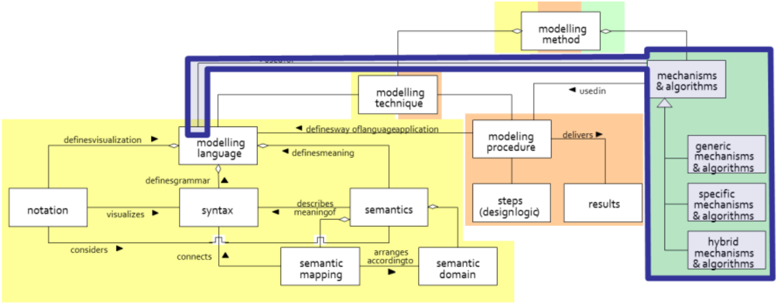


RDF Tunnel Direct Invocation

SCENARIO:
TRANSFORMATION OF CONCEPTS

Scenario Description

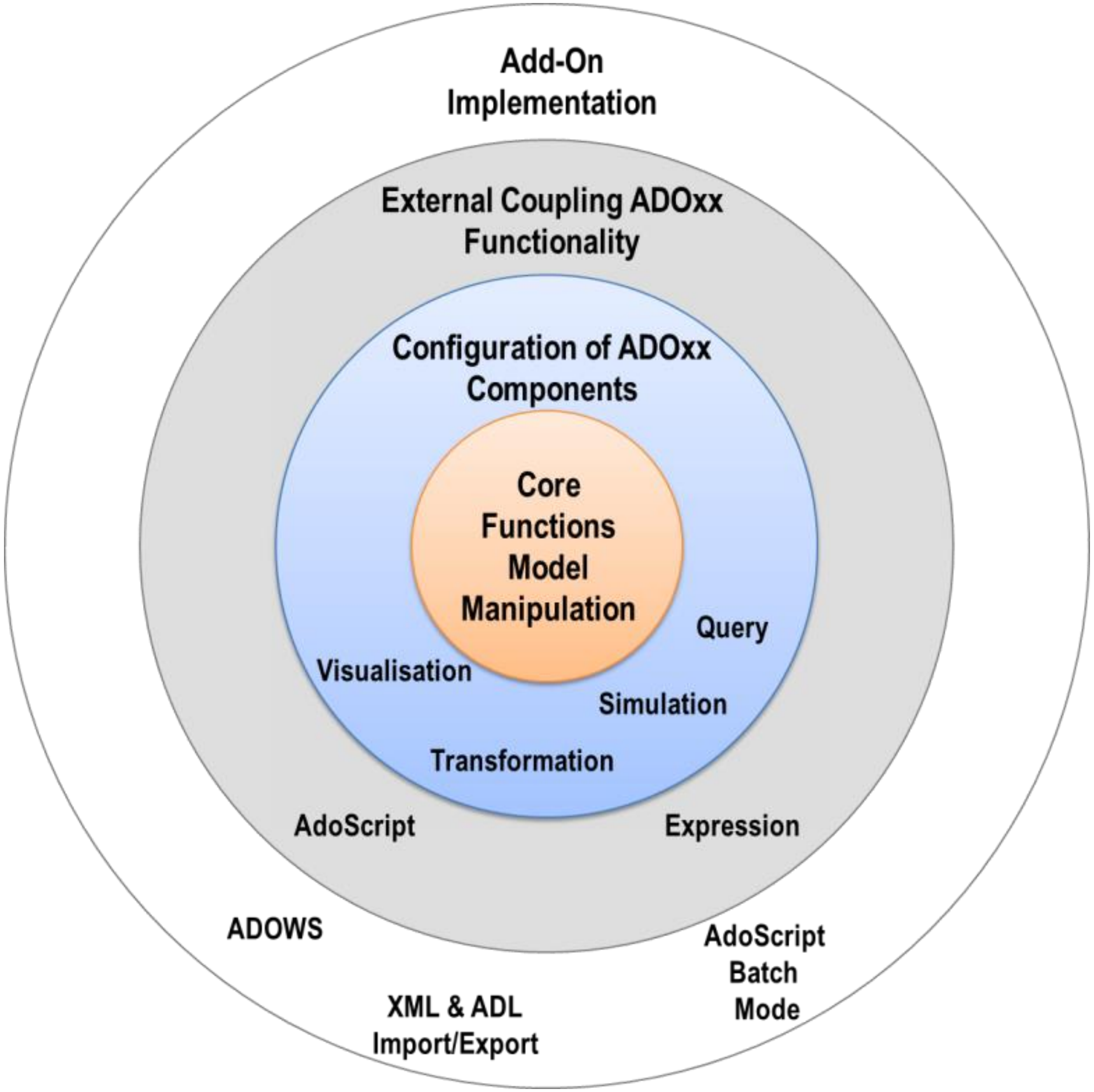
Case:

Transformation of objects from class “Actor”, who is in a certain space, has a certain role and participates to task(s) in a certain process. That actors going to be transformed into objects from class “Agent”. Attribute values of Agent objects are set with information collected from models in source library.

