

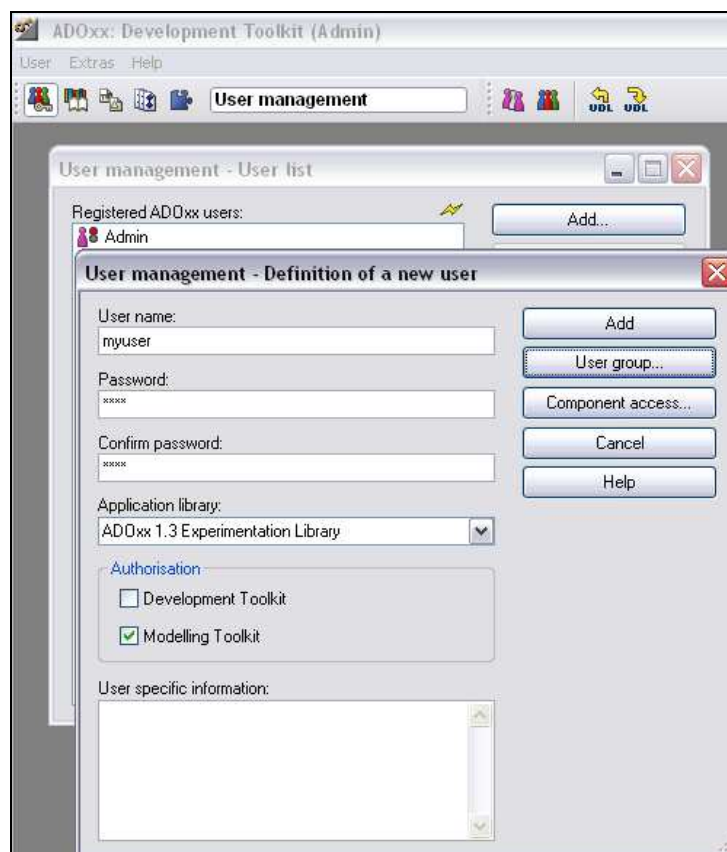
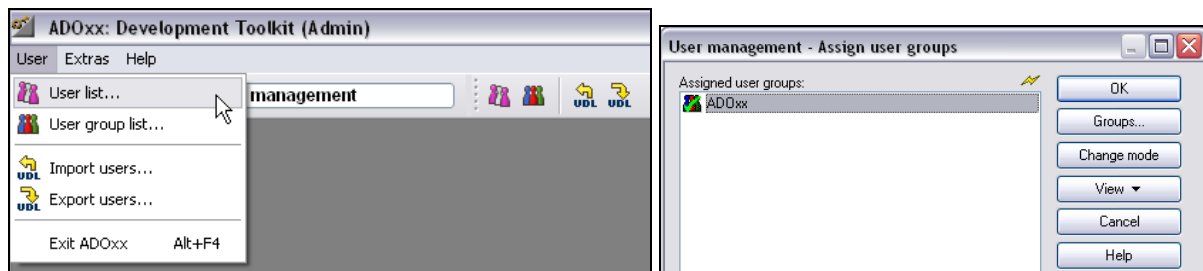


ADOxx Tutorial – Cookbook: MODELLING LANGUAGE IMPLEMENTATION ON ADOxx

Phase 1. PREPARATION

In this phase you will install and set up ADOxx 1.3 UL1. If you have already installed ADOxx 1.3 UL1, you can skip this and continue with Phase 2.

1. Download the latest version of ADOxx from the **adoxx.org** portal
Link: <http://www.adoxx.org/live/download>
2. Install ADOxx to your computer at a location of your choice. If needed, consult the [Installation Guide](#).
3. Download and unpack the tutorial package from the **adoxx.org** portal.
Link: <http://www.adoxx.org/live/tutorial> [tutorial package](#)
4. Unpack the tutorial package to allocation of your choice. This location will be referred to as "**<mypath>\<tutorial_package>**".
5. Log into the ADOxx Development Toolkit with the username "Admin" and the password "password".
6. Create an user for modeling and testing purposes in the "User Administration" area



Phase 2. IMPLEMENT MODELLING LANGUAGE

2.1. Introduction to the modeling language

After having installed and set up ADOxx 1.3 UL1, you can start developing a basic modeling method. The classes and relation classes for the modeling language implemented in this exercise are described below.

