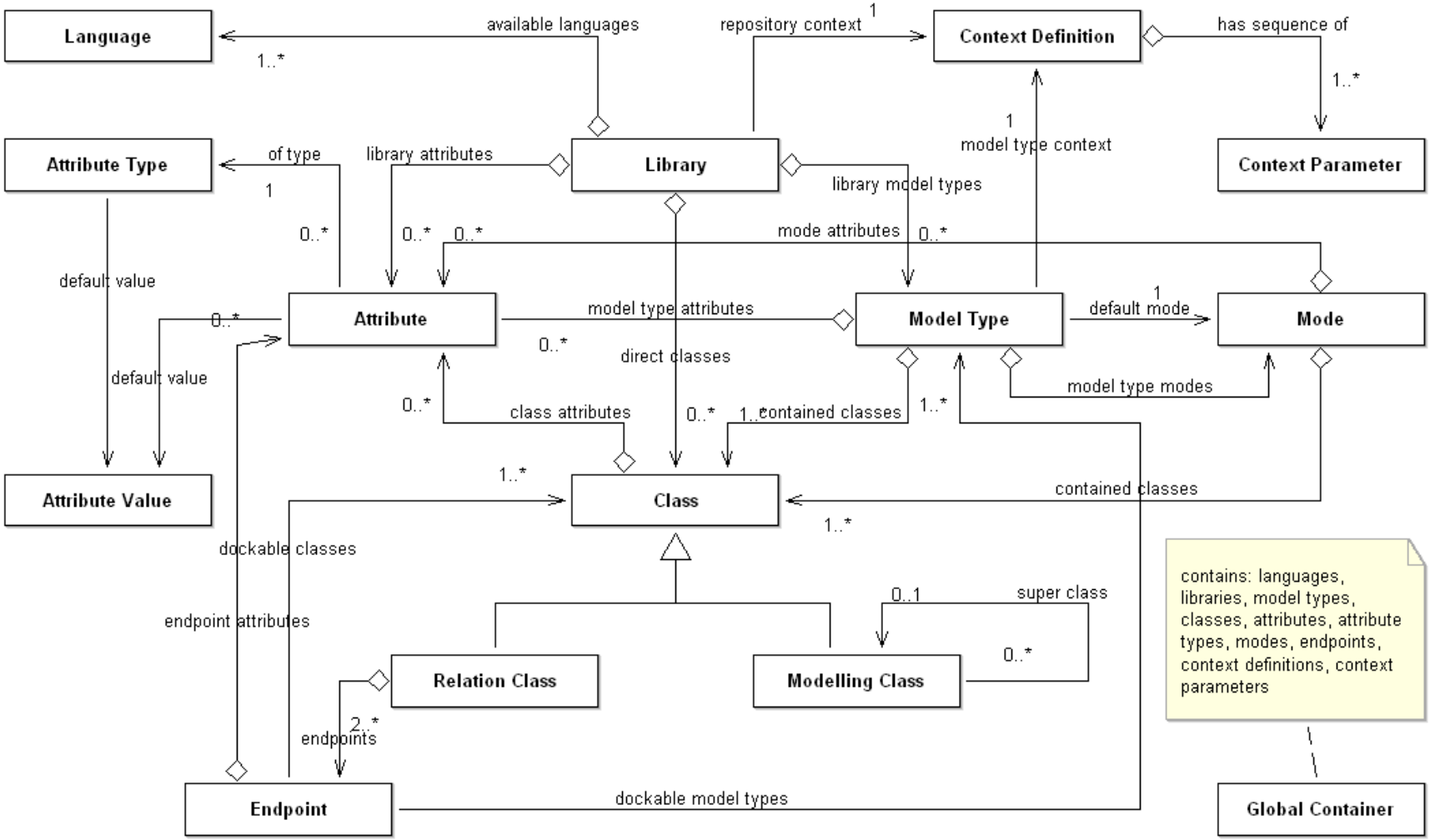
Goal:

Demonstrate the development of a model editor for a defined modelling languages using common constructs from ADOxx Meta^{2}Model.

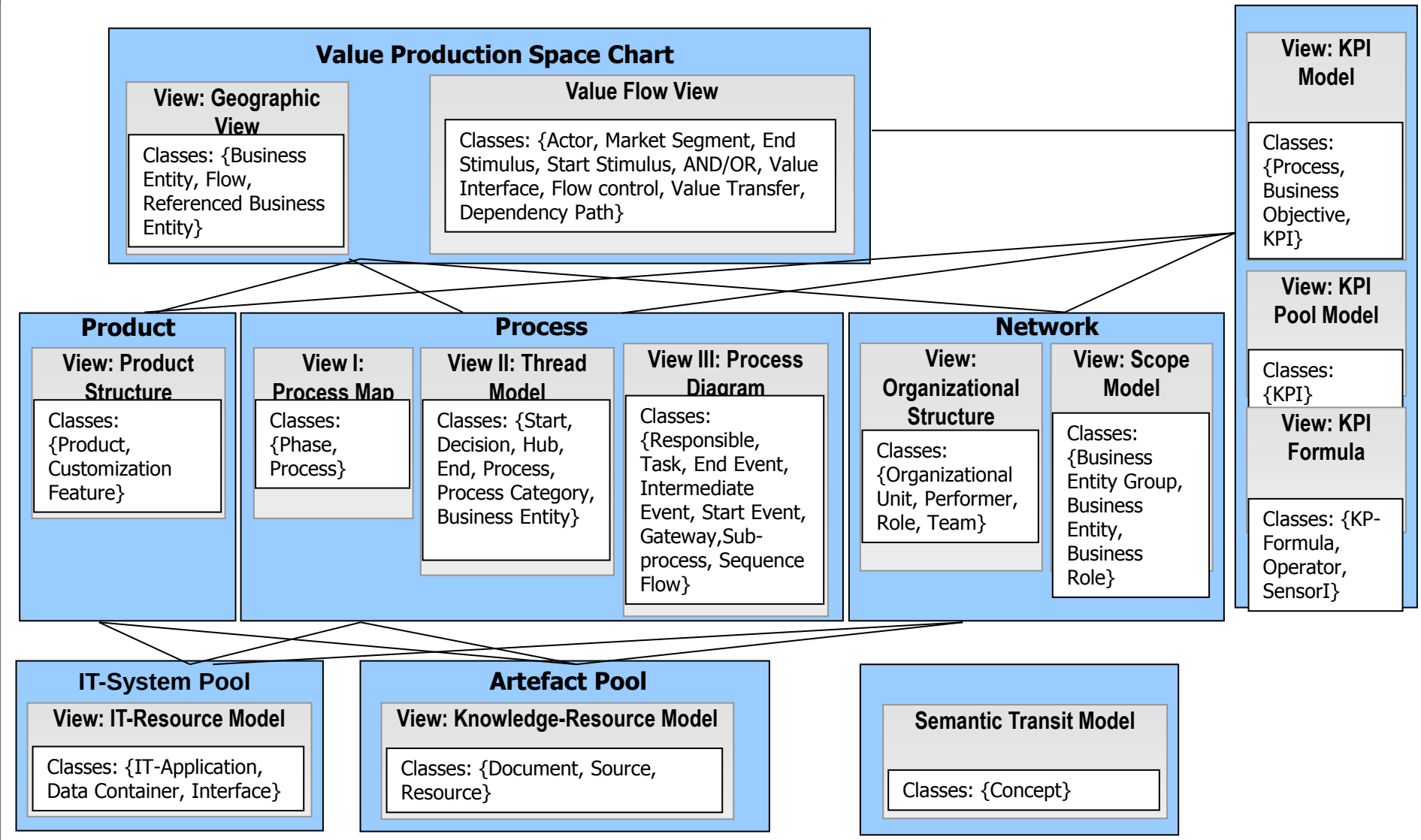
Case:

Realise a modelling tool for the Modelling Language “BIVEE Meta Model”.

