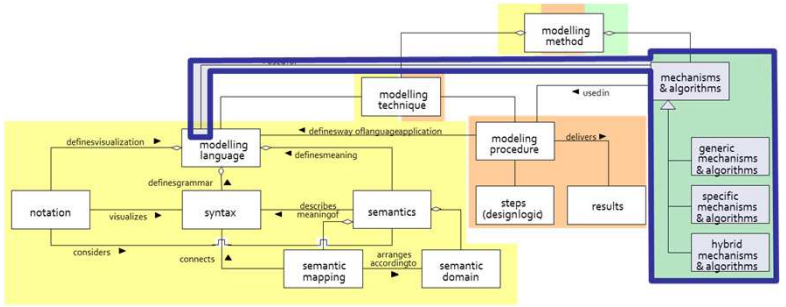


Export of OWL Models as RDF

SCENARIO:
Configuration of ADOxx Component

Scenario Description

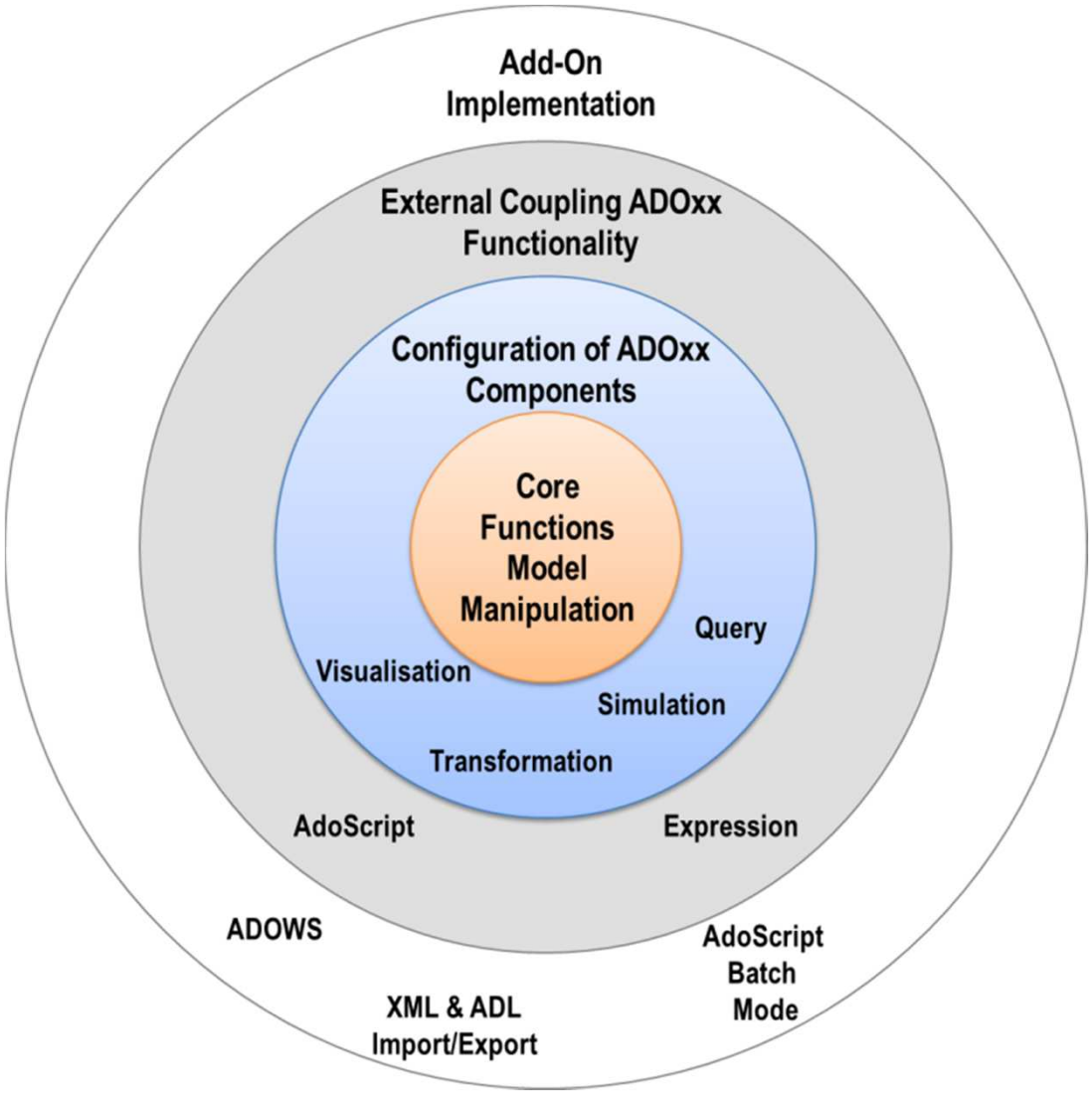
Case:

Export of OWL Model in RDF format via configuration of ADOxx Components. ADOscript invokes ADOxx Import/Export and ADOxx Documentation Components, establishes interaction with a XSLT Processor and XSLT Processor transforms OWL Model XMLs complying ADOxx XML Schema into RDF Files complying RDF Schema.

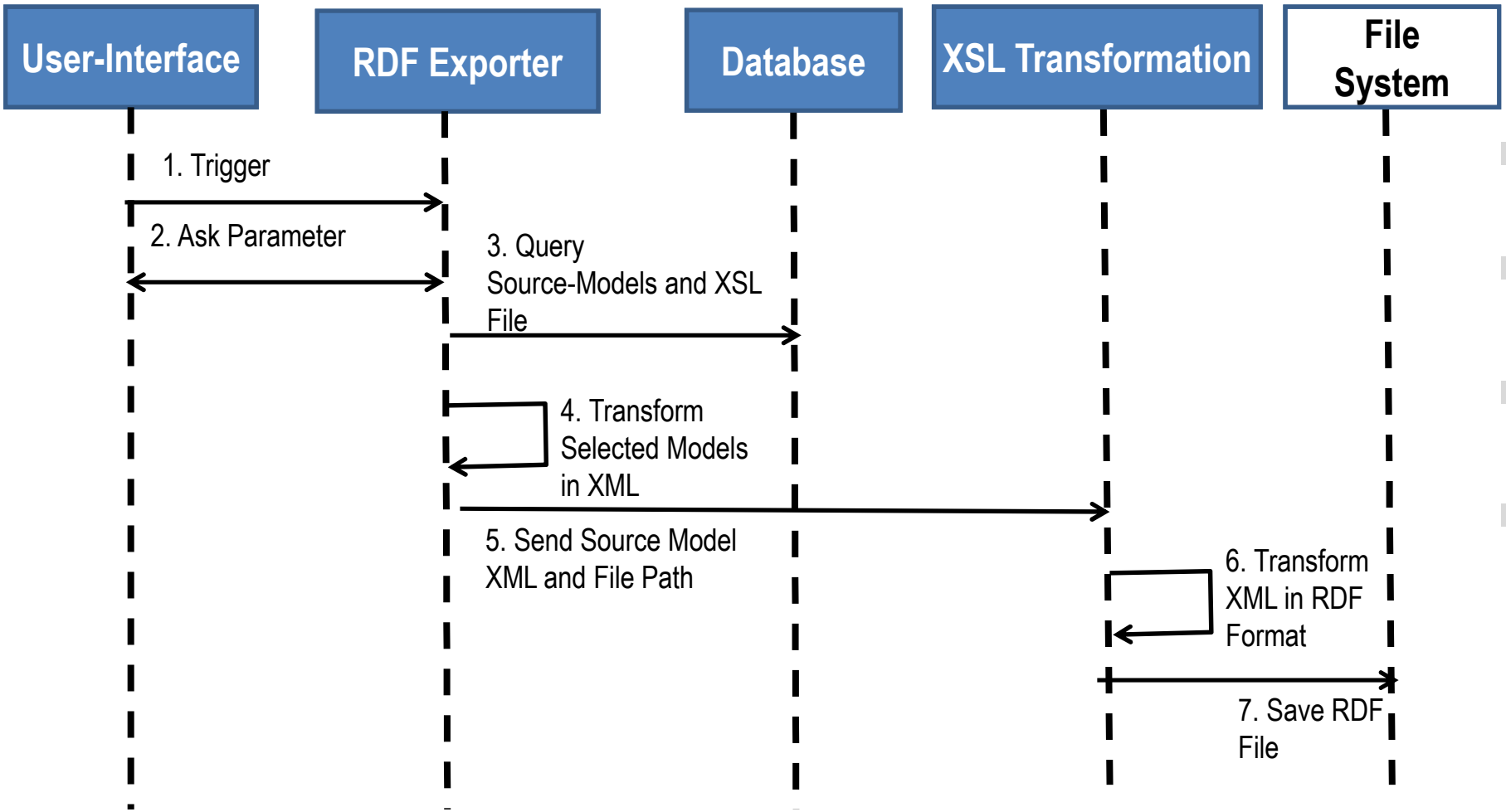
GOAL:

Demonstrate how to use AdoScript to invoke ADOxx Components and to establish interaction with a external system in order to transform Model format.

