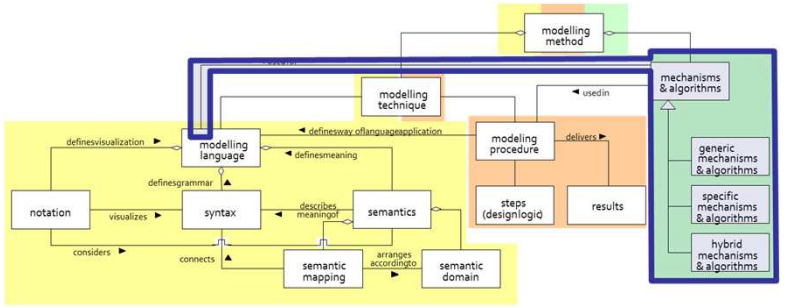


Configuration/Manipulation of Liferay Pages

SCENARIO:

Model Driven Configuration/Manipulation of External Systems

Scenario Description

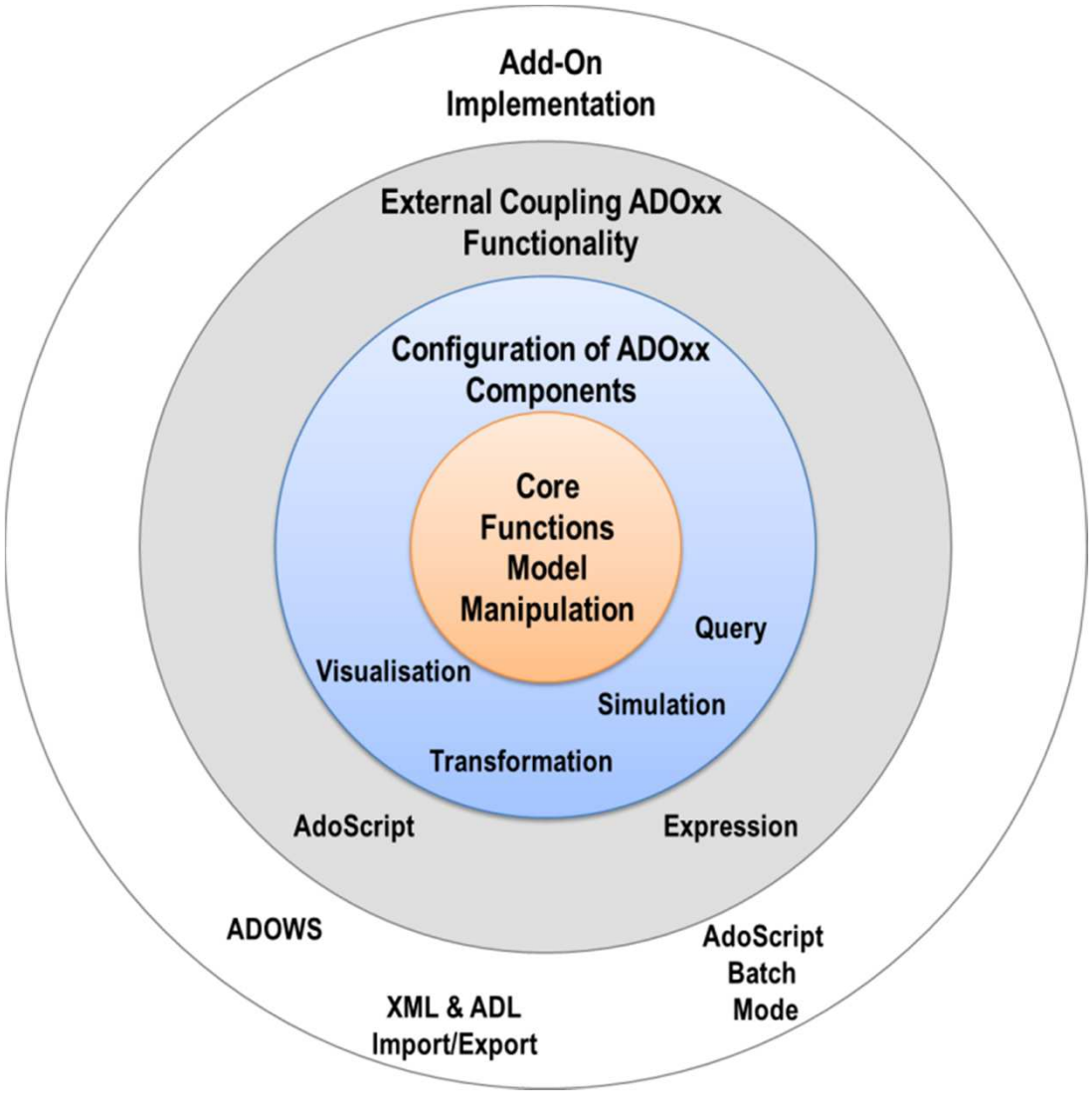
Case:

Configuration/Manipulation of Liferay individual pages through the configuration/manipulation of space models with invoking a specific Liferay API

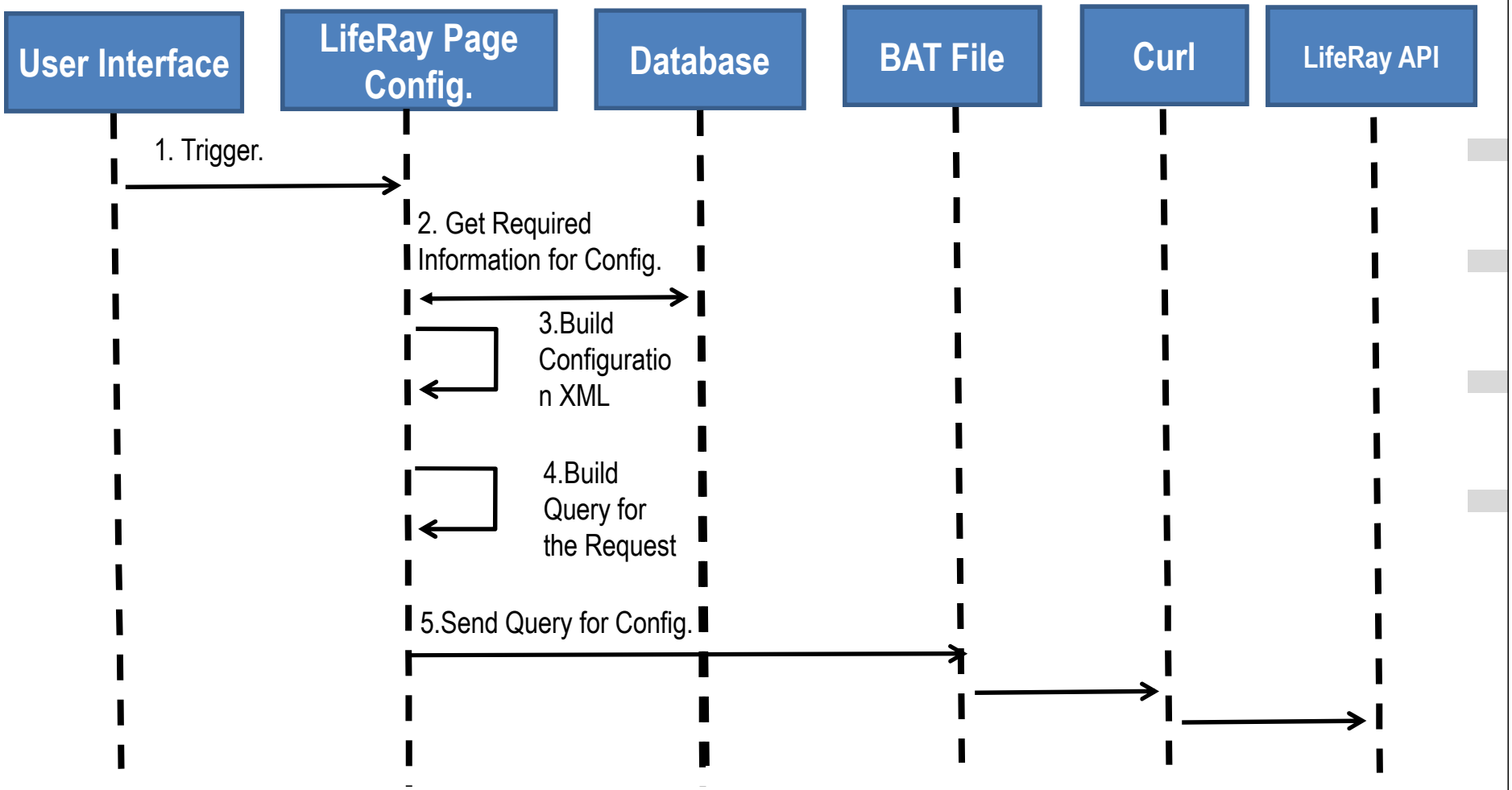
GOAL:

Demonstrate how to configure external system with using models

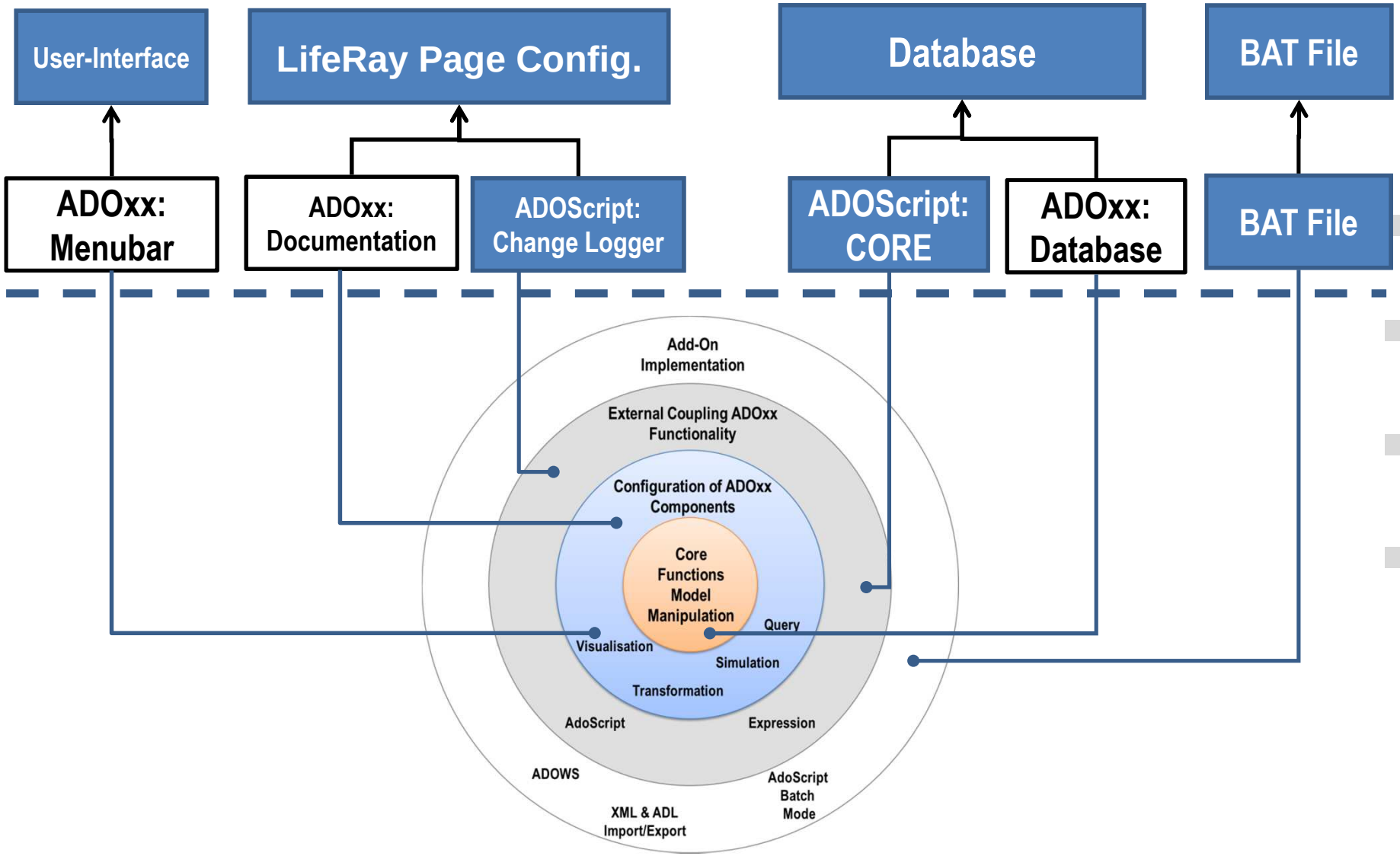
ADOxx Functionality on Meta Level



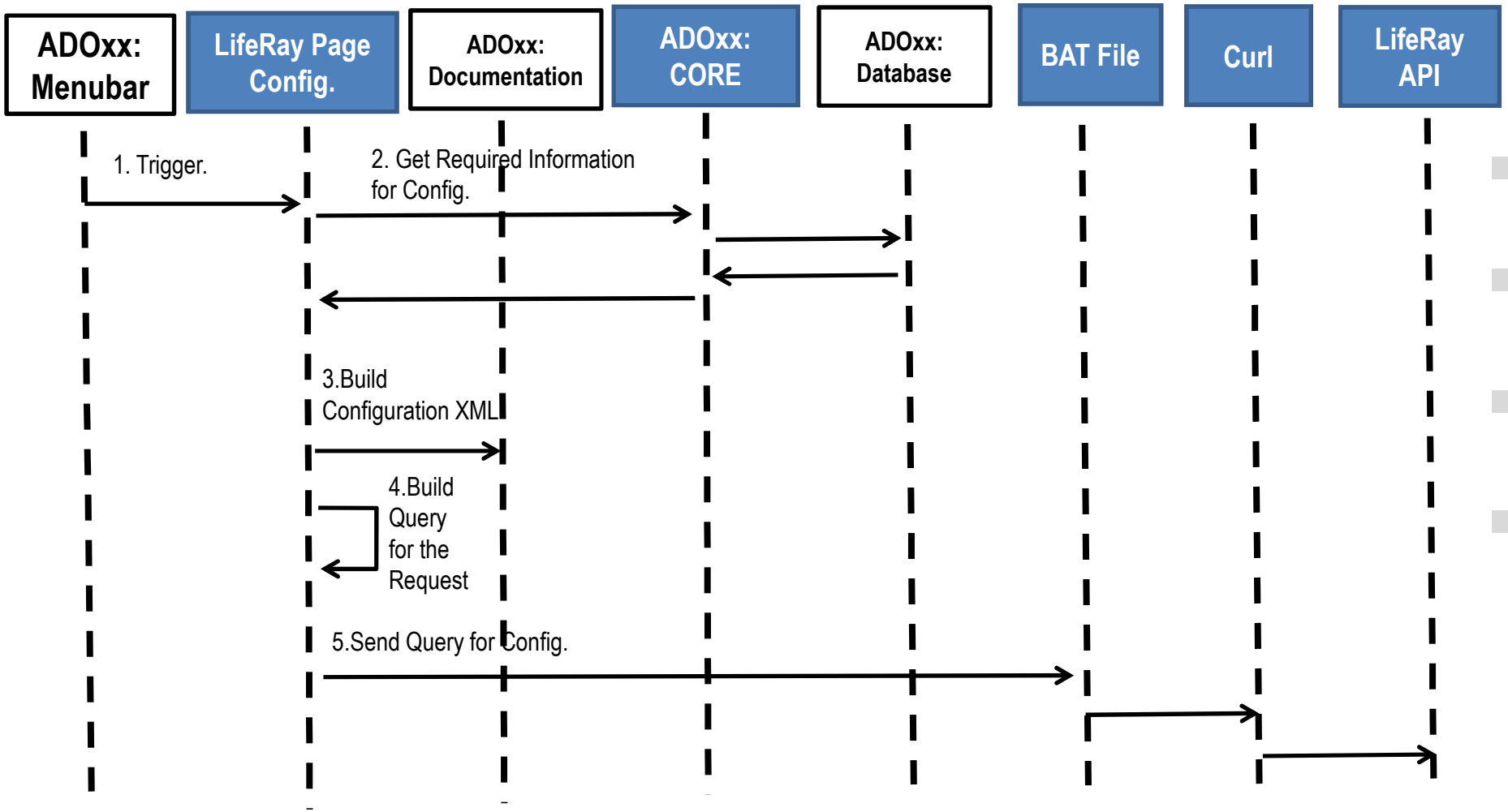
Description of Algorithm



Mapping ADOxx Functionality



ADOxx Realisation Approach



Added Value of Metamodelling Platform

Used meta-modelling functionality for realisation of the scenario:

- **ADOScript:** ADOScript can retrieve model information, sends request to the API
- **ADOxx Visualisation Component:** is provided by the platform and enables configuration of the user interface of model editor
- **ADOxx Documentation Component:** is provided by the platform and enables transformation between formats.
- **ADOScript Service:** ADOScript Service: ADOScript Service listens a certain port to get and interpret requests