ADOxx Meta2Model

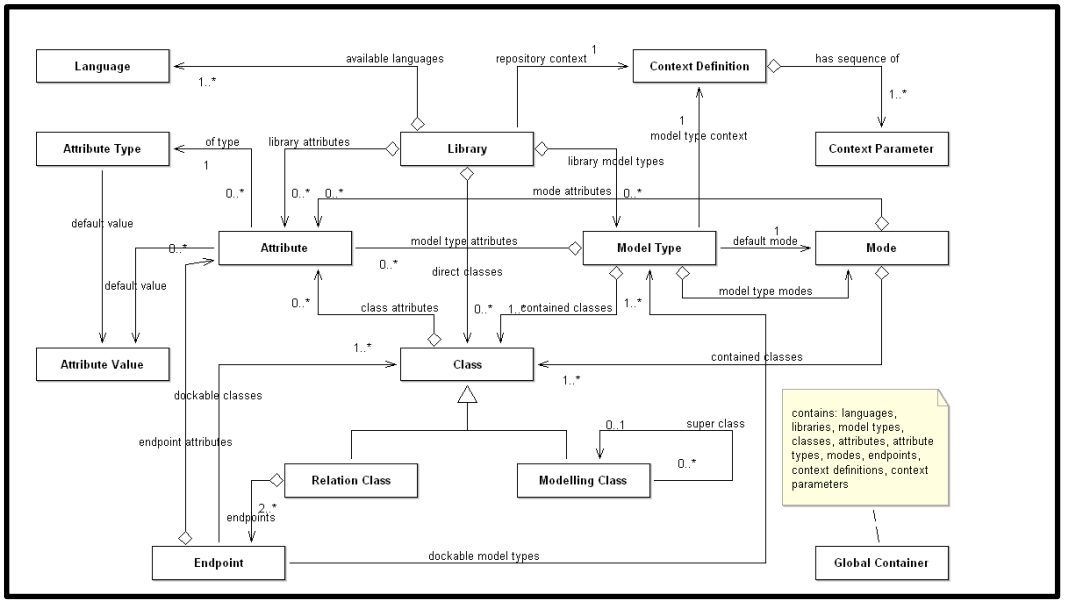


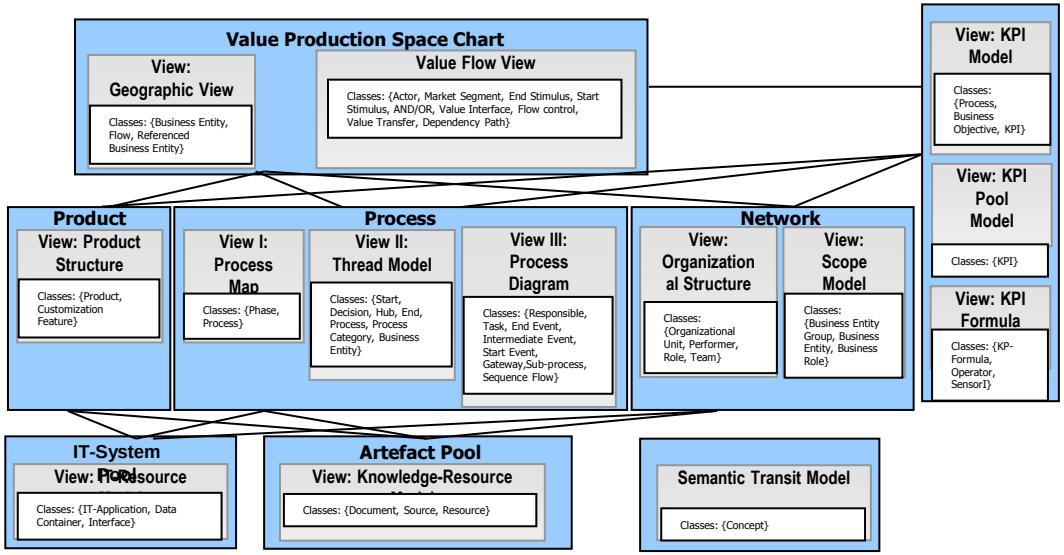
OVERVIEW OF BIVEE META MODEL



Modelling Method is to be find in ALL Repository in Adoxx.org

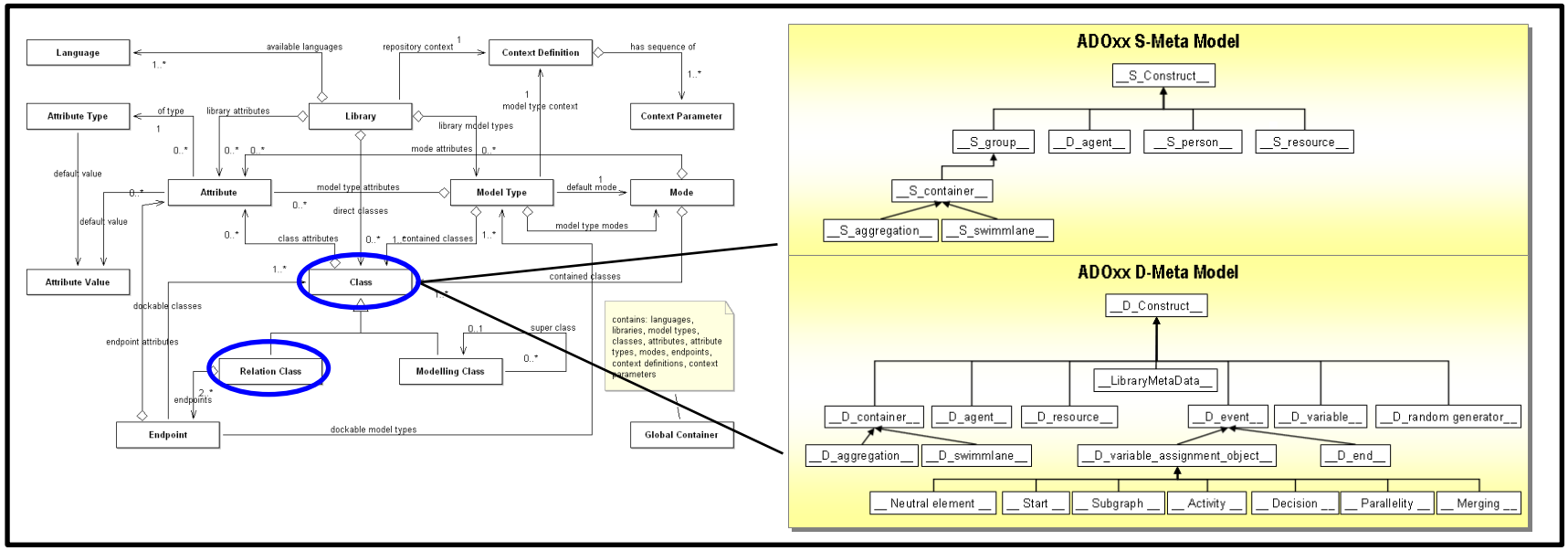
Mapping Meta2Model with BIVEE-Meta Model





How to map generic Meta²Model to a concrete Modelling Language ?

Operationalization of “CLASS” Concept



Operationalization : BIVEE Modelling Language

D-Meta Model

CLASS: __Cooperative_Class__, __Process_consultant_modelElement__, __GEO_Thing__, KPI, Preference Container, Business Objective, Note, Knowledge Resource

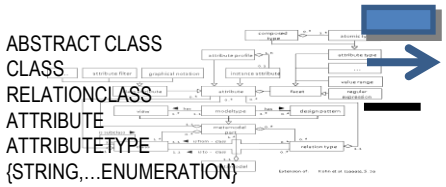
RELATIONCLASS: subsequent, connect, inside

S-Meta Model

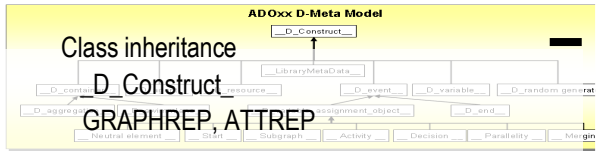
CLASS: Organizational Unit, Performer, Role, Team

Operationalizeable Meta Model

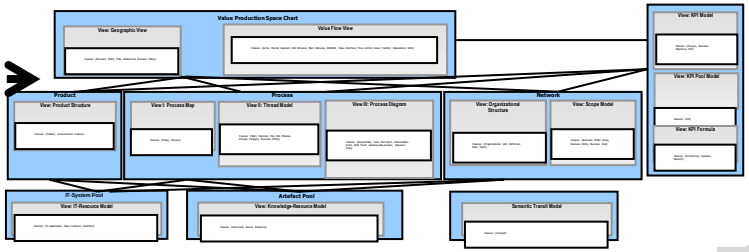
Meta²Model



Operationalisation MetaModel



BIVEE-MetaModel



Name, STRING

ADOxx MM CLASS: __D_Construct__

- ABSTRACT CLASS: __GEO_Thing__
  - CLASS: Business Entity
  - CLASS: Product
  - CLASS: Product
  - CLASS: Product
- ABSTRACT CLASS: __Cooperative_Class__
- ABSTRACT CLASS: __Process_consultant_modelElement__
  - CLASS: Actor
- CLASS: KPI
- CLASS: Preference Contatiner
- CLASS: Business Objective
- CLASS: Note
- CLASS: Knowledge Resource
- …

Meta Modelling Layer : Realising Modelling Language

Meta2Model:
GRAPHREP, MetaModel,
Modeltype, Class, Attribute

provides

ADOxx Meta2 Model

developed in C++, C#, Java

Instance of

GRAPHREP
GRAPHREP
AVAL atype:"Type"
PEN color:$000
IF (atype = "Bas
SET f:"green"
ELSIF (atype =
SET f:"orange
ELSIF (atype =
....