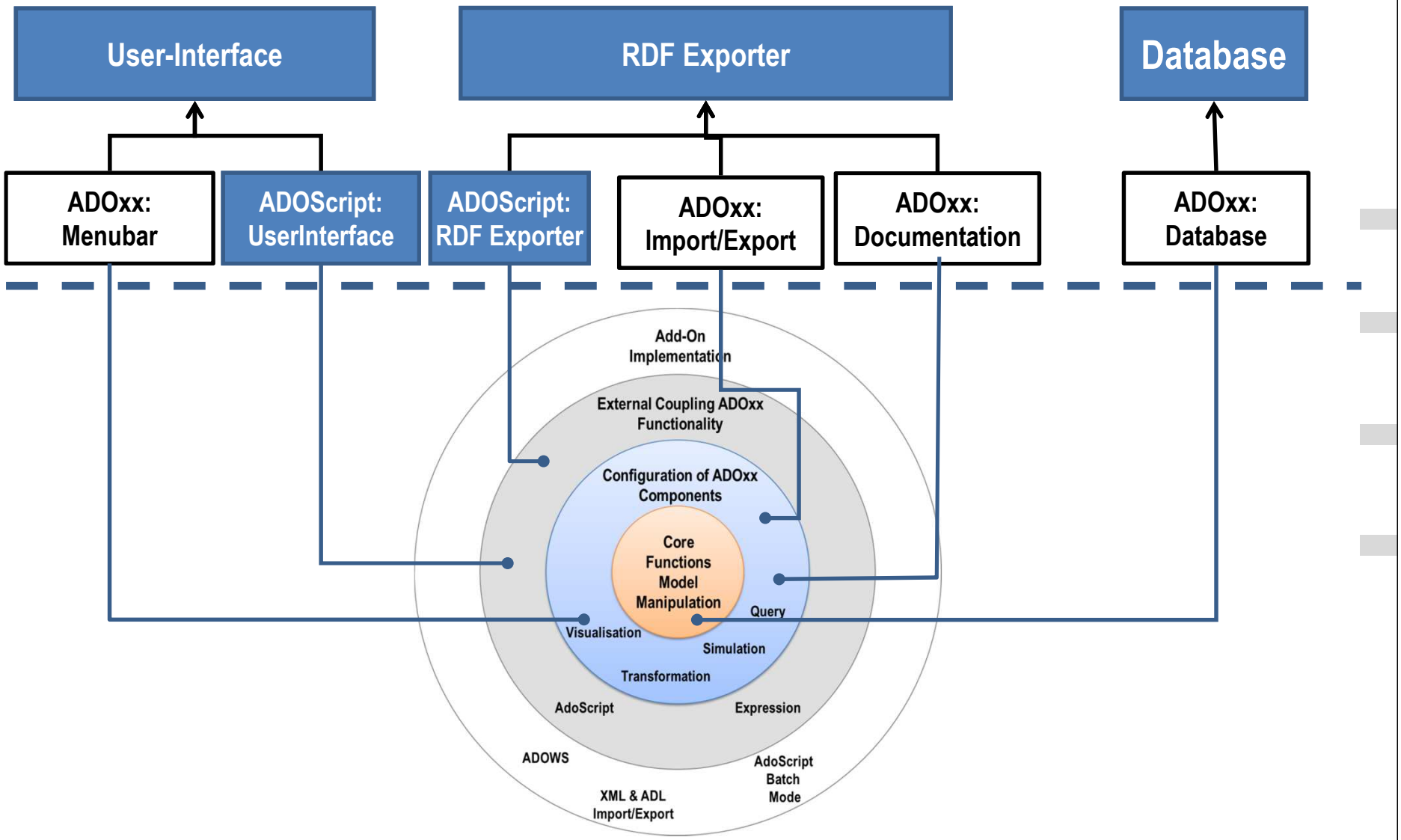
ADOxx Functionality on Meta Level



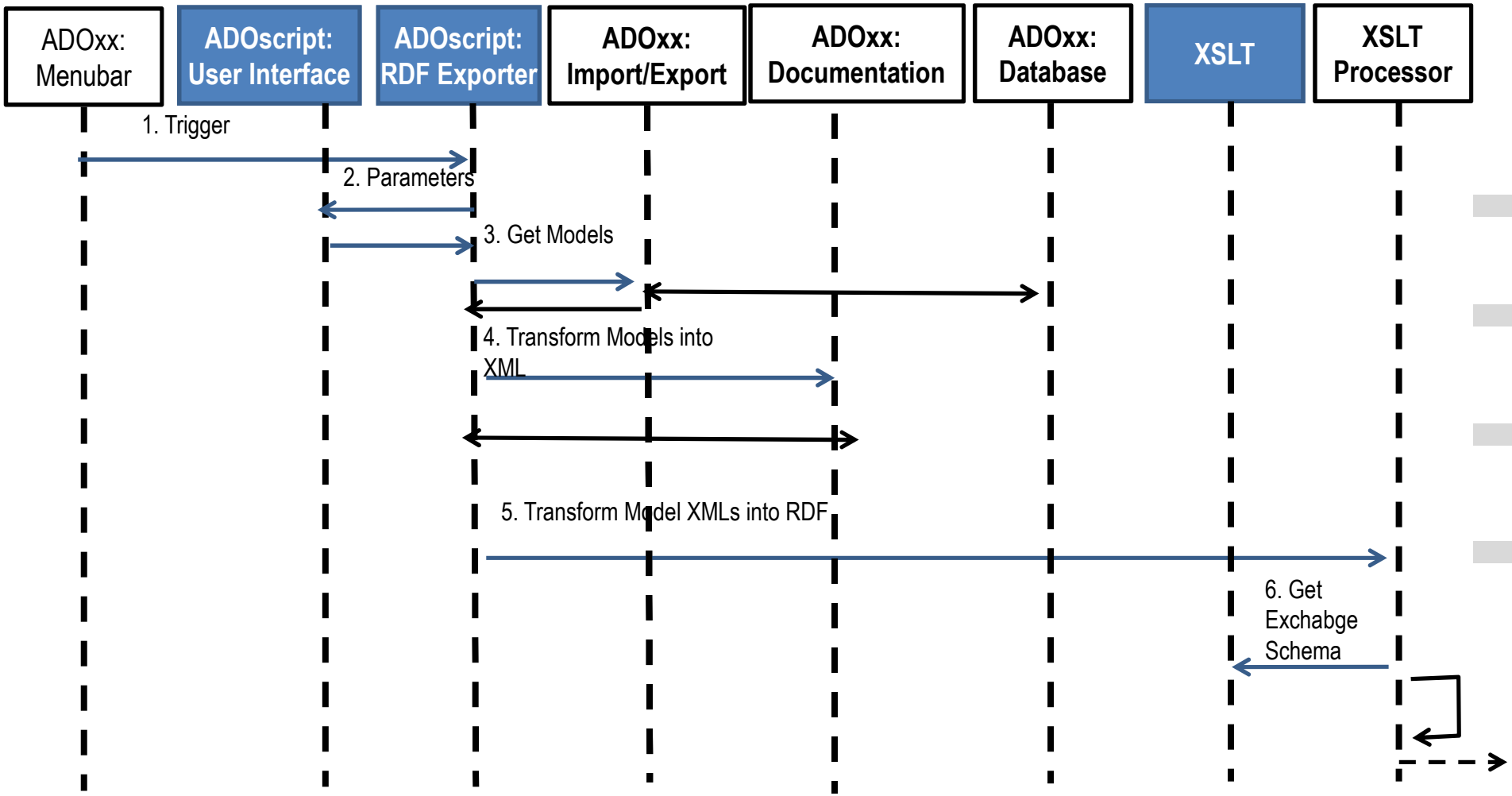
Description of Algorithm



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 and establish interaction between ADOxx and XSLT Processor.