ADOxx Realisation Hands-On

1. Modelling Language

1. Model Types “Space Model” and “Portlet Pool Model”
2. New class “Space Actor” and “Portlet”
3. Add Attributes

2. Configure ADOxx

1. Configure Space Model AttrRep

3. Implement Algorithm with ADOscript

1. LifeRay Page Configurator

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

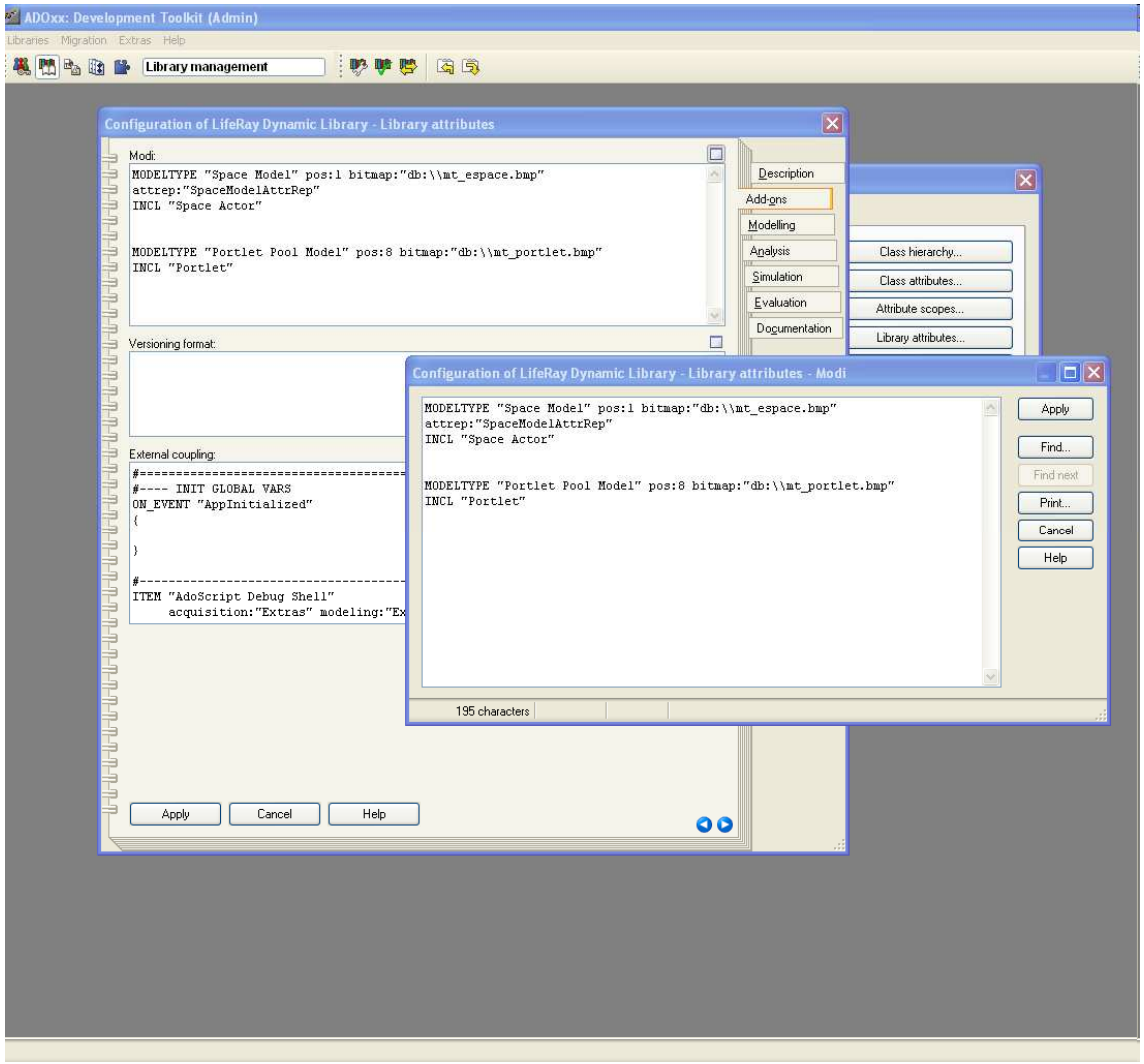
- Mechanisms & Algorithms Implementation
  - **Core Functions for Model Manipulation**
    - Database
    - **Visualisation**
    - Query
    - **Transformation**
  - Configuration of ADOxx Components
    - Visualisation
    - Query
  - **External Coupling ADOxx Functionali**
    - **ADOscript Triggers**
    - 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