Specialisation of

MODI
MODELTYPE "Product Structure" from:none plural:"Product Structures" pos:0 not-simulateable bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
…

Meta Modell

Generic ADOxx Meta Model

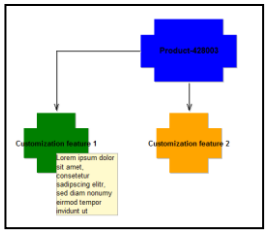
Inherited from

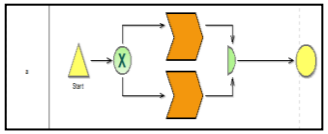
Method-specific Meta Model

developed in MM-DSL: ALL ADOxx Library Language

MM – Tool Implementaiton Part

Instance of





Model

described in ADL, XML

Provided Functionality of Metamodelling Platform

Used meta-modelling functionality :

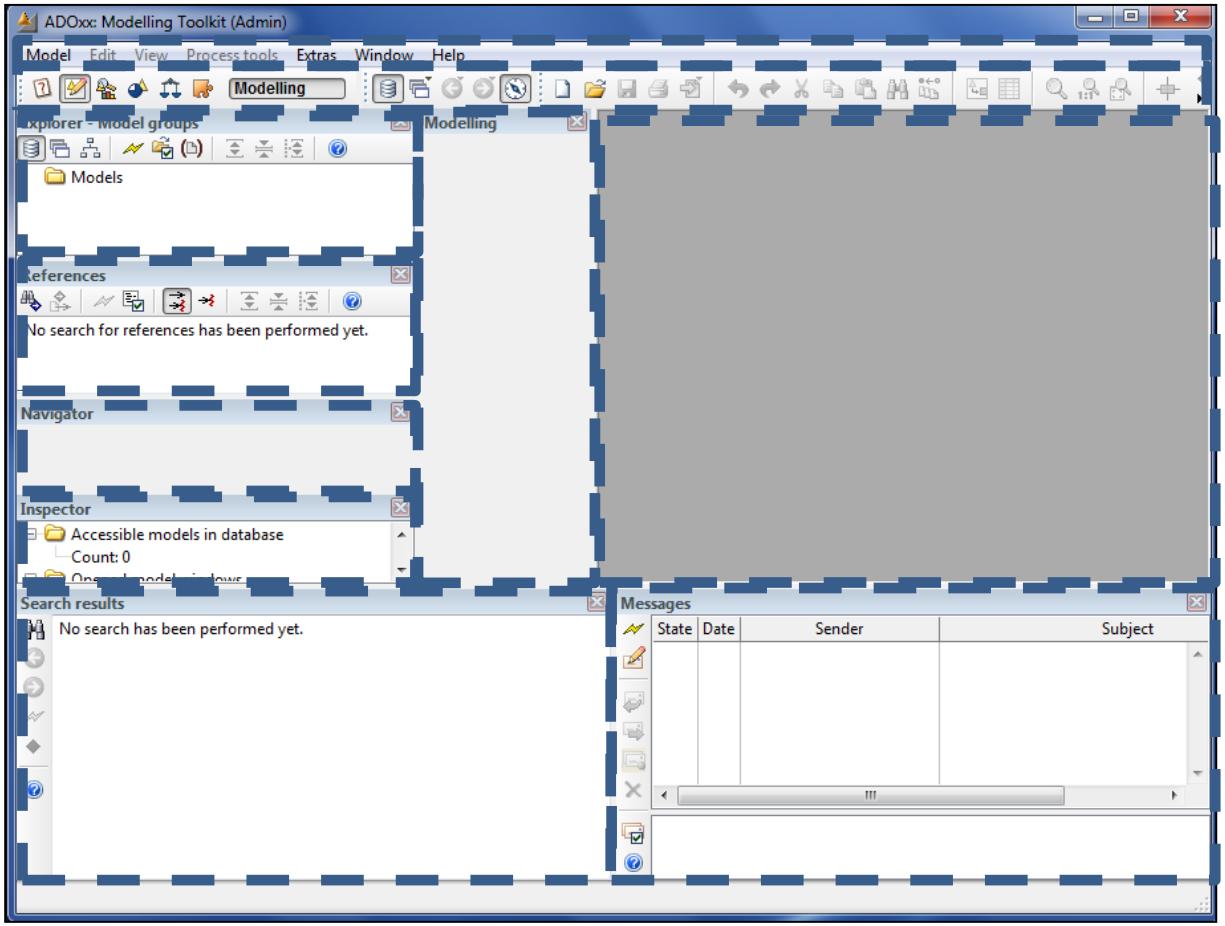
- **Meta²Model:** MODELTYPE, GRAPHREP, ATTREP, ATTRIBUTE TYPE, CLASS

- **ADOxx Meta2Model Component:**
  - Model Editor incl. Menubar
  - Query engine incl. AQL syntax
  - ADOscript interpreter and ADOscript syntax
  - Database

ADOxx Realisation HANDS-ON

1. Defining **MODELTYPES**
2. Inheriting **CLASSES** from ADOxx Meta Model
3. Implementing **GRAPHREP**
4. Inherit **RELATIONCLASSES** from ADOxx Meta Model
5. Defining **ATTRIBUTES** and **ATTREP**

GOAL: Development of Modelling Toolkit



Used ADOxx Functionality: Realising a Modelling Language

- 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 AdoScripts
    - Visualisation Expression
    - Query ADOscript
    - Transformation ADOscript
  - ADD-ON Implementation
    - ADOxx Web-Service
    - XML / ADL Import – Export
    - ADOscript Batch Mode

HANDS-ON

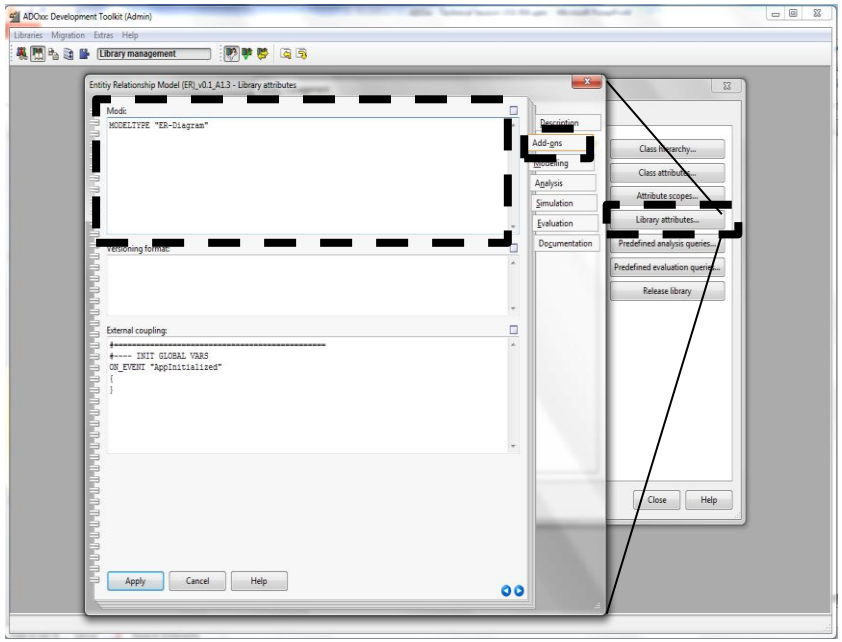
CASE: Entity Relationship Model

1. SCENARIO: REALISING A MODELLING LANGUAGE

ADOxx Realisation HANDS-ON

1. Defining **MODELTYPES**
2. Inheriting **CLASSES** from ADOxx Meta Model
3. Implementing **GRAPHREP**
4. Inherit **RELATIONCLASSES** from ADOxx Meta Model
5. Defining **ATTRIBUTES** and **ATTREP**

Define Modeltype



Define the Model-Types:

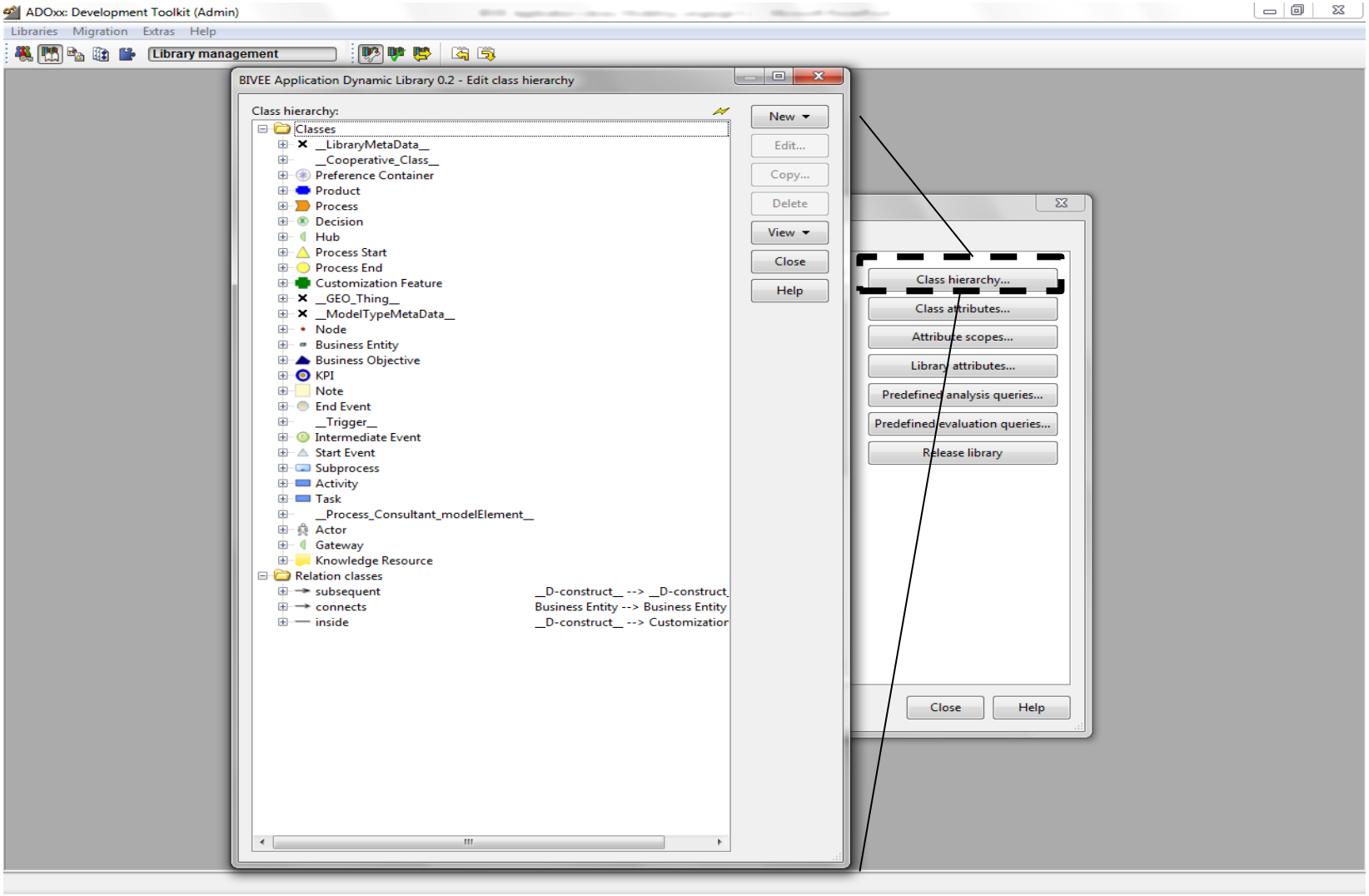
1. Click “Library attribute” of the BIVEE Application Dynamic Library
2. Go to “Add-on” chapter
3. Define the Modeltypes in the Modi textfield
   - "Thread Model“
   - "Product Structure“
   - "Business Process Diagram“
   - "Preferences Pool Model“
   - "KPI Model“

