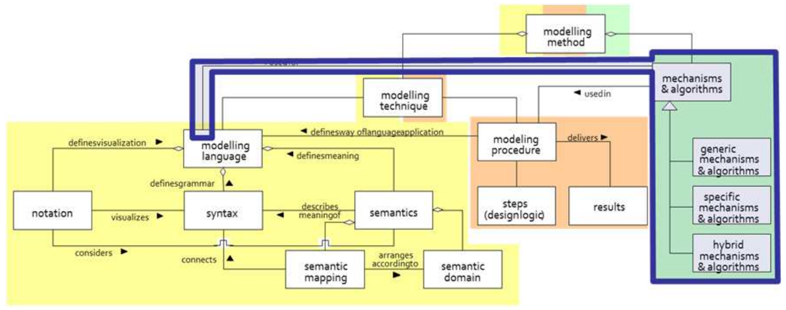


Export of OWL Models as RDF

SCENARIO:
Configuration of ADOxx Component

Scenario Description

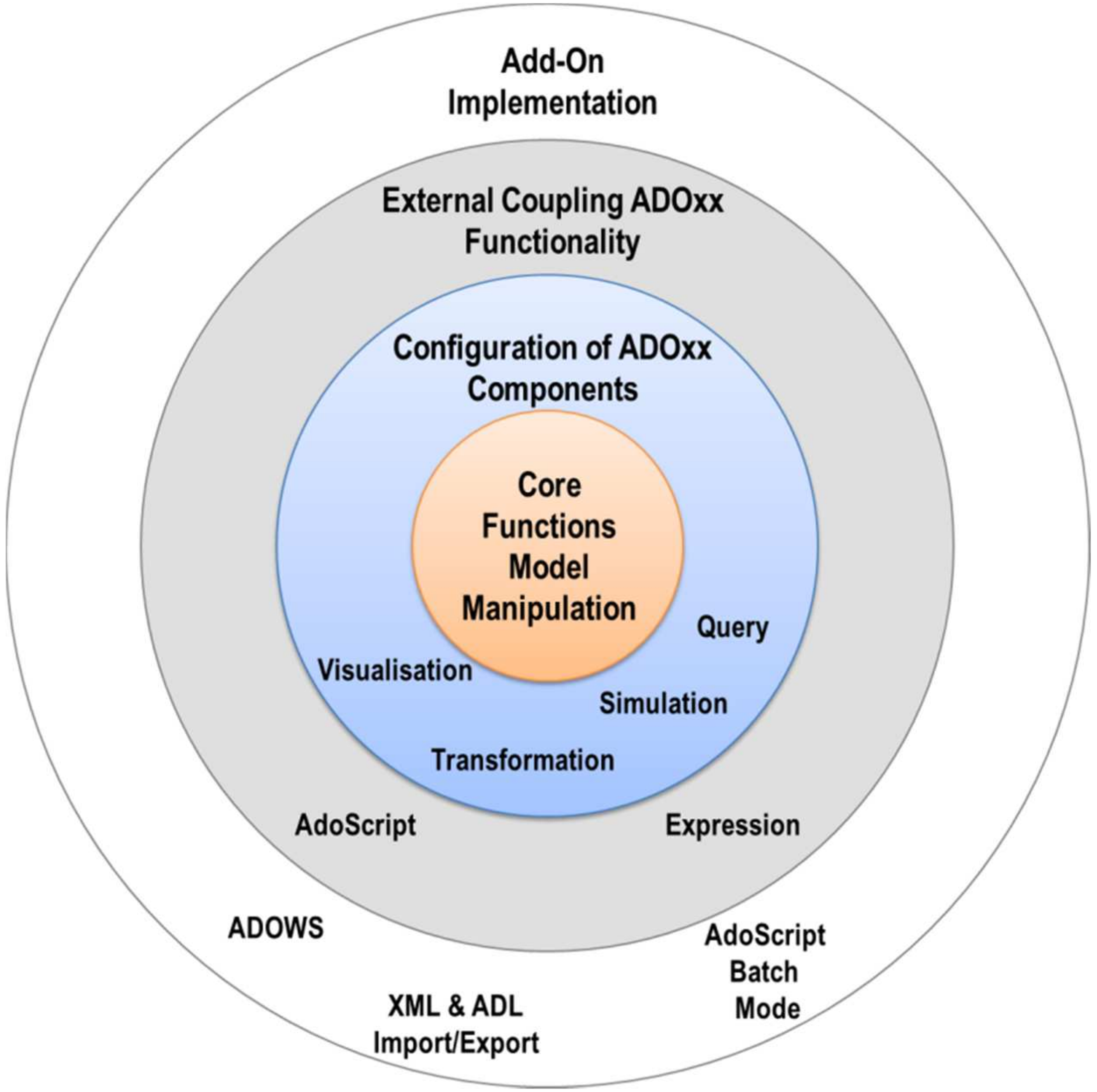
Case:

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.

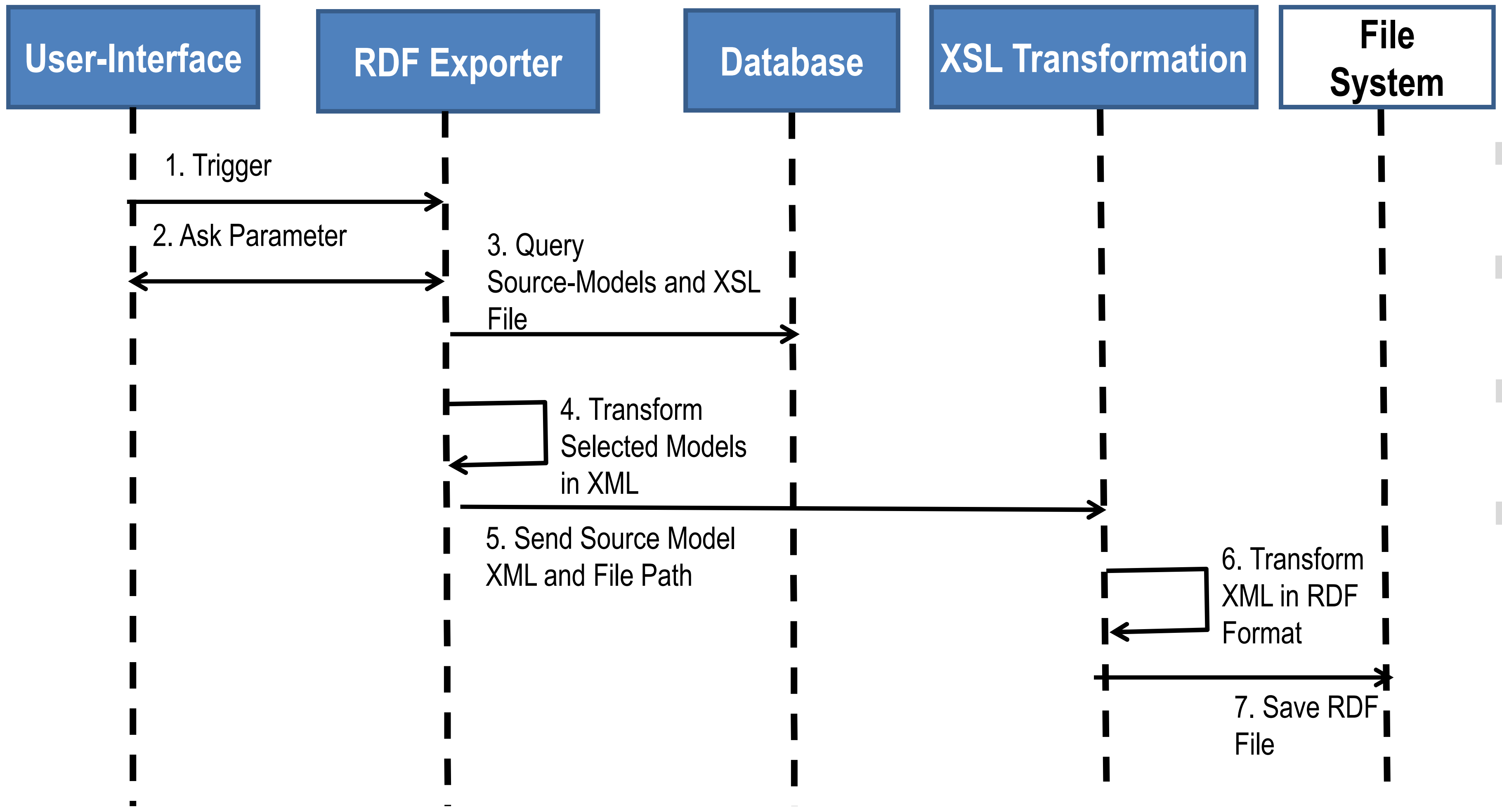
GOAL:

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

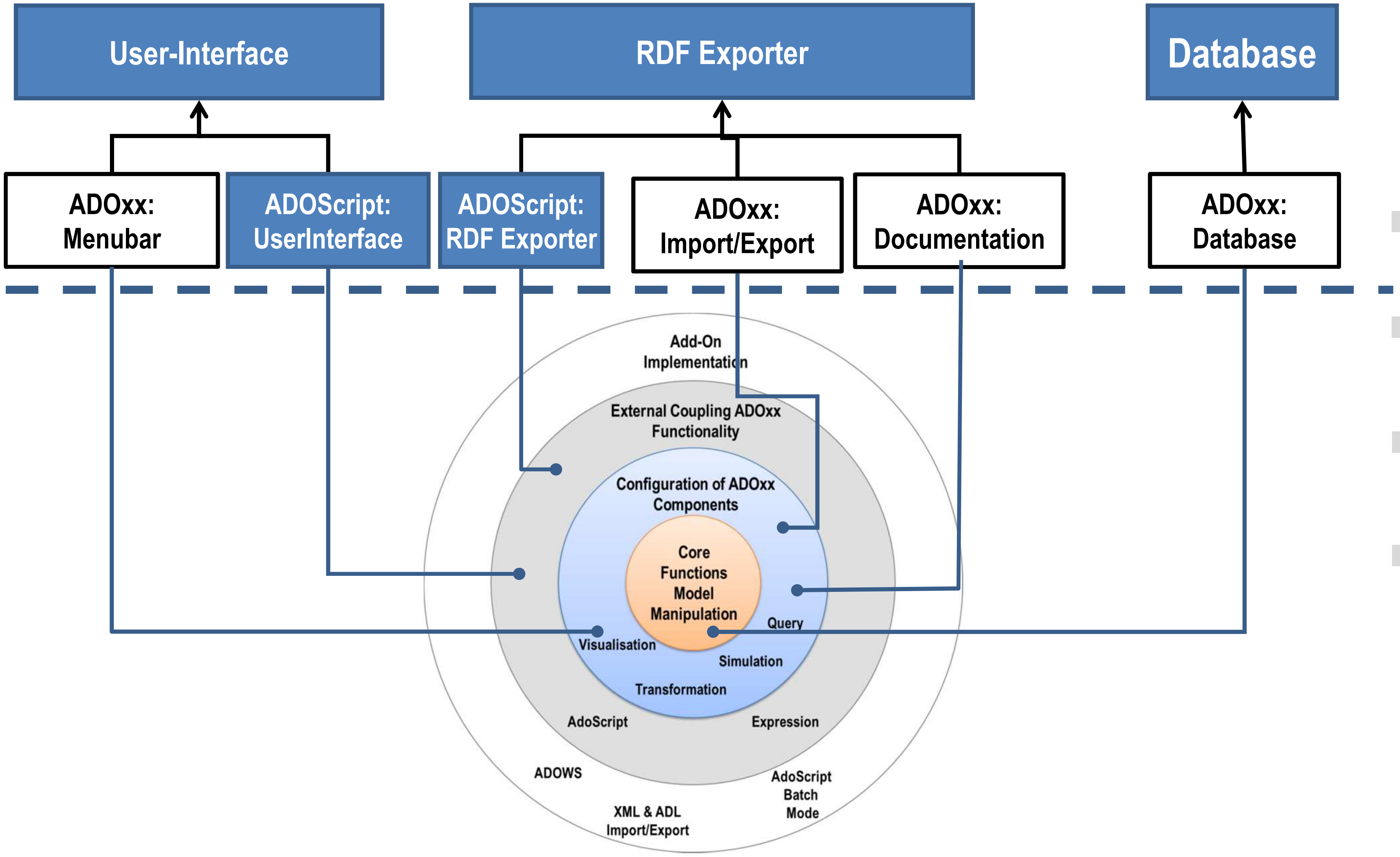
ADOxx Functionality on Meta Level



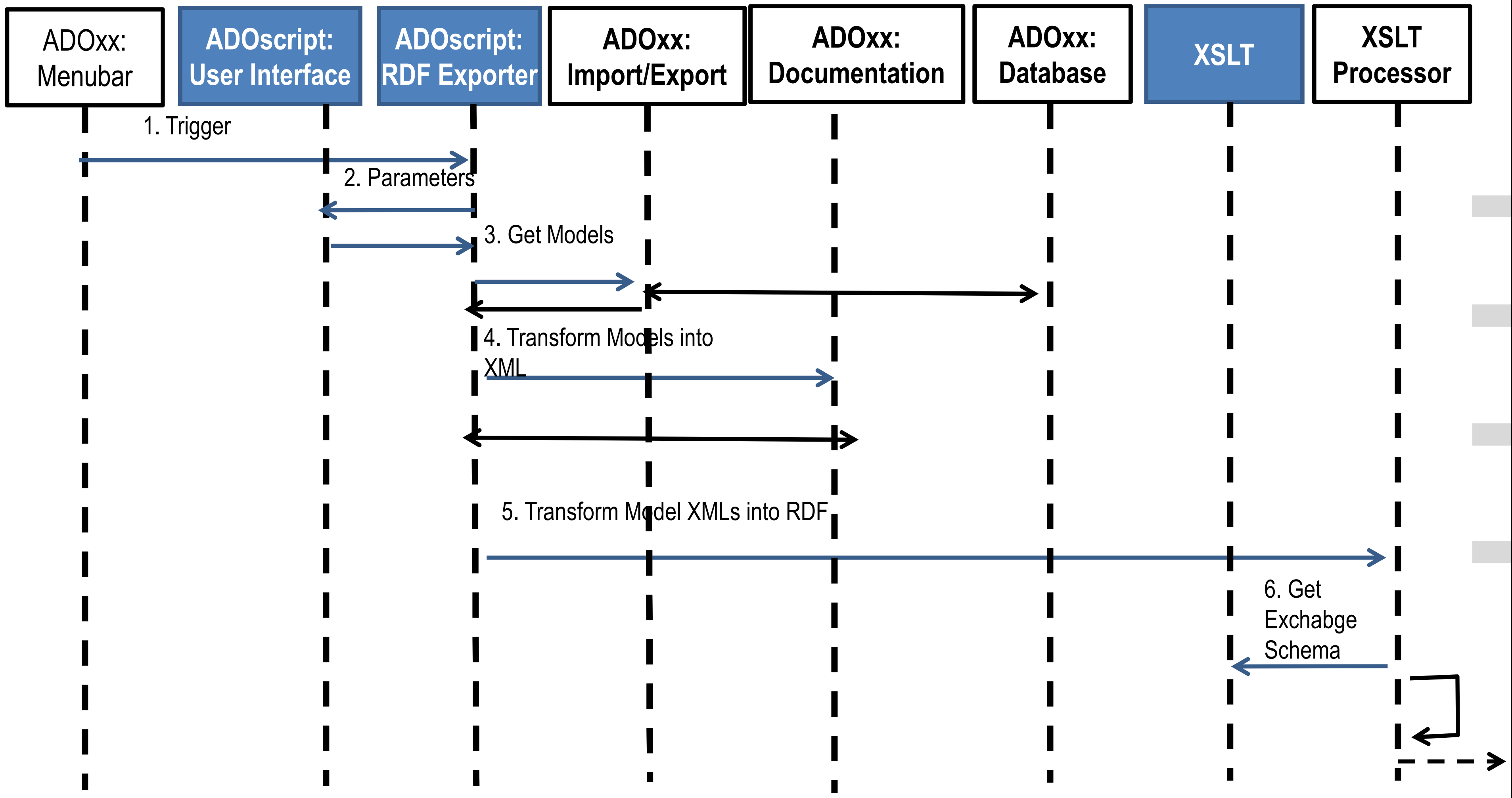
Description of Algorithm