(Relation-) Class	Shape	Colour	Size
A	Circular	Color: blue	Radius: 1.0 cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	a1	INTEGER	Description
	a2	RECORD	Description
	a3	STRING	Description
	a4	INTEGER	Description
	a5	INTERREF	References
B	Triangular	Color: blue	Height: 1.6cm, Width: 1.6cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	b1	INTEGER	Description
	b2	RECORD	Description
	b3	STRING	Description
	b4	STRING	Description
C	Circular Circular	Colour1: blue Colour2: yellow	Radius: 1.0 cm Radius: 0.5 cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	a1	INTEGER	Description
	a2	RECORD	Description
	a3	STRING	Description
	a4	INTEGER	Description
	a5	INTERREF	References
	__Conversion__	LONGSTRING	
D	Triangular Triangular	Color: blue Color: yellow	Height: 1.6cm, Width: 1.6cm Height: 0.7cm, Width: 0.8cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	b1	INTEGER	Description
	b2	RECORD	Description
	b3	STRING	Description
	b4	STRING	Description
	__Conversion__	LONGSTRING	
E	abstract class		

(Relation-) Class	Shape	Colour	Size
	Attribute Name	Type	Chapter
	Name	STRING	Description
	a1	INTEGER	Description
	a2	RECORD	Description
	a3	STRING	Description
	a4	INTEGER	Description
	a5	INTERREF	References
	__Conversion__	LONGSTRING	
	b1	INTEGER	Description
	b2	RECORD	Description
	b3	STRING	Description
E	Circular	Color: blue	Radius: 1.0cm
	Triangular	Color: blue	Height: 1.6cm, Width: 1.6cm
	Circular	Color: yellow	Radius: 0.5cm
	Triangular	Color: yellow	Height: 0.7cm, Width: 0.8cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	a1	INTEGER	A
	a2	RECORD	A
	a3	STRING	A
	a4	INTEGER	A
	a5	INTERREF	A, References
	__Conversion__	LONGSTRING	
	b1	INTEGER	B
	b2	RECORD	B
	b3	STRING	B
	b4	STRING	B
	e1	INTEGER	Description
	e2	RECORD	Description
	e3	STRING	Description
	e4	ATTRIBUTEPROFILEREFERENCE	References
V derived from class __Aggregation__	Rectangular	Color: blue, red or yellow, depending on the value of the attribute "Type-Selection"	Height: 6.0cm, Width: 6.0cm
	Attribute Name	Type	Chapter
	Name	STRING	Description
	Type-Selection	ENUMERATION	Description
	Values: type-1 , type-2, type-3		
	V-Text	LONSTRING	Description

W derived from class __D_construct__	Rectangular	Color: blue, red or yellow, depending on the value of the attribute "Type-Selection"	Height: 6.0cm, Width: 6.0cm
	Attribute Name	Type	Chapter

(Relation-) Class	Shape	Colour	Size
	Name	STRING	Description
	Type-Selection	ENUMERATION	Description
	Values: type-1 , type-2, type-3		
	V-Text	LONSTRING	Description
G	abstract class		
	Attribute Name	Type	Chapter
	Name	STRING	Description
	Description	LONGSTRING	Description
H	Rectangle, with two "actors" inside	Color: white Lines: black	Height: 6.0cm, Width: 6.0cm
	Attribute Name	Type	Chapter
	Name	STRING	Definition
	Description	LONGSTRING	Definition, Group "Definition"
	External Content	LONSTRING	Definition, Group "Definition"
I	Arrow, as a polygon	Color: blue	Height: 2.0cm, Width: 3.0cm
	Attribute Name	Type	Chapter
	Name	STRING	Definition
	Description	LONGSTRING	Definition
	Influencing Dialectics	LONSTRING, lines:10	Dialectic Influence
X	External graphic	Color: -	Height: 5.0cm, Width: 5.0cm
	Attribute Name	Type	Chapter
	Name	STRING	Definition
	External graphic	STRING	Definition
aRb	arrow	black	1 pt
	connects:	A -> B	
	Attribute Name	Type	Chapter
	Name	STRING	Description
	Comments	LONGSTRING	Description
anyRany	line	darkblue	1 pt
	connects:	any -> any	
	Attribute Name	Type	Chapter
	Name	STRING	Description
	Comments	LONGSTRING	Description
RC1	record class	no representation	
	Attribute Name	Type	Chapter
	RC1_a1	STRING	
	RC1_a2	LONGSTRING	
	RC1_a3	INTEGER	
	RC1_a4	TIME	
AP1	attribute profile class	no representation	
	Attribute Name	Type	Chapter
	AP1_a1	STRING	
	AP1_a2	LONGSTRING	