Configuration/Manipulation of Liferay Pages

SCENARIO:
Model Driven Configuration/Manipulation of External Systems

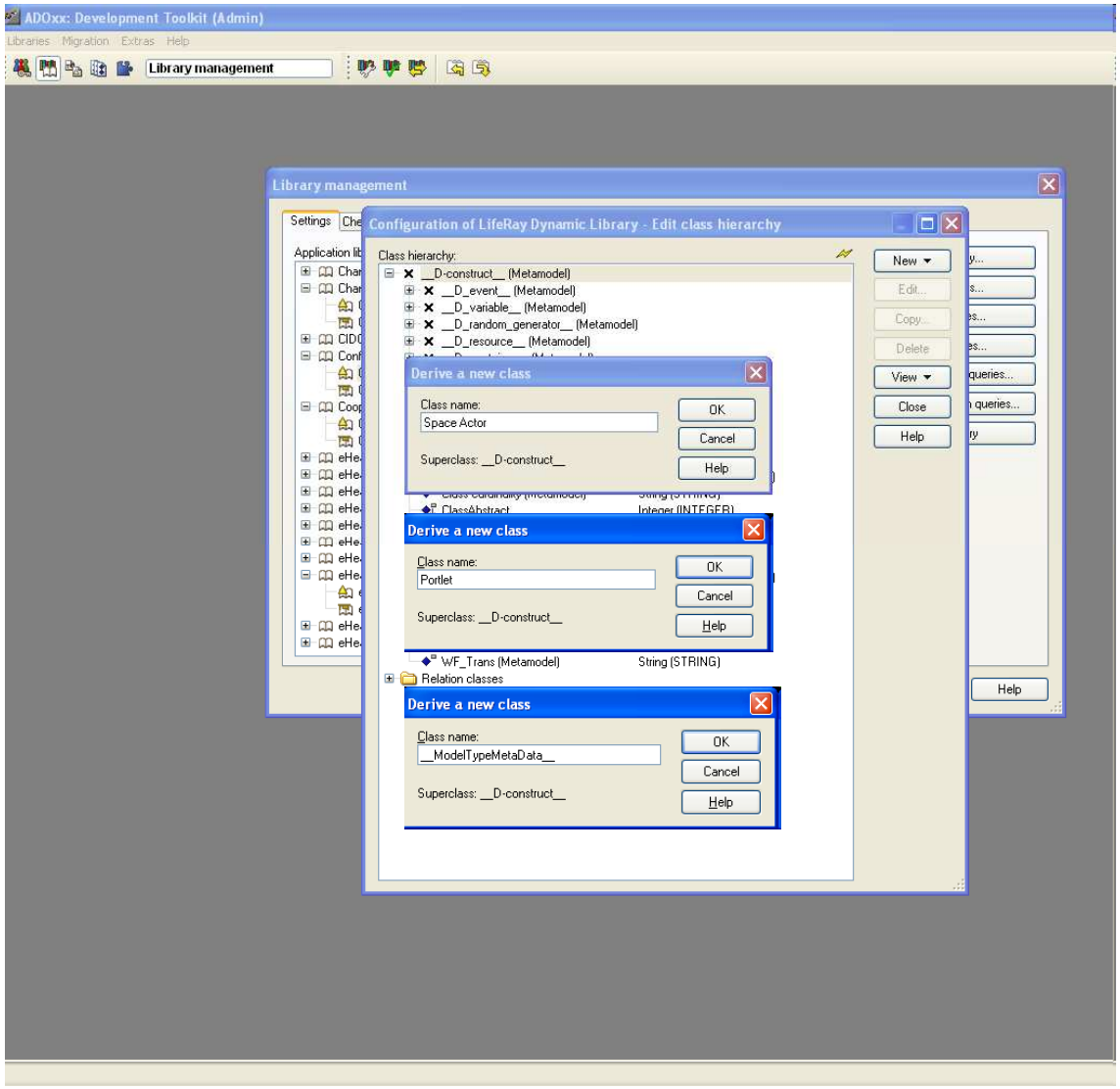
Define new Modeltypes Space Model", "Portlet Pool Model"



New Modeltypes:

- Select "Configuration of Liferay Dynamic Library" and open Library attributes.
- Got to Add Ons
- Add the Modeltypes "Space Model" and "Portlet Model" in the Modi attribute
- When the classes are defined, you need to INCLUDE "Space Actor" under "Space Model" and "Portlet" under " Portlet Pool Model"