   (See Dynamic_MT.txt)

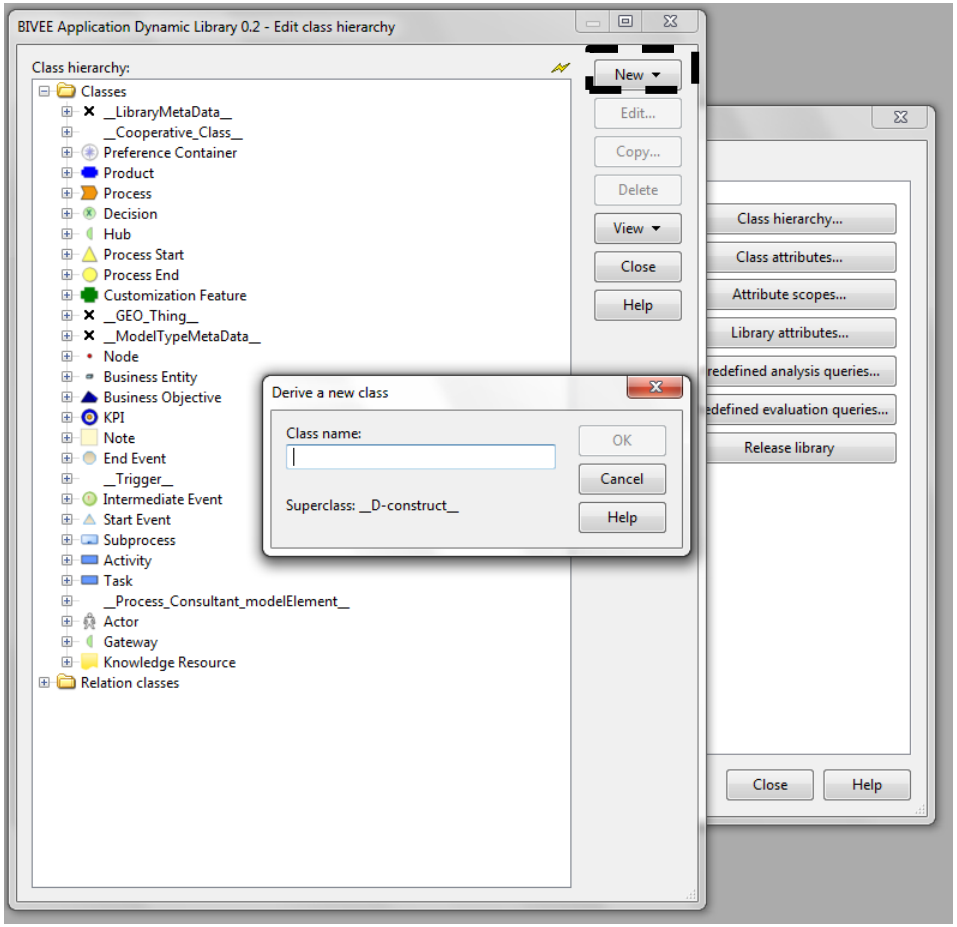
4. Click “Library attribute” of the BIVEE Application Static Library
5. Go to “Add-on” chapter
6. Define the Modeltypes in the Modi textfield
   - "Organizational Structure Model“

   (See Static_MT.txt)

Library Management



Define new class



Define Abstract Class:

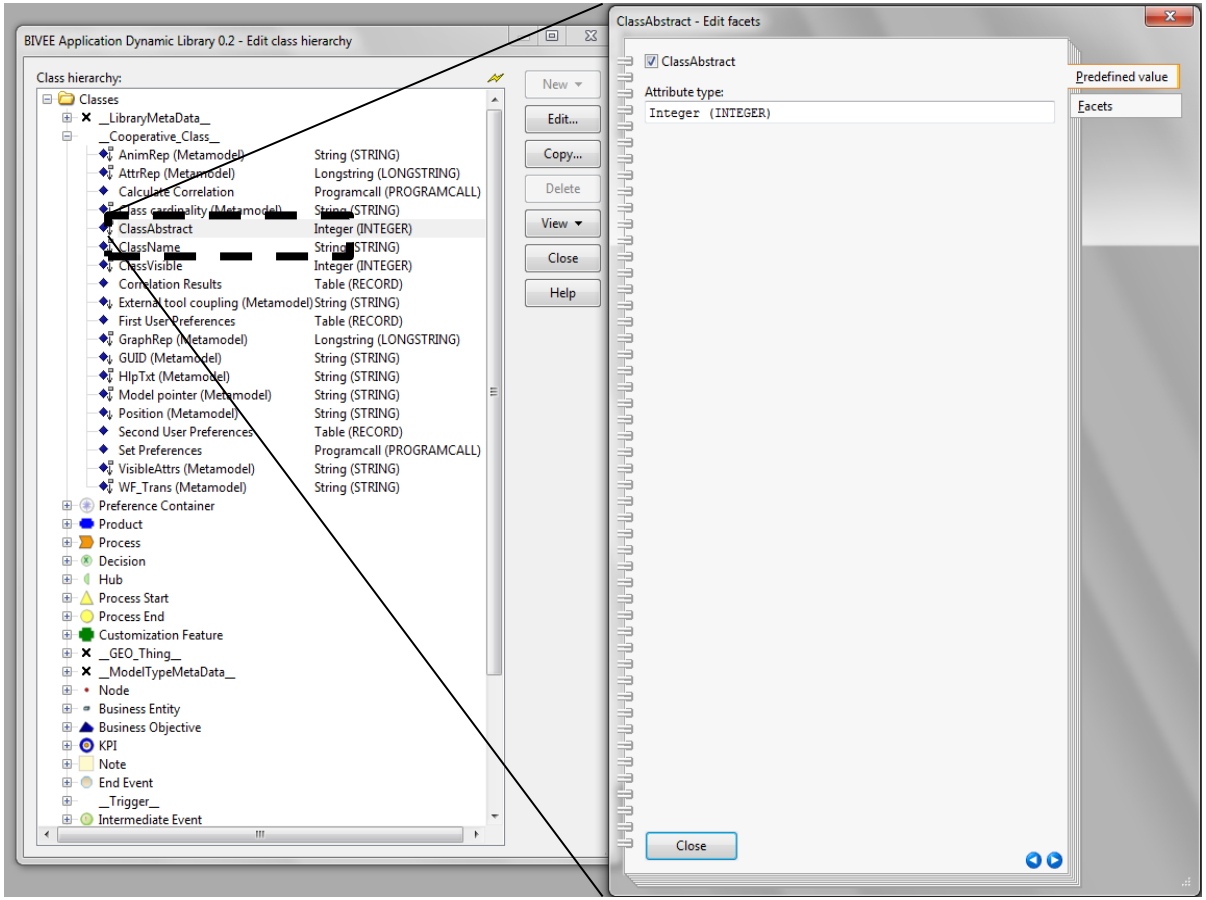
Add new abstract classes

1. Select root class, click “New” -> “New class”
   - __Cooperative_Class__,
   - __Process_consultant_modelElement__,
   - __GEO_Thing__,

Note:

Naming convention for abstract classes Pre- and Postfix is “_”