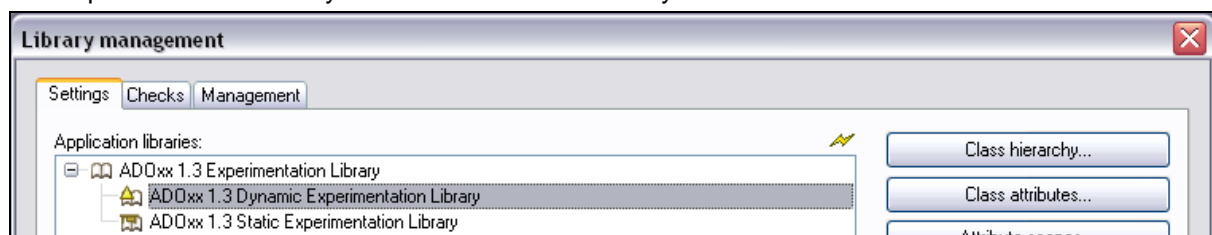
(Relation-) Class	Shape	Colour	Size
	AP1_a3	INTEGER	
	AP1_a4	TIME	

Slides for this phase are located in the file “**2 Modelling Language Implementation on ADOxx-PUBLIC.pdf**”, located in the folder “<mypath>\tutorial_package\Tutorial_Slides”