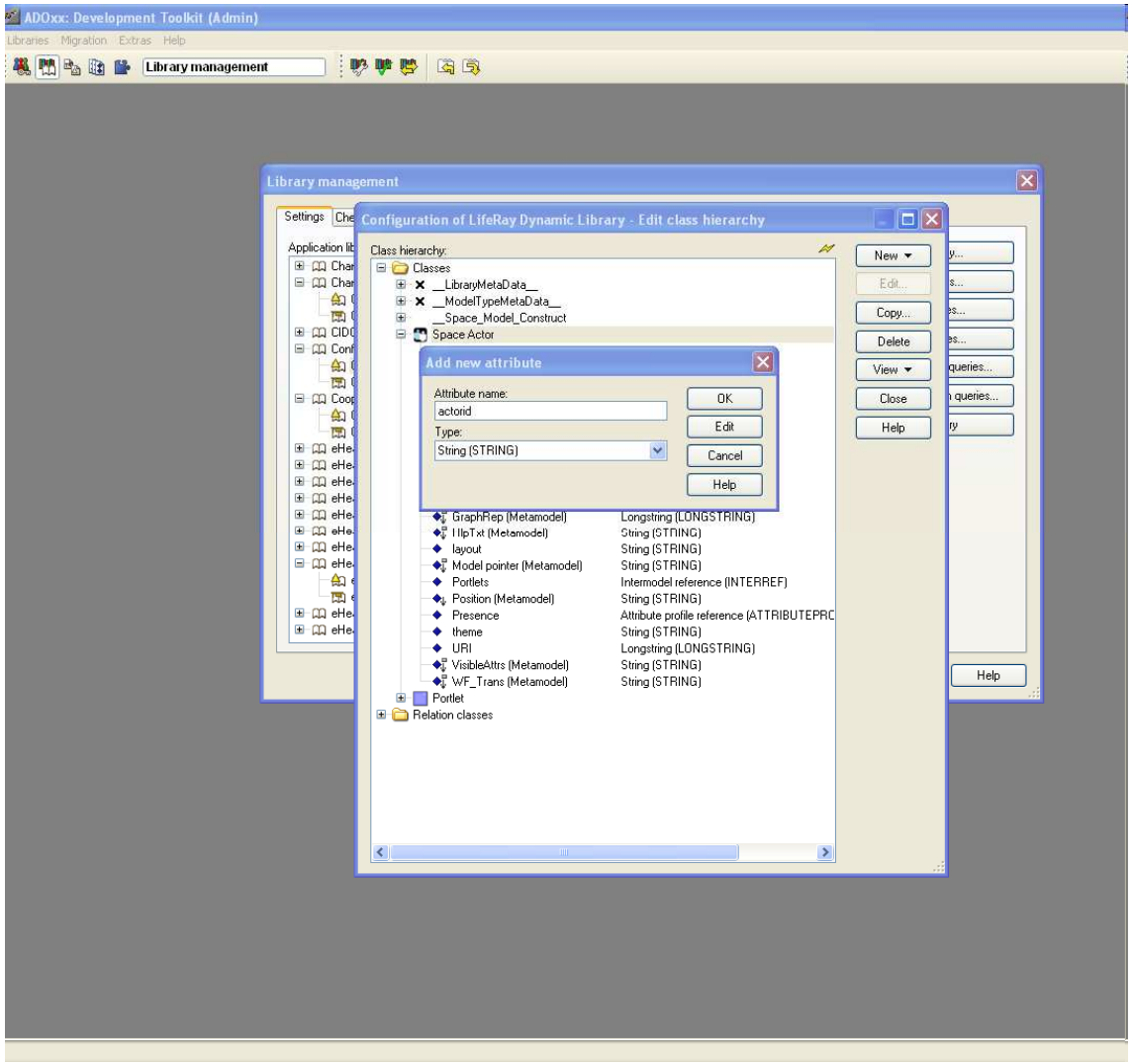
Create New Classes



Create New Classes

- Select “Configuration of Liferay 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: “Space Actor”, “Portlet” and ”__ModelTypeMetaData__”
- “Space Actor”, “Portlet” and ”__ModelTypeMetaData__”
- are now sub-classes of __D-construct__