Make Class Abstract

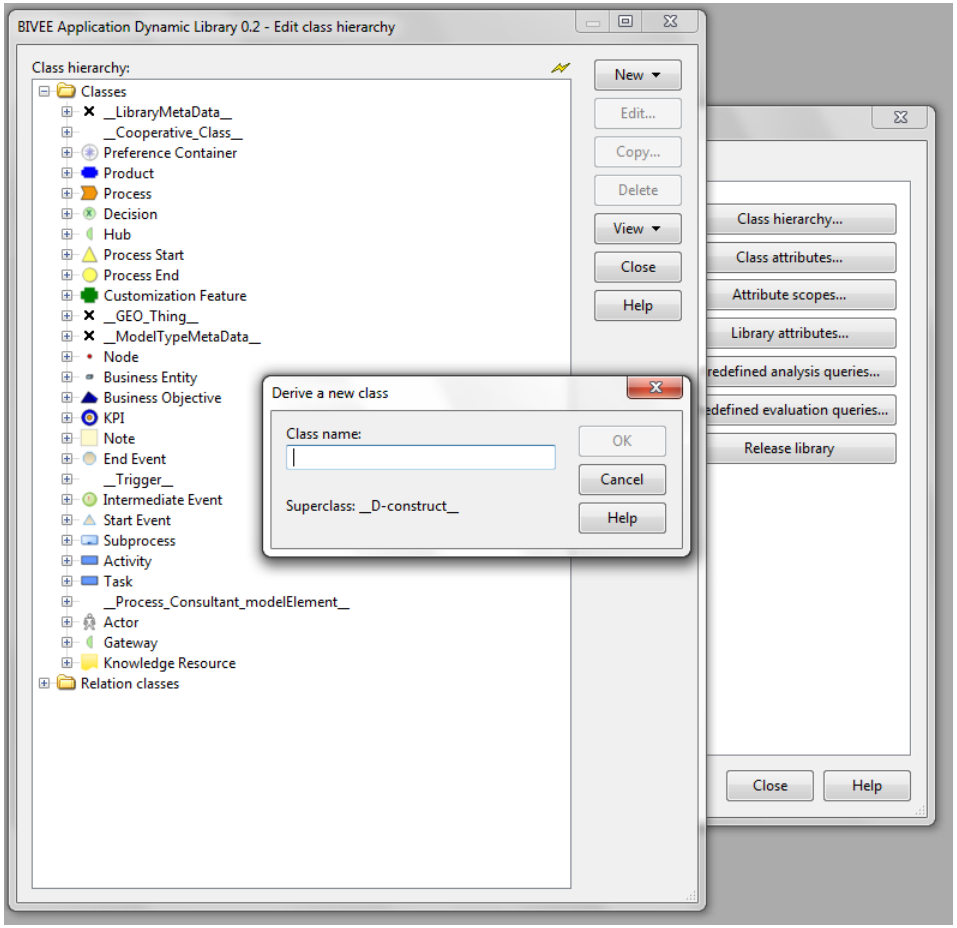


Make class abstract using “ClassAbstract“ attribute

--> Effect:

class can not be instantiated in the modelling tool.

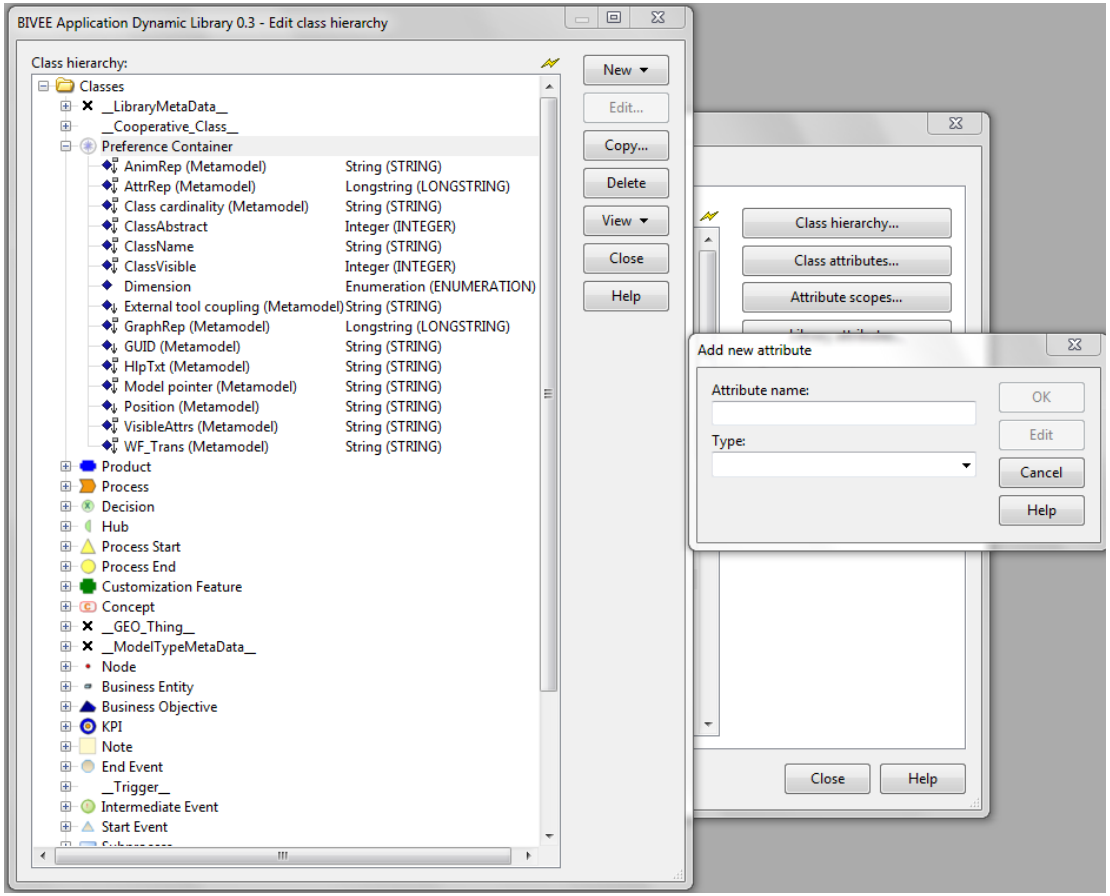
Define ER-Classes



Make new classes:

1. Preference Container
2. Product
3. Process
4. Decision
5. Hub
6. Process Start
7. Process End
8. Customization Feature
9. Business Entity
10. Business Objective
11. KPI
12. Note
13. End Event
14. Intermediate Event
15. Start Event
16. Subprocess
17. Activity
18. Task
19. Actor
20. Gateway
21. Knowledge Resource

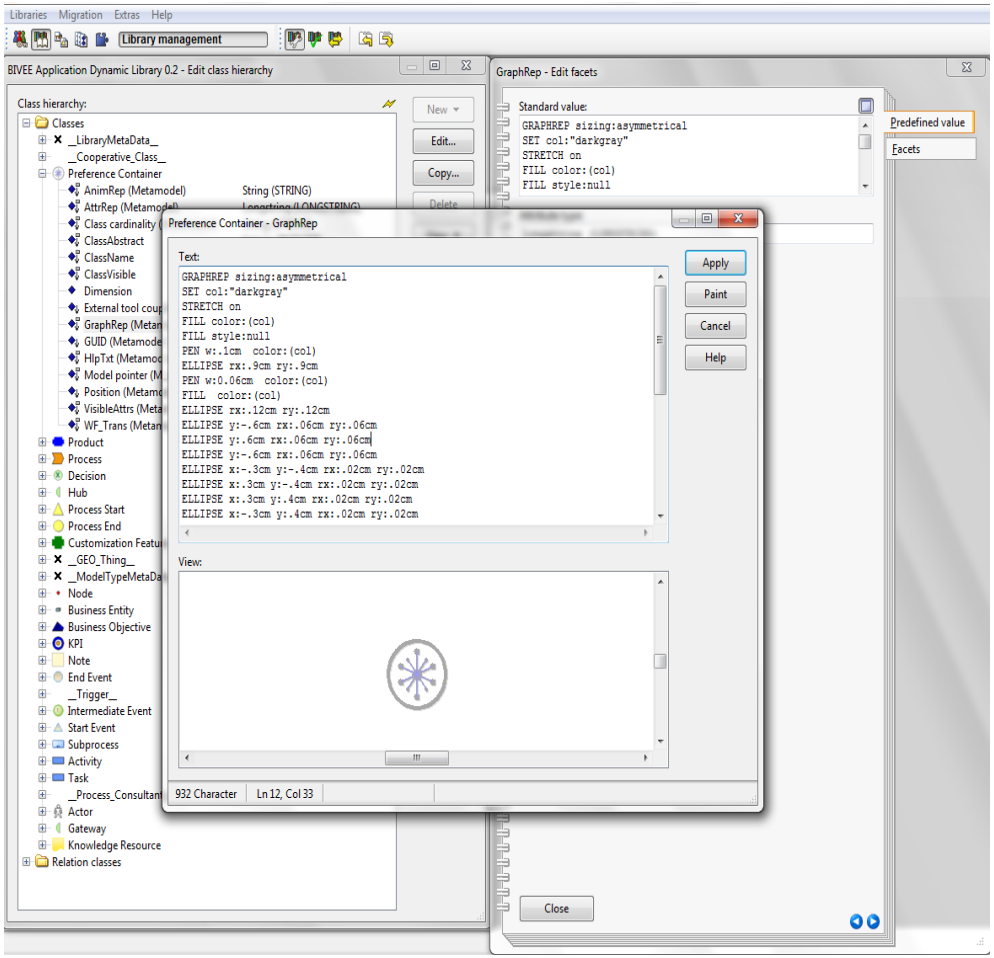
Attribute Definition



New Attribute

1. Select Class in which the new attribute has to be added
2. Add new attribute
3. Attribute name “<Attr_Name>”
4. Choose Type:
   - Attribute profile reference
   - Date
   - Datetime
   - Enumeration
   - Enumerationlist
   - Expression
   - Floating number
   - Integer
   - Intermodel reference
   - Longstring
   - Programcall
   - String
   - Table
   - Time