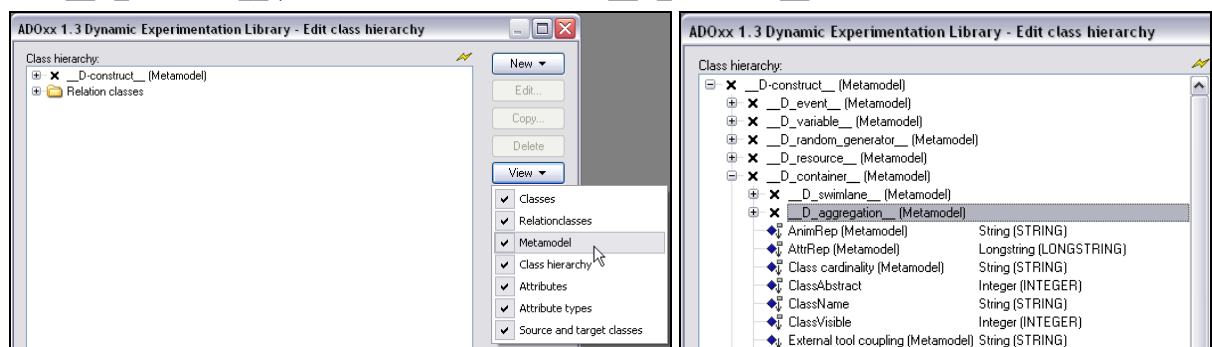
2.2. Create the classes

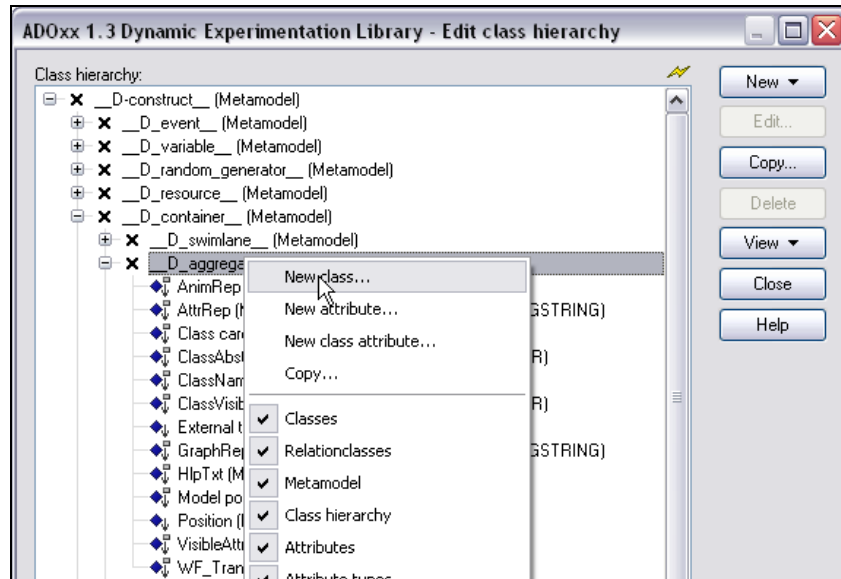
For help with steps 1 to 12, use the slides 15 – 34.

1. Open the Class hierarchy window: click the icon “Library Management” -> “Libraries” -> “Settings...” -> expand “ADOxx 1.3 Experimentation Library” -> select ADOxx 1.3 Dynamic Experimentation Library -> click the “Class hierarchy” button



2. Create a concrete class “A”, derived from class “__D_construct__”.
3. Create a concrete class “C”, derived from class “A”.
4. Create an abstract class “_E_”, derived from class “C”.
5. Create a concrete class “E”, derived from class “_E_”.
6. Create a concrete class “B”, derived from class “__D_construct__”.
7. Create a concrete class “D”, derived from class “B”.
8. Create an abstract class “_G_”, derived from class “__D_construct__”.
9. Create a concrete class “H”, derived from class “_G_”.
10. Create a concrete class “I”, derived from class “__D_event__”. For completing this step, you will need to activate the “Metamodel” and “Class hierarchy” views in the “Class hierarchy” display: “Class hierarchy” -> “View”. The class “__D_aggregation__” is derived from class “__D_container__”, who is derived from class “__D_construct__”.





11. Create a concrete class "V", derived from class "__D_aggregation__" (ref. slides 15 - 34). The class "__D_aggregation__" is derived from class "__D_container__", who is derived from class "__D_construct__".
12. Create a concrete class "X", derived from class "__D_aggregation__" (ref. slides 15 - 34).
13. Create a concrete class "W", derived from class "__D_construct__" (ref. slides 15 - 34)

2.3. Create the relation classes

For help with steps 13 and 14, use the slides 35 - 42.

14. Create a relation class "anyRany" from class "__D_construct__" to class "__D_construct__"
15. Create a relation class "aRb" from class "A" to class "B"

2.4. Define the model types

16. Define a model type "Sample" containing the classes "A", "B", "C", "D" and "E" and the relation classes "anyRany" and "aRb" (see slides 142 - 151)
17. Define a model type "Practice" containing the classes "A", "B", "V" and "W" and the relation classes "anyRany" (see slides 142 - 151)

2.5. Define the record class and attribute profile class

18. Define a record class "RC1" (see slides ...)
19. Define an attribute profile class "AP1" (see slides ...)

2.6. Create the attributes for the classes

For help with steps 15 to 23, use the slides 43 - 124.