- **ADOxx Visualisation Component:** is provided by the platform and enables configuration of the user interface of model editor

- **ADOxx Import/Export Component:**
  - **ADOxx Import/Export Component:** is provided by the platform and can retrieve models from database .
  - **ADOScripts** can invoke the ADOxx Import/Export Component
- **ADOxx Documentation Component**
  - **ADOxx Documentation Component:** is provided by the platform and can transform the models in required format, in our case in XML format
  - **ADOScripts** can invoke the ADOxx Documentation Component

ADOxx Realisation Hands-On

- **Implementation of XSL File**
- **Configure ADOxx**
  1. Configure Menubar

1. Implement Algorithm with ADOscript

1. ADOscript User Interface
2. Invoking Import/Export Component with ADOscript
3. Invoking Documentation Component with ADOscript
4. Invoking XSLT Processsor with ADOscript

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

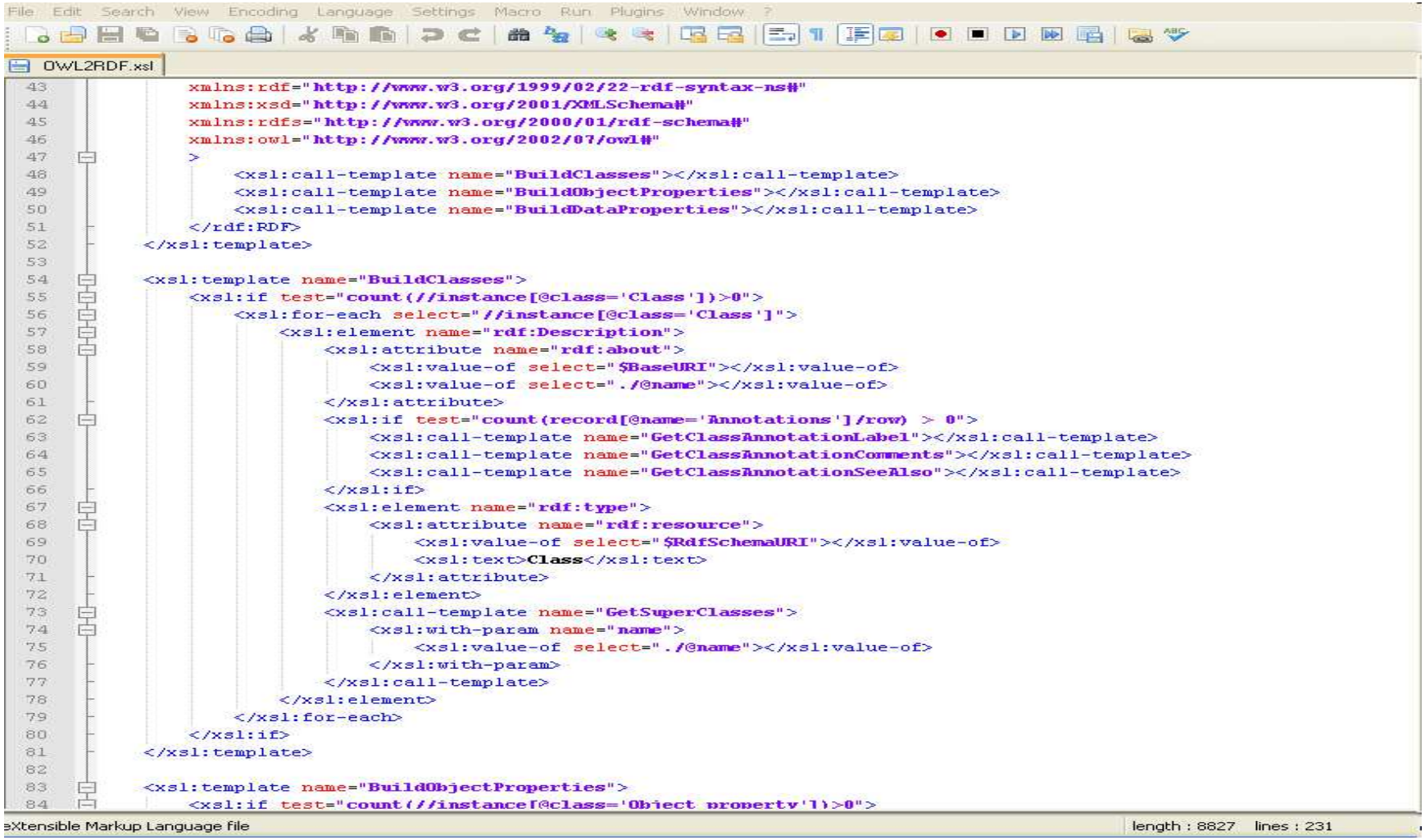
- Mechanisms & Algorithms Implementation
  - Core Functions for Model Manipulation
    - **Database**
    - Visualisation
    - Query
    - **Transformation**
  - Configuration of ADOxx Components
    - Visualisation
    - Query
  - **External Coupling ADOxx Functionality**
    - **ADOscript Triggers**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