User-Interface
RDF Exporter
Database
XSL Transformation
File System
1. Trigger
2. Ask Parameter
3. Query Source-Models and XSL File
4. Transform Selected Models in XML
5. Send Source Model XML and File Path
6. Transform XML in RDF Format
7. Save RDF File

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

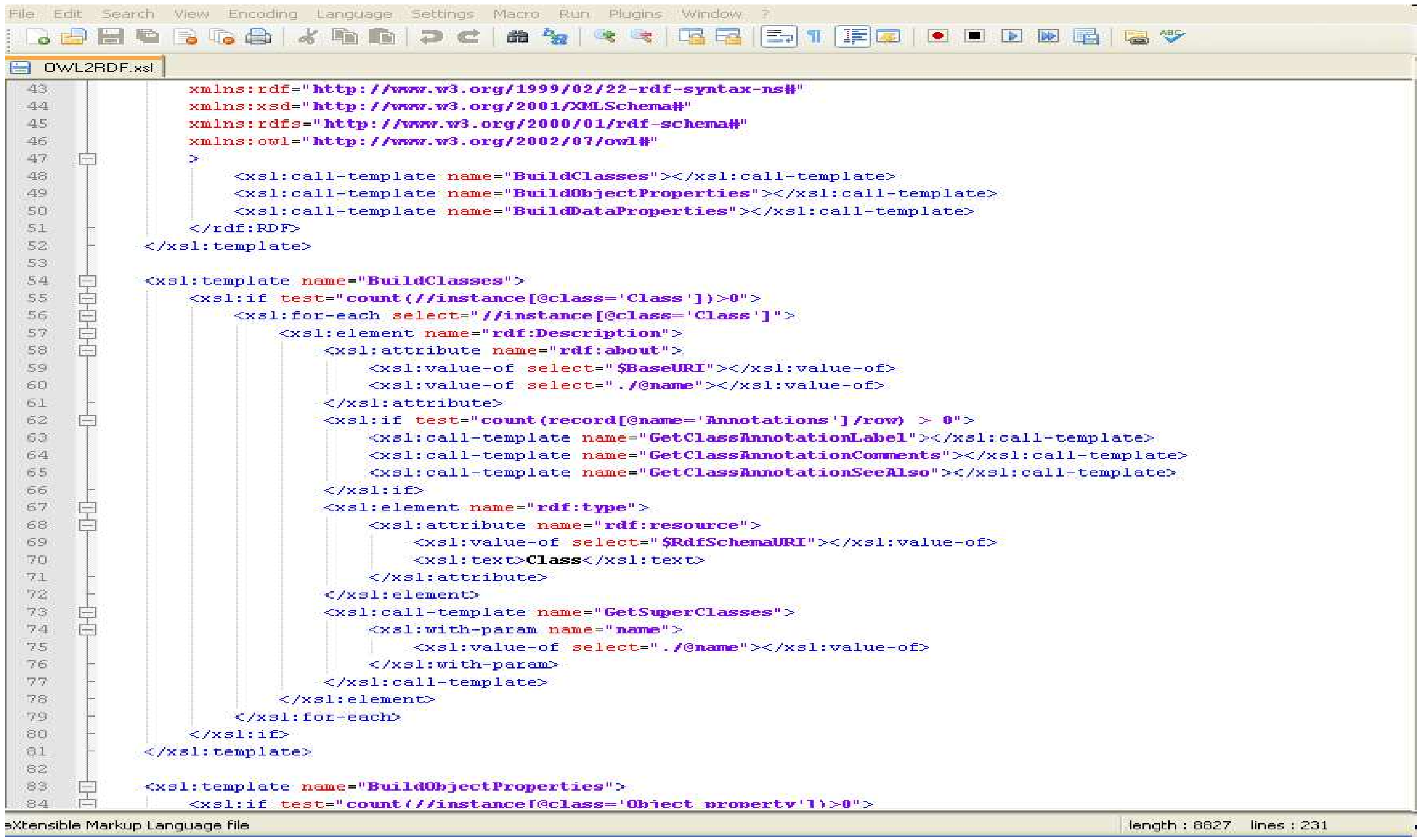
Used ADOxx Functionality: Implementing an Algorithm