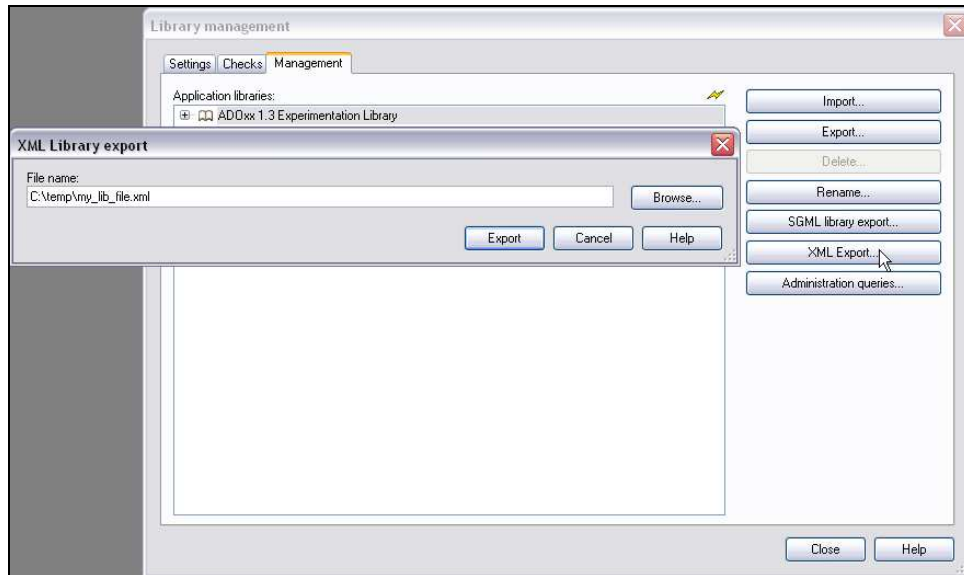
20. For class "A", create the following attributes:
 - a. attribute "a1" of type INTEGER,
 - b. attribute "a2" of type RECORD, referring to record class "RC1"
 - c. attribute "a3" of type STRING,
 - d. attribute "a4" of type INTEGER,
 - e. attribute "a5" of type INTERREF, referring to a class "V" in a model type "Practice"
21. For class "B", create the following attributes:

- a. attribute "b1" of type INTEGER,
 - b. attribute "b2" of type RECORD, referring to record class "RC1"
 - c. attribute "b3" of type STRING,
 - d. attribute "b4" of type STRING
- 22. For class "C", create attribute "__Conversion__" of type LONGSTRING (see slides 48, 101-106)
- 23. For class "D", create attribute "__Conversion__" of type LONGSTRING (see slides 48, 101-106)
- 24. For class "_E_", create the following attributes:
 - a. attribute "b1" of type INTEGER,
 - b. attribute "b2" of type RECORD, referring to record class "RC1"
 - c. attribute "b3" of type STRING
- 25. For class "E", create the following attributes:
 - a. attribute "e1" of type INTEGER,
 - b. attribute "e2" of type RECORD, referring to record class "RC1"
 - c. attribute "e3" of type STRING,
 - d. attribute "b4" of type STRING
 - e. attribute "e2" of type ATTRIBUTEPROFILEREFERENCE, referring to attribute profile class "AP1"
- 26. For class "V" create the following attributes:
 - a. attribute "Type-Selection" of type ENUMERATION, with the values "type-1", "type-2", "type-3"
 - b. attribute "V-Text" of type STRING
- 27. For class "W" create the following attributes:
 - a. attribute "Type-Selection" of type ENUMERATION, with the values "type-1", "type-2", "type-3"
 - b. attribute "V-Text" of type STRING
- 28. For record class RC1 create the following attributes:
 - a. attribute "RC1_a1" of type STRING
 - b. attribute "RC1_a2" of type LONGSTRING
 - c. attribute "RC1_a3" of type INTEGER
 - d. attribute "RC1_a4" of type TIME
- 29. For record class RC1 create the following attributes:
 - a. attribute "AP1_a1" of type STRING
 - b. attribute "AP1_a2" of type LONGSTRING
 - c. attribute "AP1_a3" of type INTEGER
 - d. attribute "AP1_a4" of type TIME