Implement GRAPHREP



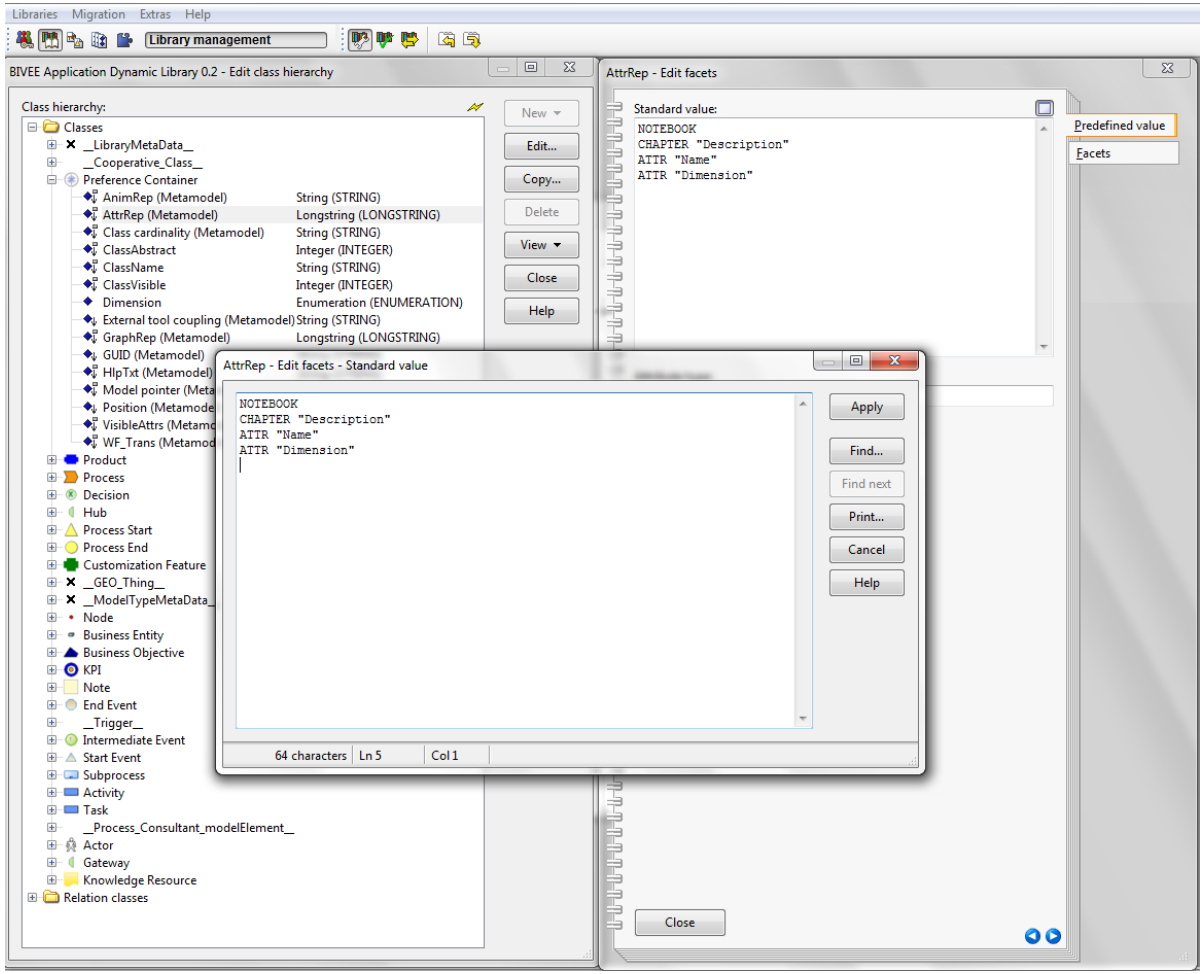
Implement GRAPHREP:

1. Edit Attribute GRAPHREP
2. Open GRAPHREP Editor
3. Write GRAPHREP Code in Text field
4. View current GRAPHREP with “Paint”
5. Store the GRAPHREP with “Apply”

All GraphReps of the classes can be found in the GraphRep Repository of ADOxx.org:

http://www.adoxx.org/live/adoxx-graphrep-repository

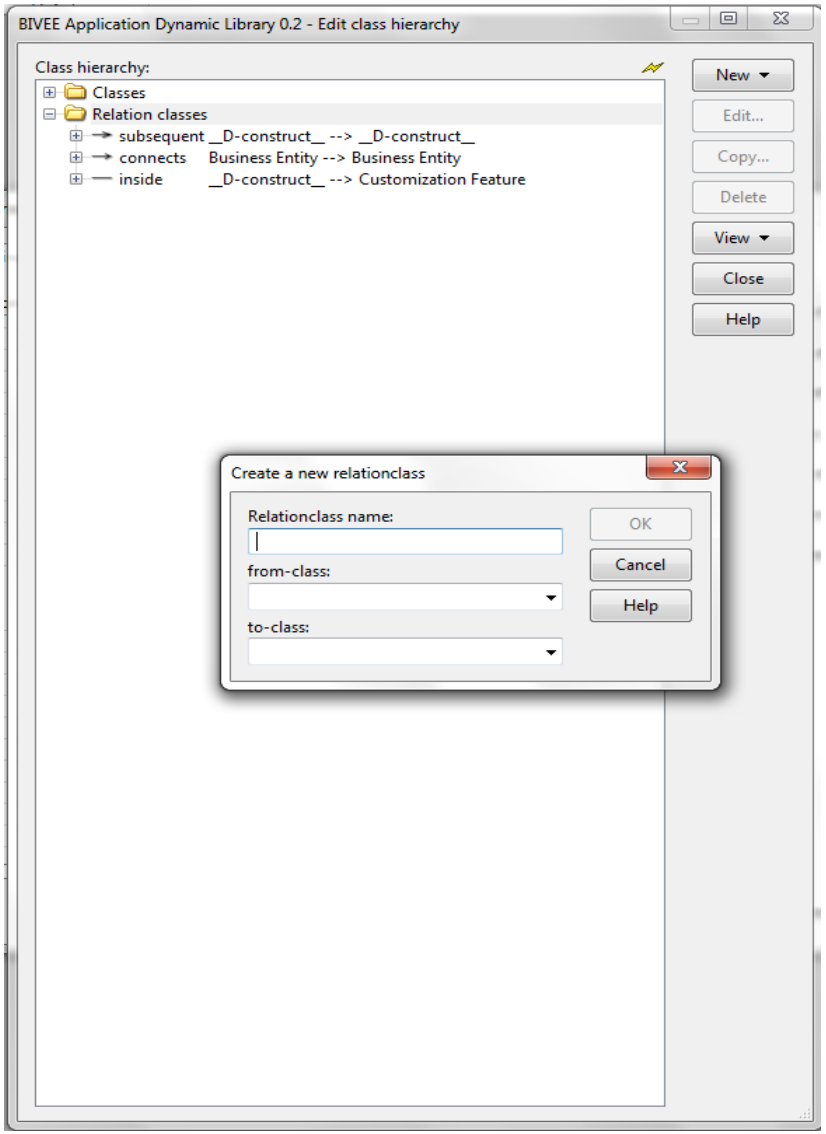
Implement ATTRREP



Implement ATTREP

1. Select Class
2. Edit ATTREP Attribute
3. Define NOTEBOOK
   - Chapter
   - ATTR
   - Group
   - ctrltype

Definition of RELATIONCLASS

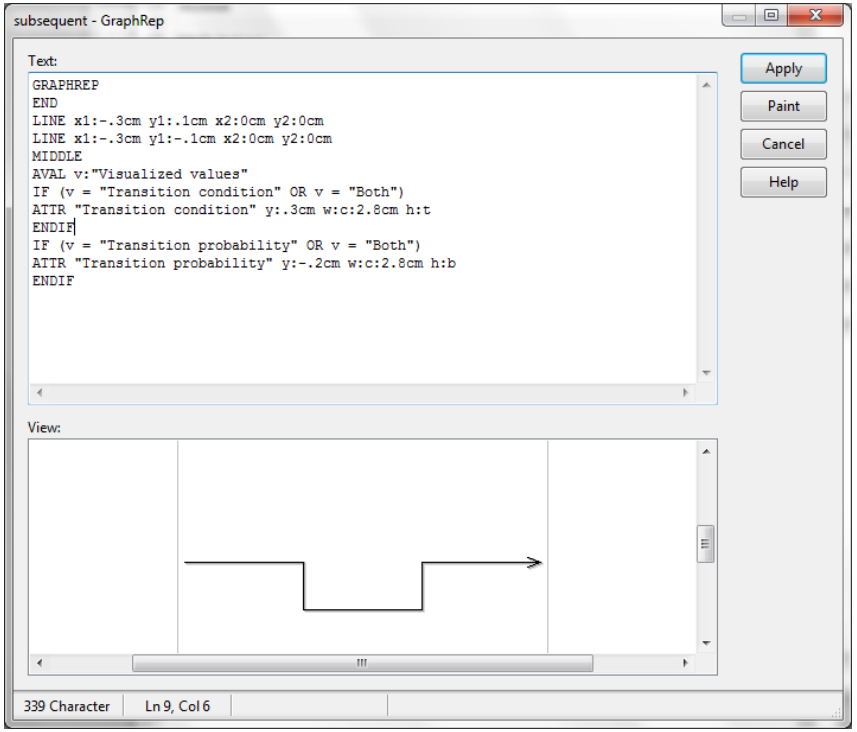


Define RELATIONCLASS:

- "subsequent"
- "connects"
- "inside"

1. Click new RelationClass
2. Define:
   - Name
   - From-class
   - To-class

Implement RELATION GRAPHREP



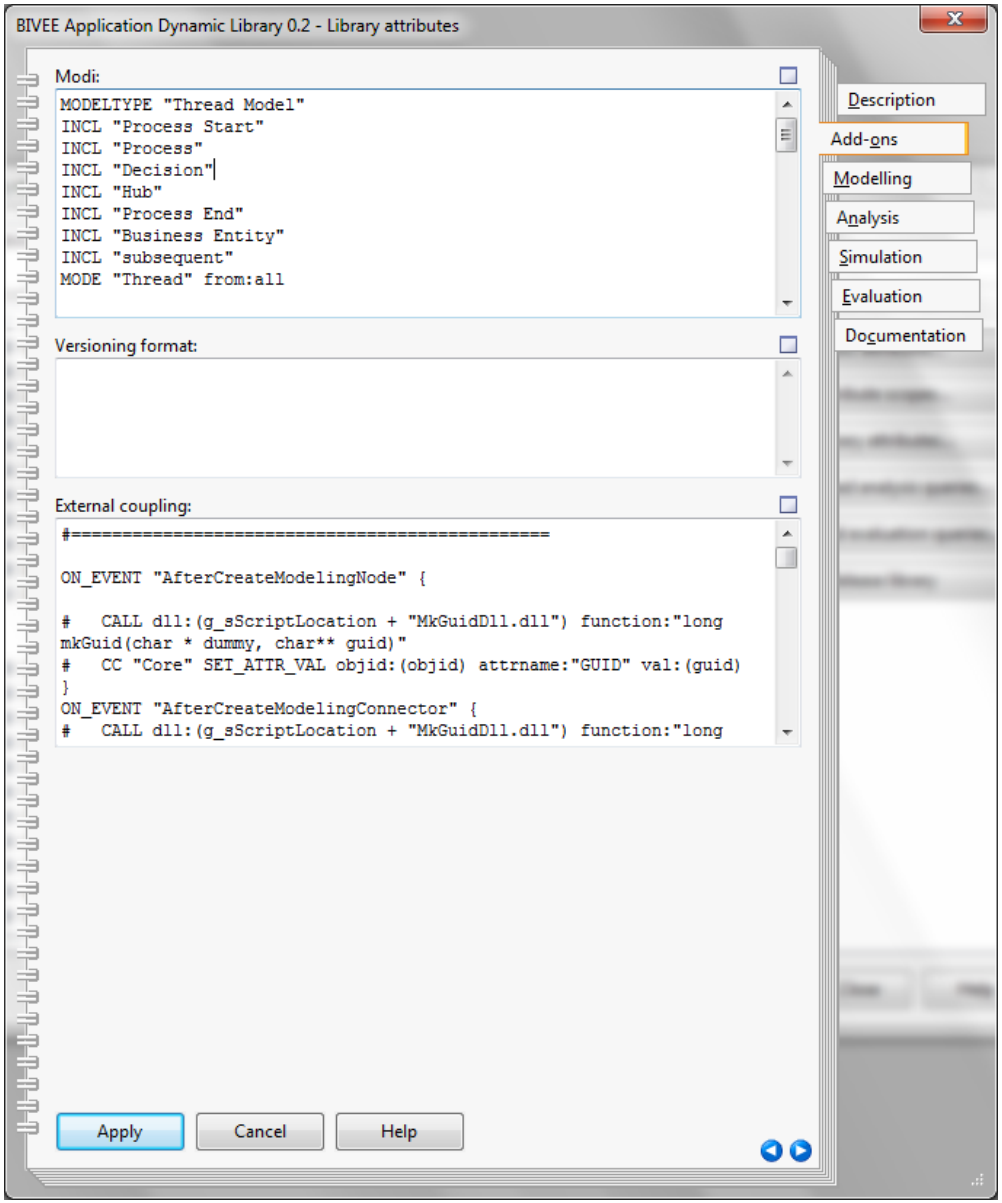

```
GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm
MIDDLE
AVAL v:"Visualized values"
IF (v = "Transition condition" OR v = "Both")
ATTR "Transition condition" y:.3cm w:c:2.8cm h:t
ENDIF
IF (v = "Transition probability" OR v = "Both")
ATTR "Transition probability" y:-.2cm w:c:2.8cm h:b
ENDIF
```

Define RELATION GRAPHREP:

1. Edit Attribute GRAPHREP
2. Open GRAPHREP Editor
3. Write GRAPHREP Code in Text field
4. View current GRAPHREP with “Paint”
5. Store the GRAPHREP with “Apply”

All GraphReps of the relation classes can be found in the GraphRep Repository of ADOxx.org:
http://www.adoxx.org/live/adoxx-graphrep-repository

Modeltype: Inclusion of Classes



Define the Model-Type:

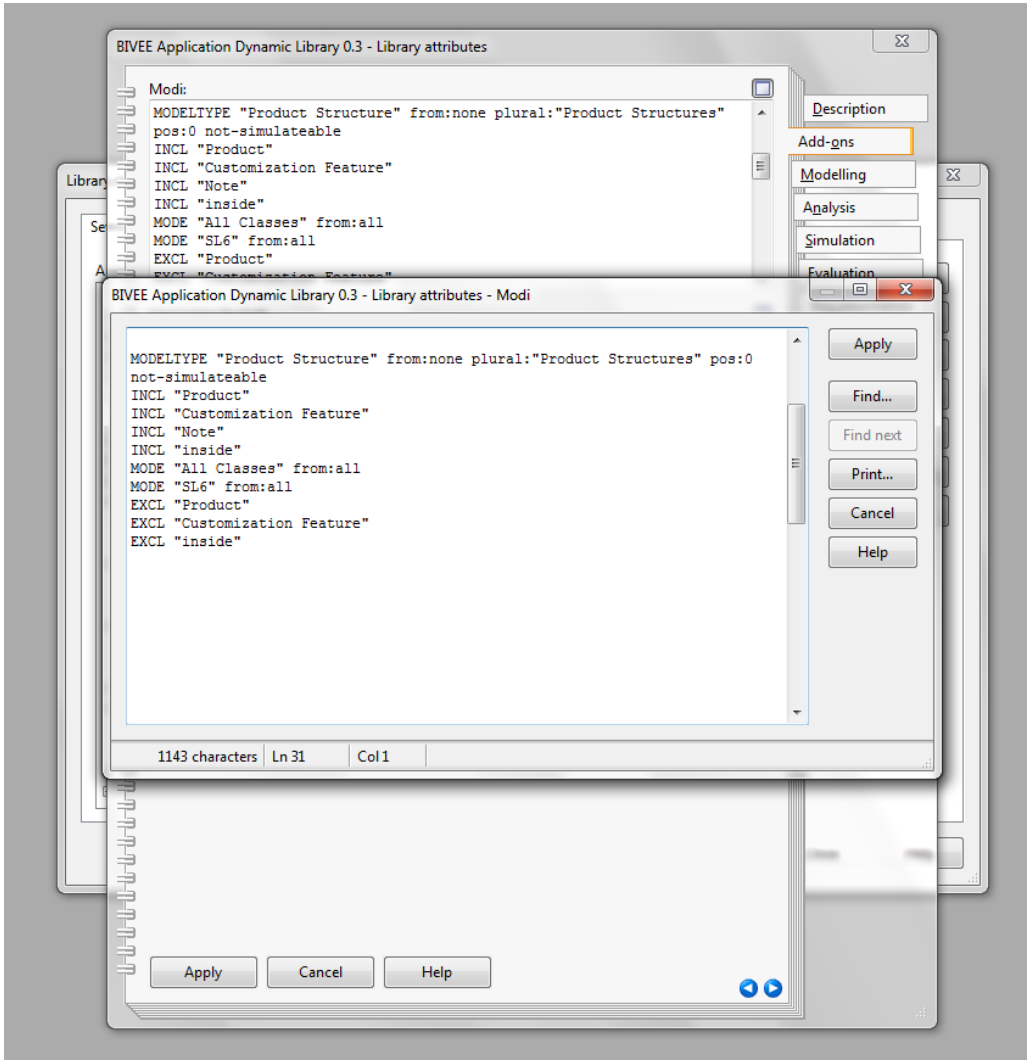
1. Click “Library attribute” of the ER-library
2. Go to “Add-on” chapter
3. Define the Modeltype in the Modi textfield.
4. MODELTYPE "Thread Model“

   Include classes:

   INCL "Process Start"
   INCL "Process"
   INCL "Decision"
   INCL "Hub"
   INCL "Process End"
   INCL "Business Entity"
   INCL "subsequent"
   MODE "Thread" from:all

   MODELTYPE "Preferences Pool Model"
   INCL "Preference Container"