HANDS-ON

Export of OWL Models as RDF

SCENARIO:
Configuration of ADOxx Component

Implement XSL File


File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
OWL2RDF.xsl

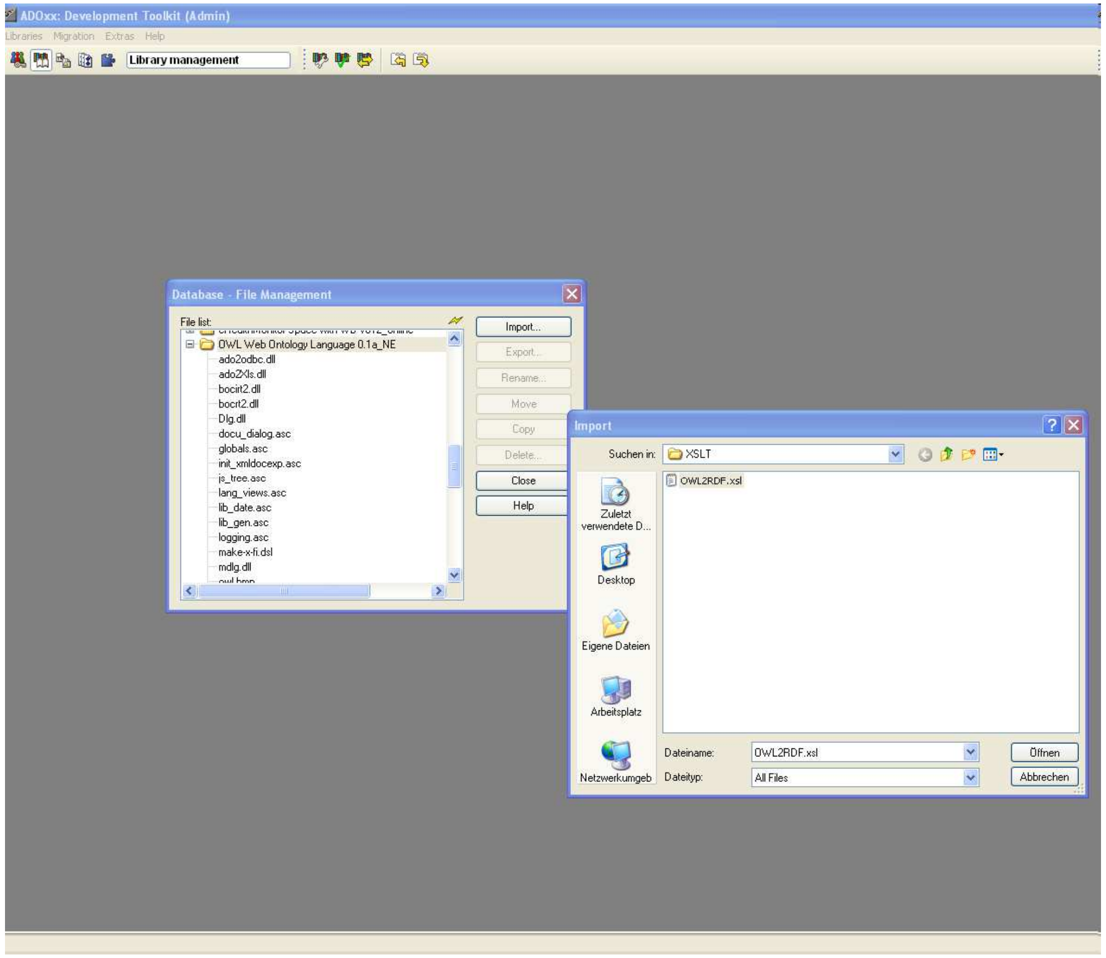
```
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        >
            <xsl:call-template name="BuildClasses"></xsl:call-template>
            <xsl:call-template name="BuildObjectProperties"></xsl:call-template>
            <xsl:call-template name="BuildDataProperties"></xsl:call-template>
        </rdf:RDF>
    </xsl:template>

    <xsl:template name="BuildClasses">
        <xsl:if test="count(//instance[@class='Class'])>0">
            <xsl:for-each select="//instance[@class='Class']">
                <xsl:element name="rdf:Description">
                    <xsl:attribute name="rdf:about">
                        <xsl:value-of select="$BaseURI"></xsl:value-of>
                        <xsl:value-of select="./@name"></xsl:value-of>
                    </xsl:attribute>
                    <xsl:if test="count(record[@name='Annotations']/row) > 0">
                        <xsl:call-template name="GetClassAnnotationLabel"></xsl:call-template>
                        <xsl:call-template name="GetClassAnnotationComments"></xsl:call-template>
                        <xsl:call-template name="GetClassAnnotationSeeAlso"></xsl:call-template>
                    </xsl:if>
                    <xsl:element name="rdf:type">
                        <xsl:attribute name="rdf:resource">
                            <xsl:value-of select="$RdfSchemaURI"></xsl:value-of>
                            <xsl:text>Class</xsl:text>
                        </xsl:attribute>
                    </xsl:element>
                    <xsl:call-template name="GetSuperClasses">
                        <xsl:with-param name="name">
                            <xsl:value-of select="./@name"></xsl:value-of>
                        </xsl:with-param>
                    </xsl:call-template>
                </xsl:element>
            </xsl:for-each>
        </xsl:if>
    </xsl:template>

    <xsl:template name="BuildObjectProperties">
        <xsl:if test="count(//instance[@class='Object property'])>0">
```