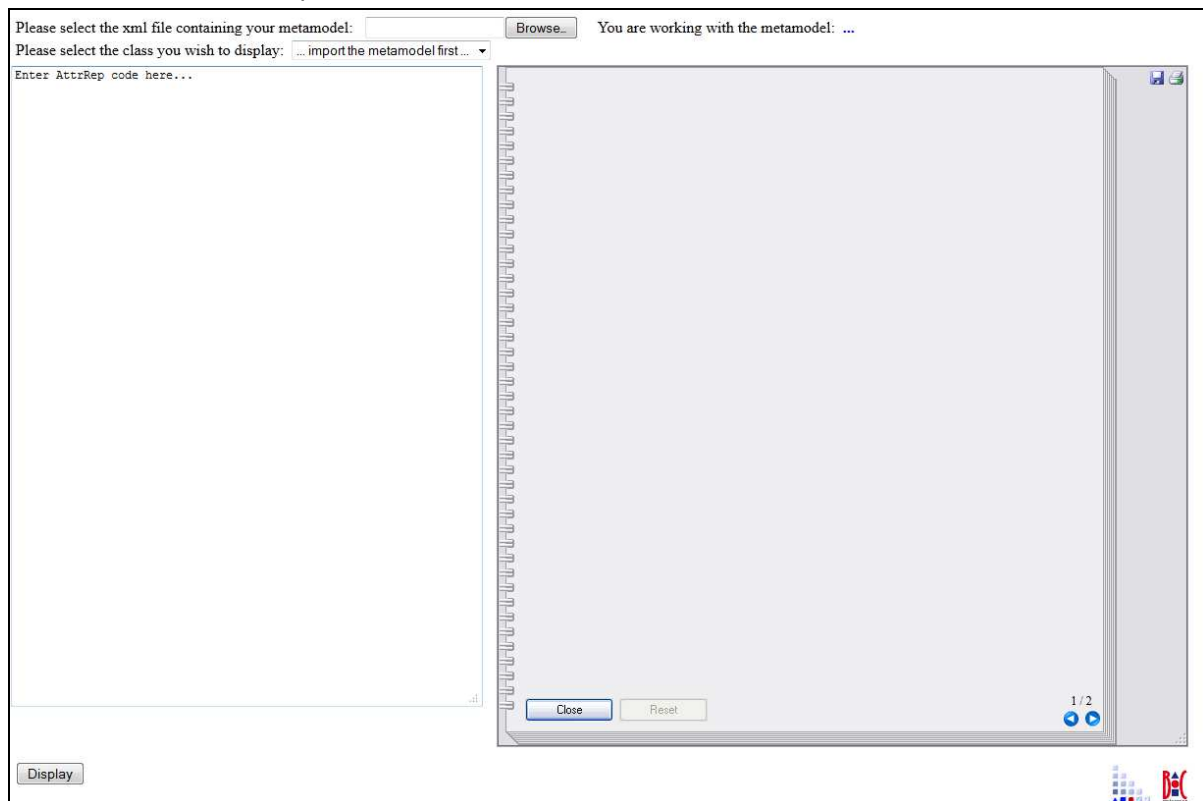
2.7. Add graphical representation and attribute representation for the classes and relation classes

- 30. Write the code for the GraphRep of each concrete class and relation class you have defined (see slides 60 - 85); after changing the contents of the GraphRep class attribute, you can have a preview of the graphical representation by clicking the "Paint" button. Various samples of GraphRep code can be found in the folder "`<mypath>\<tutorial_package>\Tutorial_Material\1_Tutorial1_GraphRep`".
- 31. For each concrete class and relation class in your modeling language edit the AttrRep class attribute to define how the information about each object will be displayed (see slides 92 – 94).
- 32. In order to easily test the appearance of the notebook for each class, use the "AttrRep_Viewer" tool from www.adoxx.org by following these steps:
 - a. Download the "AttrRep_Viewer" package from the download area of the adoxx.org portal; [\(internal\) Link](#)

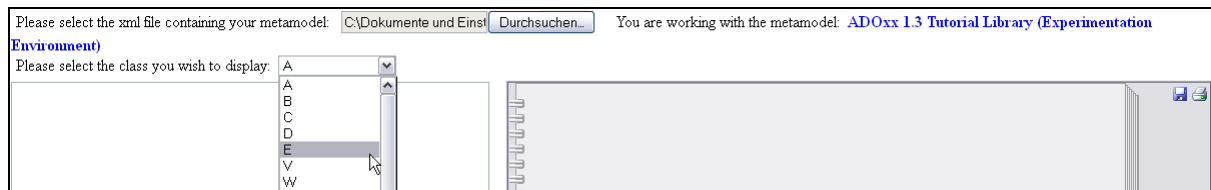
- b. Unpack the “AttrRep_Viewer” package to a location of your choice. This location will be referred to as “<attrrep_viewer_path>”
- c. Export the actual state of the ADOxx library in xml format:
“Library Management” -> “Libraries” -> “Management” -> “Export XML...”