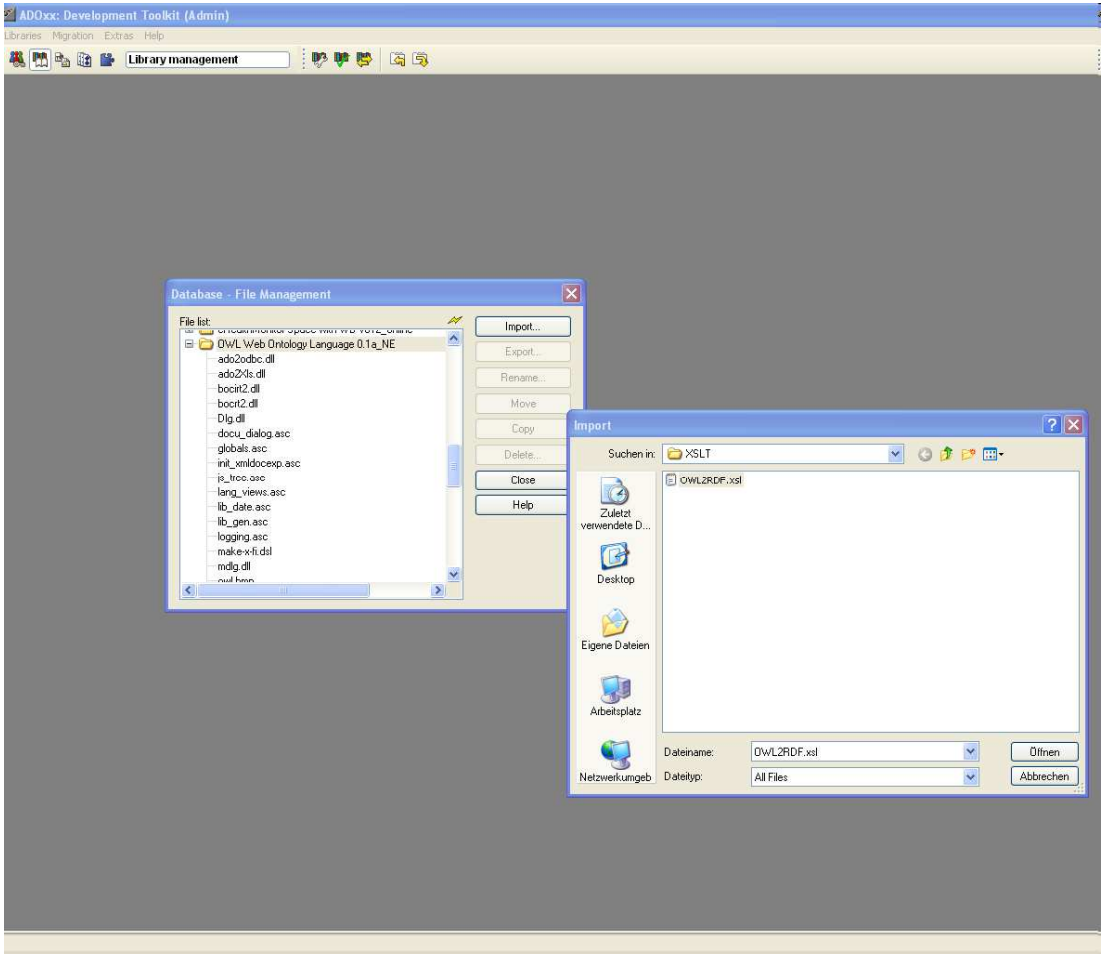
Export of OWL Models as RDF

SCENARIO:
Configuration of ADOxx Component

Implement XSL File



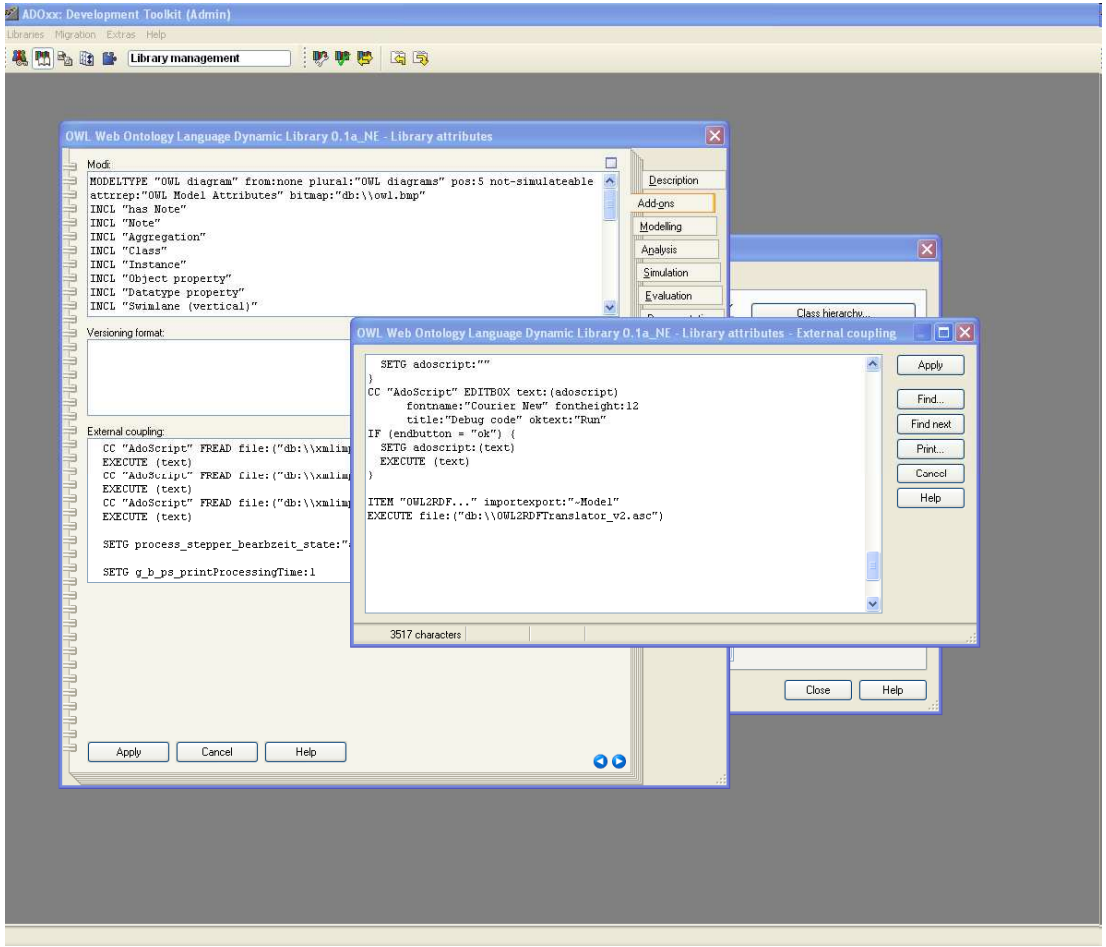
Copy XSL File into Database



Add Menubar

- Open Extras from Menubar
- Open File management
- Select ADOxx OWL Application Library
- Import XSL file

Add Menubar



Add Menubar

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

ITEM "OWL2RDF..." importexport:"~Model"
EXECUTE file:("db:\\OWL2RDFTranslator.asc")

Copy and Configure ADOscript

```
SET sXSLTfileName: "OWL2RDF.xsl"
SET sTempFileName: "__rdf_temp.xml"

#Show Export Dialog and Invoke Import/Export Component
CC "ImportExport" SHOW_EXPORT_DLG mode: "xml" title: "XML_MODELS export"
filedescription: "XML files" fileextension: "*.xml"

IF (endbutton = "ok") {

 SET sModelIDs: (modelids)
 SET sModelGroups: (mgroupids)
 SET sOutFilename: (filename)

 SET nPosFileName: (bsearch ( sOutFilename , "\\" , (LEN sOutFilename)-1 ))
 SET sExportFolder: (copy ( sOutFilename , 0 , nPosFileName+1 ))
 SET sXSLTfilePath: (sExportFolder + sXSLTfileName)
 SET sTempFilePath: ( sExportFolder + sTempFileName)
 CC "AdoScript" FILE_COPY from: ("db:\\" + sXSLTfileName) to: (sXSLTfilePath)

#Invoke Documentation Component
 CC "Documentation" XML_MODELS modelids: (sModelIDs) mgroupids: (sModelGroups)
 attrprofs: (attrprofids) apgroups: (apgroupids)

 ...
```

Further Questions?

www.adoxx.org

tutorial@adoxx.org