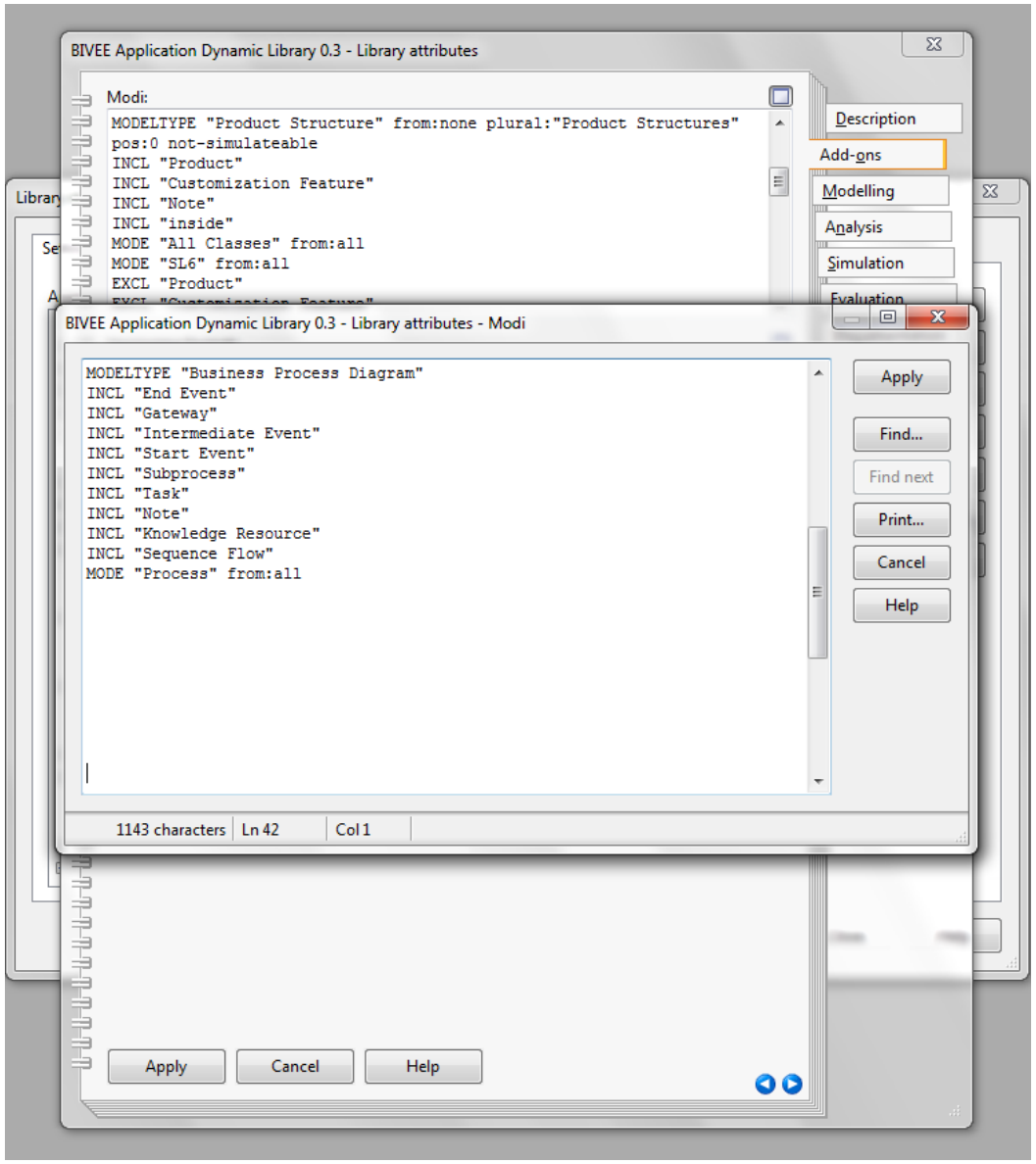
Modeltype: Inclusion of Classes



Define the Model-Type:

1. Click “Library attribute” of the ER-library
2. Go to “Add-on” chapter
3. Define the Modeltype in the Modi textfield.
4. MODELTYPE "Product Structure" from:none plural:"Product Structures" pos:0 not-simulateable
   INCL "Product"
   INCL "Customization Feature"
   INCL "Note"
   INCL "inside"
   MODE "All Classes" from:all
   MODE "SL6" from:all
   EXCL "Product"
   EXCL "Customization Feature"
   EXCL "inside"

Modeltype: Inclusion of Classes

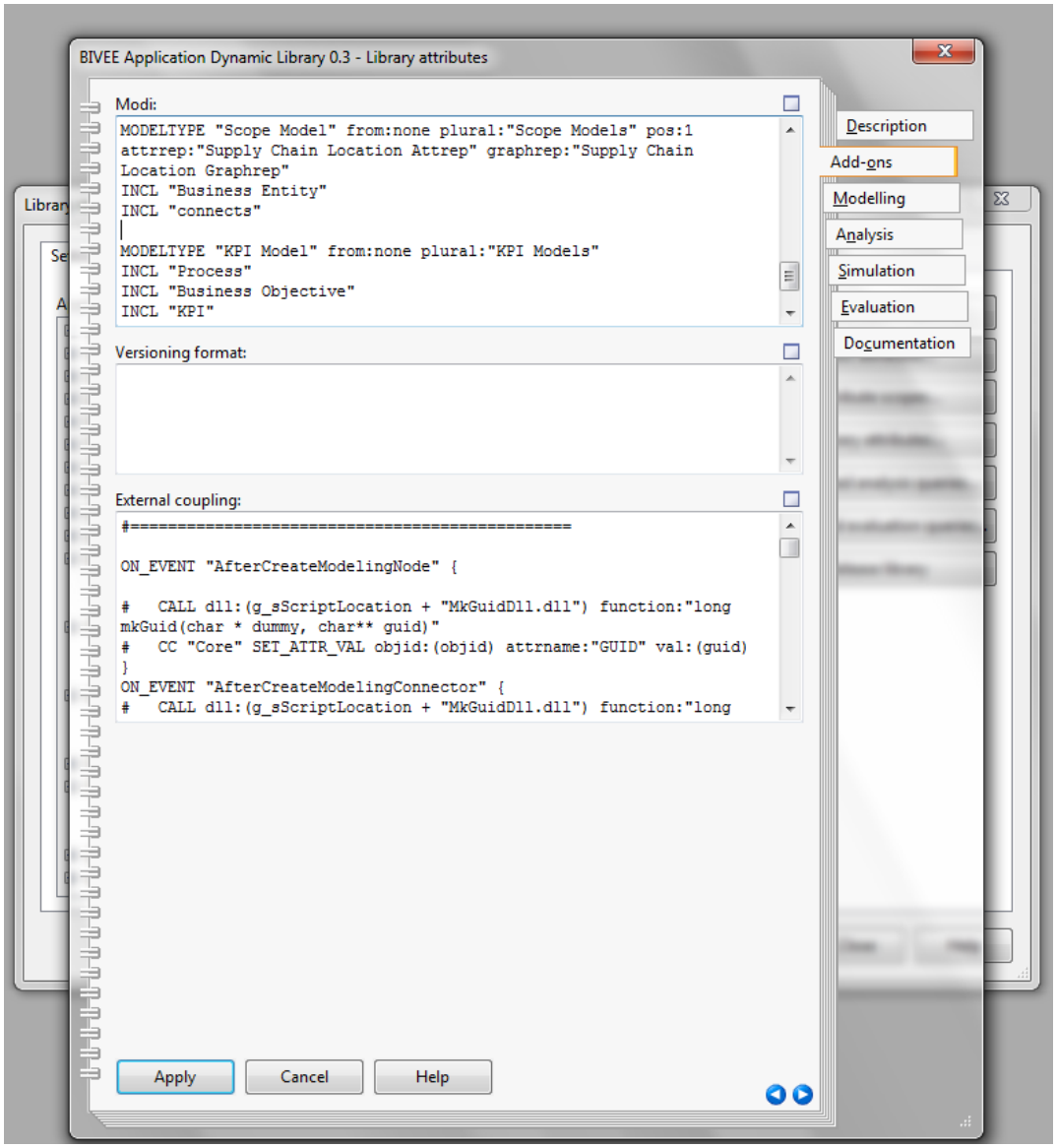


Define the Model-Type:

1. Click “Library attribute” of the ER-library
2. Go to “Add-on” chapter
3. Define the Modeltype in the Modi textfield.
4. MODELTYPE "Business Process Diagram"
   INCL "End Event"
   INCL "Gateway"
   INCL "Intermediate Event"
   INCL "Start Event"
   INCL "Subprocess"
   INCL "Task"
   INCL "Note"
   INCL "Knowledge Resource"
   INCL "Sequence Flow"
   MODE "Process" from:all

   MODELTYPE "Scope Model" from:none plural:"Scope Models" pos:1 attrrep:"Supply Chain Location Attrep" graphrep:"Supply Chain Location Graphrep"
   INCL "Business Entity"
   INCL "connects"

Modeltype: Inclusion of Classes



Define the Model-Type:

1. Click “Library attribute” of the ER-library
2. Go to “Add-on” chapter
3. Define the Modeltype in the Modi textfield.
4.

MODELTYPE "KPI Model" from:none plural:"KPI Models"
INCL "Process"
INCL "Business Objective"
INCL "KPI"

Further Questions?

www.adoxx.org

tutorial@adoxx.org