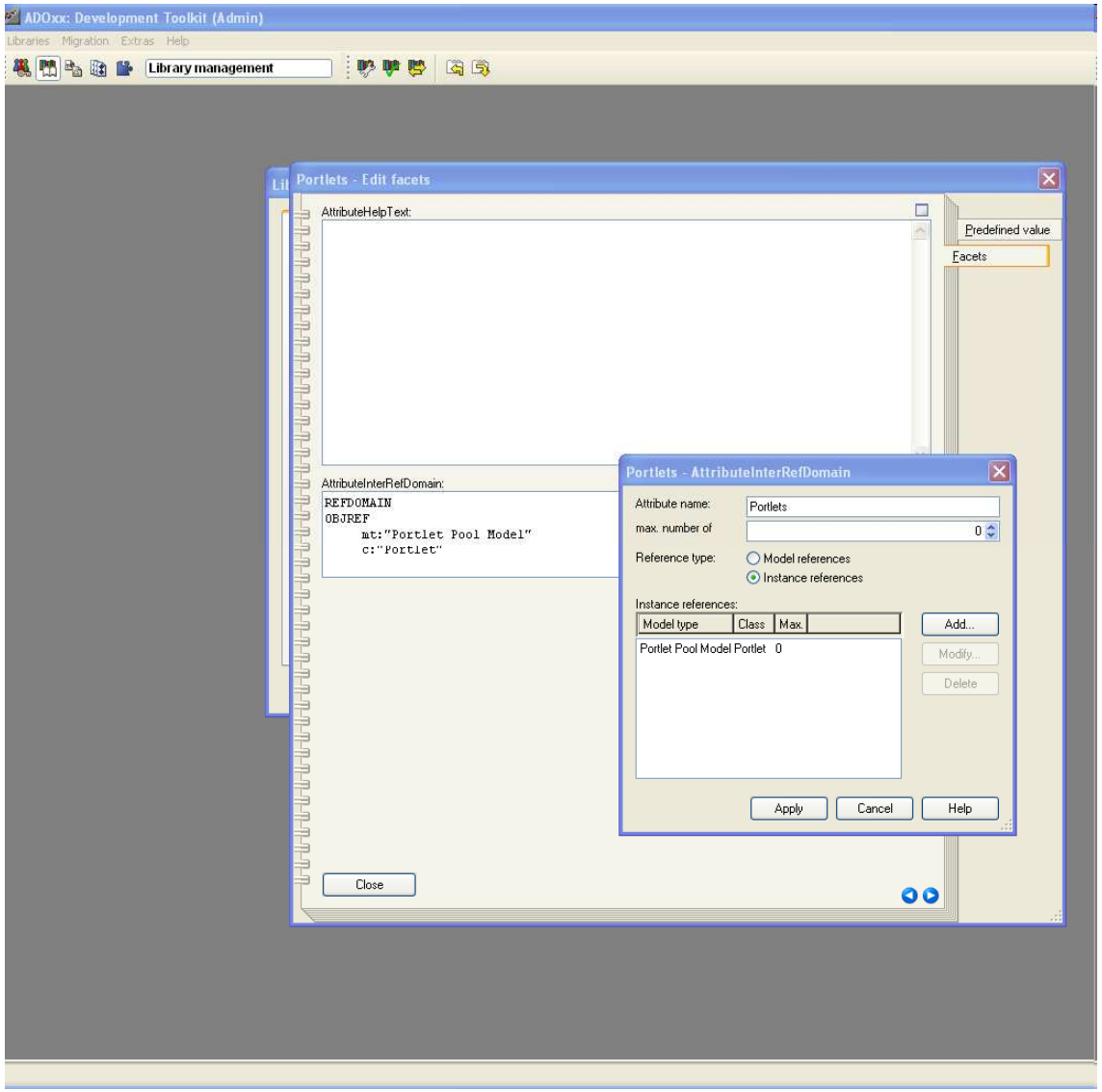
Add Attributes for Classes



Add Attributes

- Select “Space Actor” and click Newattribute.
- Make “actorid”, “theme”, “layout” as type STRING.
- Make “Portlets” as type INTERREF
- Select “Portlet” and click New, attribute.
- Make “Portlet ID” as type STRING “Configuration” as type LONGSTRING.

Edit INTERREF



Specification of INTERREF"Portlets"

- EDIT Facet
- Select AttributeInterrefDomain
- Select "Instance References" and click and select Portlet Pool Model
- Select Portlet

Implement and Import ADOscript File into Database

createLiferayPages.asc (please find whole code in the package)

```
#if C:\ is not accessible, the path where the temporary XML file is saved should be changed here
SETG sXMLFileName: "C:\\liferay_config.xml"

CC "Modeling" GET_ACT_MODEL
SET n_act_space_modelid: ( modelid )
CC "Core" GET_CLASS_ID classname: ("Space Actor")
SET n_spaceactor_classid: ( classid )
#--> RESULT ecode: intValue attrid: id
CC "Core" GET_ATTR_ID classid: (n_spaceactor_classid) attrname: ("Portlets")
#--> RESULT ecode: intValue attrid: id
SET n_spaceactor_portlets_attrid: ( attrid )

CC "AQL" EVAL_AQL_EXPRESSION modelid: (n_act_space_modelid) expr: ("<\"Space Actor\">")
SET s_space_actor_objids: ( objids )

CC "Core" GET_ATTR_VAL objid: (n_act_space_modelid) attrname: ("WS Endpoint") as-string
SET s_space_model_wsendpoint: ( val )

FOR s_space_actor_objid in:(s_space_actor_objids) {
   SET id_space_actor_objid: ( VAL s_space_actor_objid  )

            CC "Core" GET_ATTR_VAL objid: ( id_space_actor_objid ) attrname: ("actorid") as-string
            SET s_space_actor_attr_actorid:( val )
            CC "Core" GET_ATTR_VAL objid: ( id_space_actor_objid ) attrname: ("theme") as-string
            SET s_space_actor_attr_theme:( val )
            CC "Core" GET_ATTR_VAL objid: ( id_space_actor_objid ) attrname: ("layout") as-string
            SET s_space_actor_attr_layout: ( val )
            #CC "Core"  GET_ATTR_VAL objid:(id_space_actor_objid) attrname:("portletids") as-string
            GET_PORTLET_IDS spaceactorobjid:( id_space_actor_objid ) s_portlet_ids: s_space_actor_attr_portletid
```

...

Results

Results

Further Questions?

www.adoxx.org

tutorial@adoxx.org