eXtensible Markup Language file   length : 8827   lines : 231

Copy XSL File into Database



Suchen in: XSLT

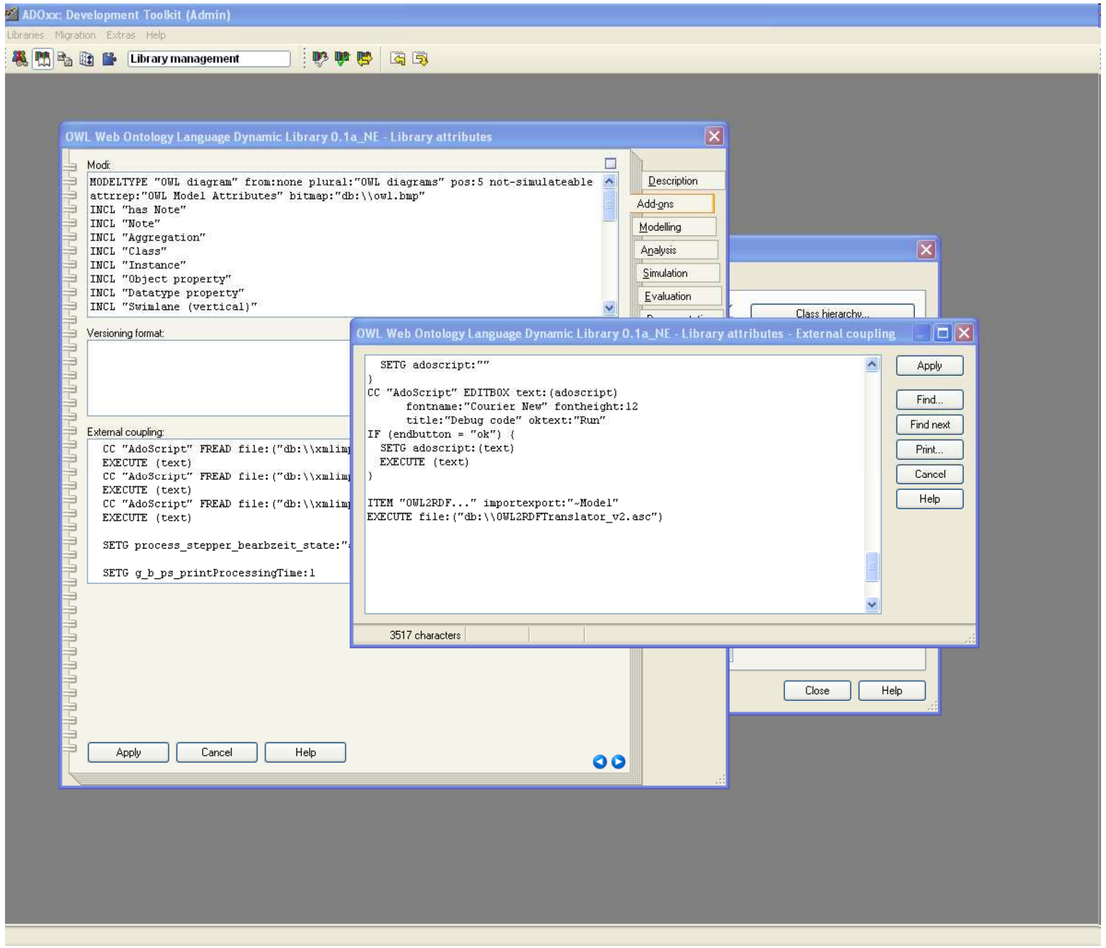
Dateiname: OWL2RDF.xsl

Dateityp: All Files

Add Menubar

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

Add Menubar


ADOxx: Development Toolkit (Admin)
Libraries Migration Extras Help
Library management
OWL Web Ontology Language Dynamic Library 0.1a_NE - Library attributes
Description
Add-ons
Modelling
Analysis
Simulation
Evaluation
Class hierarchy...
Apply
Cancel
Help
Close

Modi:

```
MODELTYPE "OWL diagram" from:none plural:"OWL diagrams" pos:5 not-simulateable
attrrep:"OWL Model Attributes" bitmap:"db:\\owl.bmp"
INCL "has Note"
INCL "Note"
INCL "Aggregation"
INCL "Class"
INCL "Instance"
INCL "Object property"
INCL "Datatype property"
INCL "Swimlane (vertical)"
```

Versioning format:

External coupling:

```
  CC "AdoScript" FREAD file:("db:\\xmlim
  EXECUTE (text)
  CC "AdoScript" FREAD file:("db:\\xmlim
  EXECUTE (text)
  CC "AdoScript" FREAD file:("db:\\xmlim
  EXECUTE (text)

  SETG process_stepper_bearbzeit_state:"

  SETG g_b_ps_printProcessingTime:1
```

OWL Web Ontology Language Dynamic Library 0.1a_NE - Library attributes - External coupling

```
  SETG adoscript:""
}
CC "AdoScript" EDITBOX text:(adoscript)
      fontname:"Courier New" fontheight:12
      title:"Debug code" oktext:"Run"
IF (endbutton = "ok") {
  SETG adoscript:(text)
  EXECUTE (text)
}

ITEM "OWL2RDF..." importexport:"~Model"
EXECUTE file:("db:\\OWL2RDFTranslator_v2.asc")
```

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