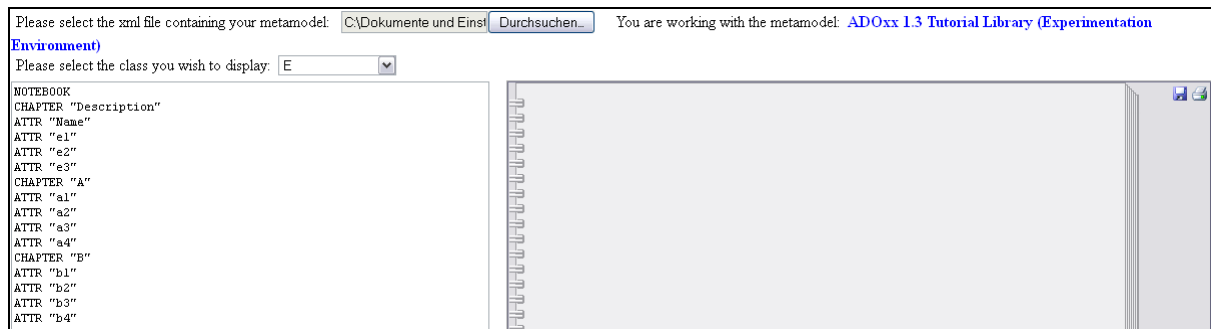
- d. Open the AttrRep Viewer by opening the html file **AttrRep_Viewer.html**, located at <attrrep_viewer_path>
- e. Click on the “**Browse...**” button, select the XML file where you exported the library and then click “Open”



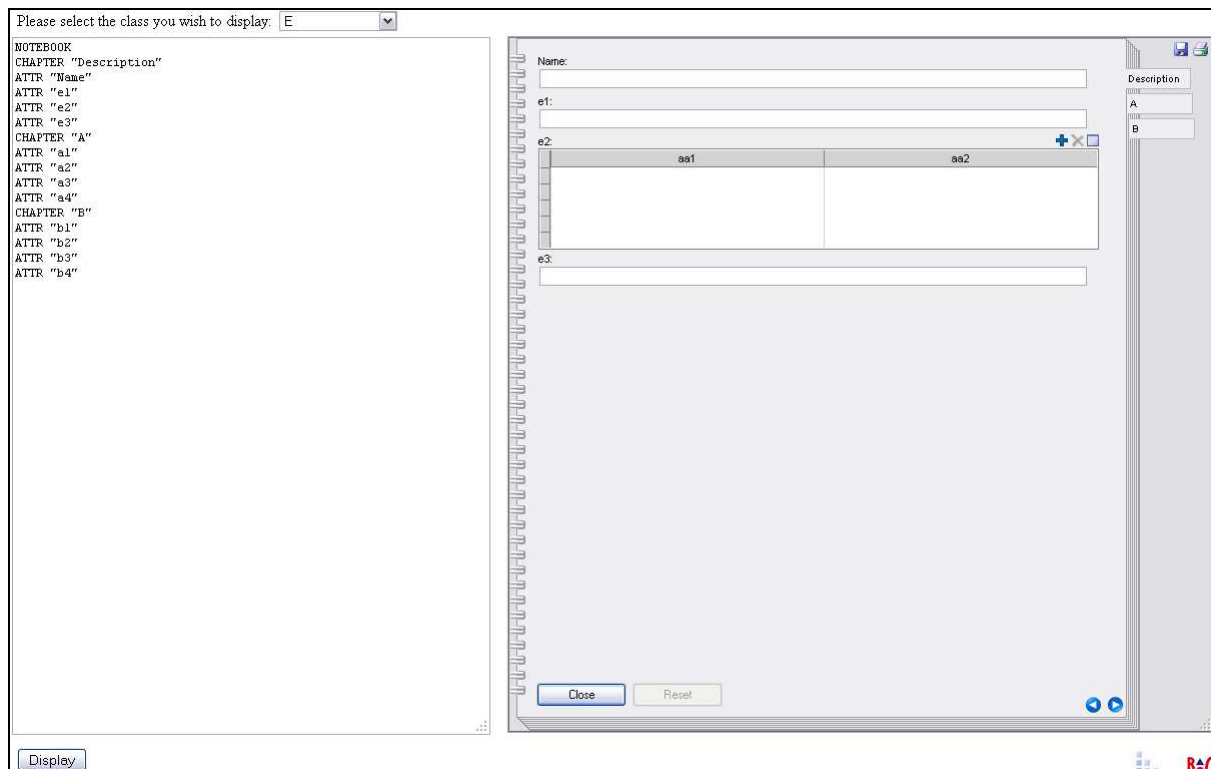
- f. You will receive the following confirmation message “The metamodel has been loaded!” and the full name of the library will be displayed on top of the page:



- g. In the drop-down list select the class for which you wish to preview the notebook and change the AttrRep; the contents of the AttrRep class attribute for the class you selected will be displayed in the text area to the left



- h. Click the "Display" button at the bottom left of the page and the preview of the notebook for the selected class (and AttrRep) will be displayed on the panel to the right:



- i. Click on each tab, to display the respective chapter or click the left and right arrows (bottom right) to browse through chapters and their pages.
- j. You can change the contents for the AttrRep class attribute and use the "Display" button to preview the appearance of the notebook after each change.

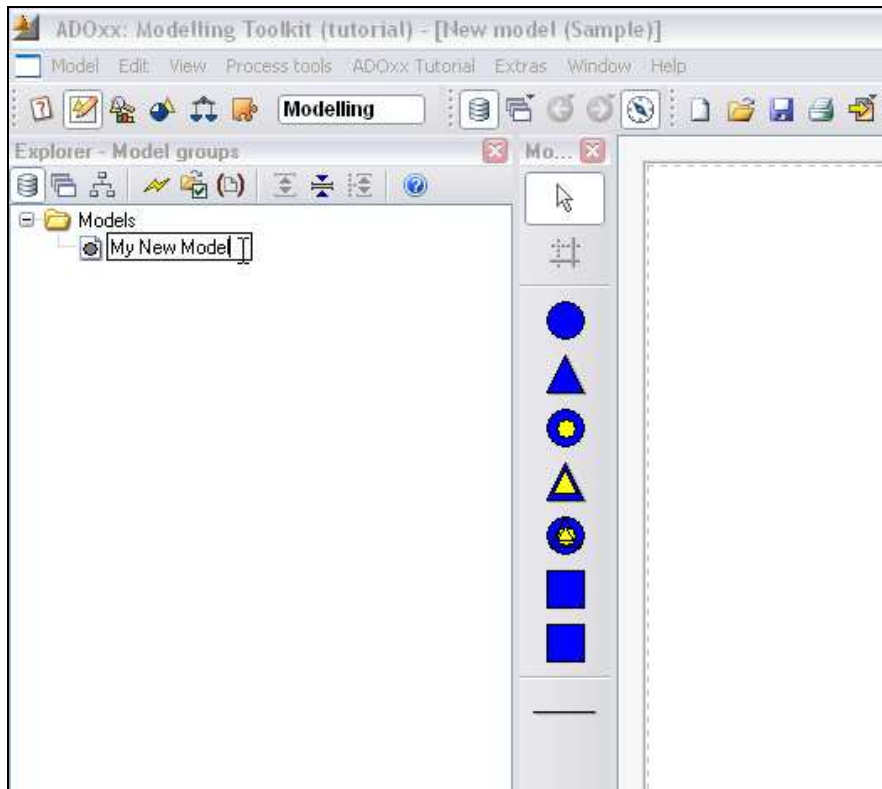
- k. When you have achieved the desired form of the notebook, select the AttrRep text from the text area, copy it, in the ADOxx Development Toolkit open the AttrRep class attribute of the class you selected, paste the text there and save the library.
- l. Repeat the steps above (g. – k.) for each class in the library.

Phase 3. TESTING THE MODELLING LANGUAGE

1. Open the ADOxx Modelling Toolkit
2. Right click on the model group "Models" and select the model type "Sample"



3. All the classes and relation classes you mentioned in the definition of the model type (Phase 2. step 16) are displayed in the Modelling bar
4. Enter a name for the newly created model



5. To add an object to the model, click on the class in the modeling bar and then click on the model canvas.