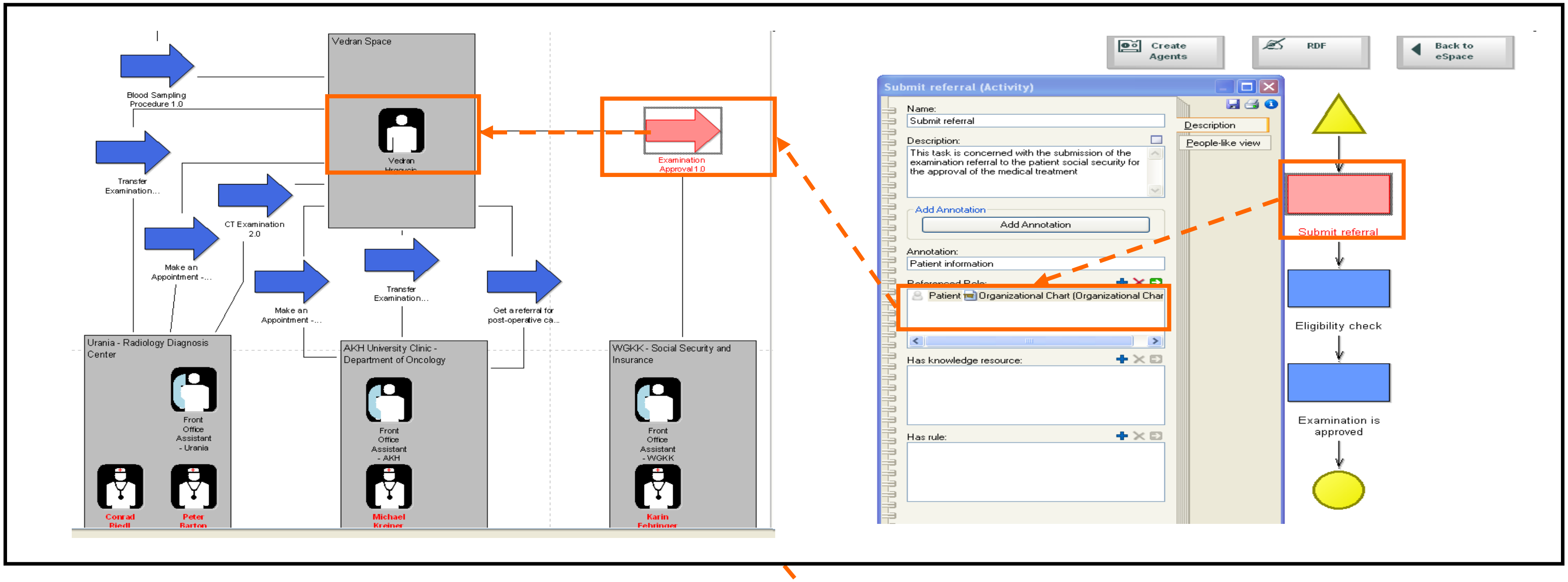
GOAL:

Demonstrate how to transform concepts in one Modelling Language into another modelling language with using ADOWS

ADOxx Functionality on Meta Level



Description of Algorithm



Submit referral (Activity)

Name:
Submit referral

Description:
This task is concerned with the submission of the examination referral to the patient social security for the approval of the medical treatment

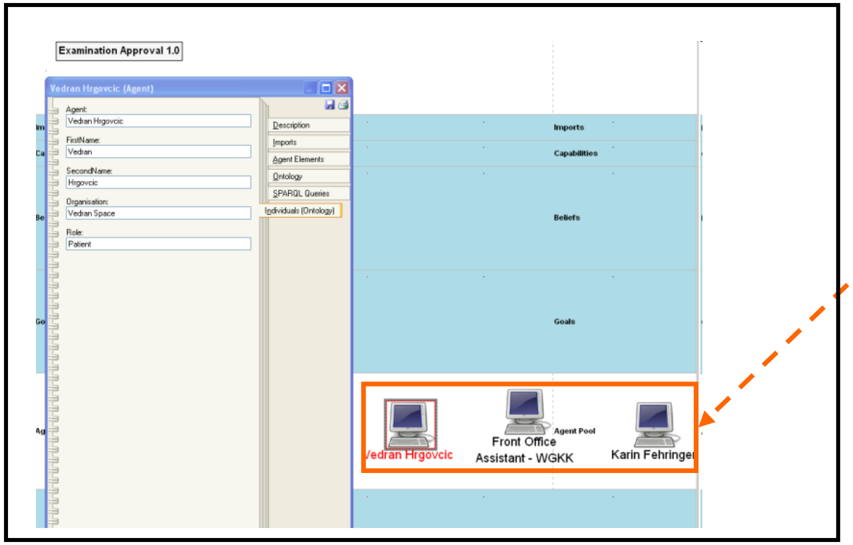
Add Annotation

Annotation:
Patient information

Referenced Role:
Patient Organizational Chart (Organizational Char

Has knowledge resource:

Has rule:



Vedran Hrgovcic (Agent)

Agent:
Vedran Hrgovcic

FirstName:
Vedran

SecondName:
Hrgovcic

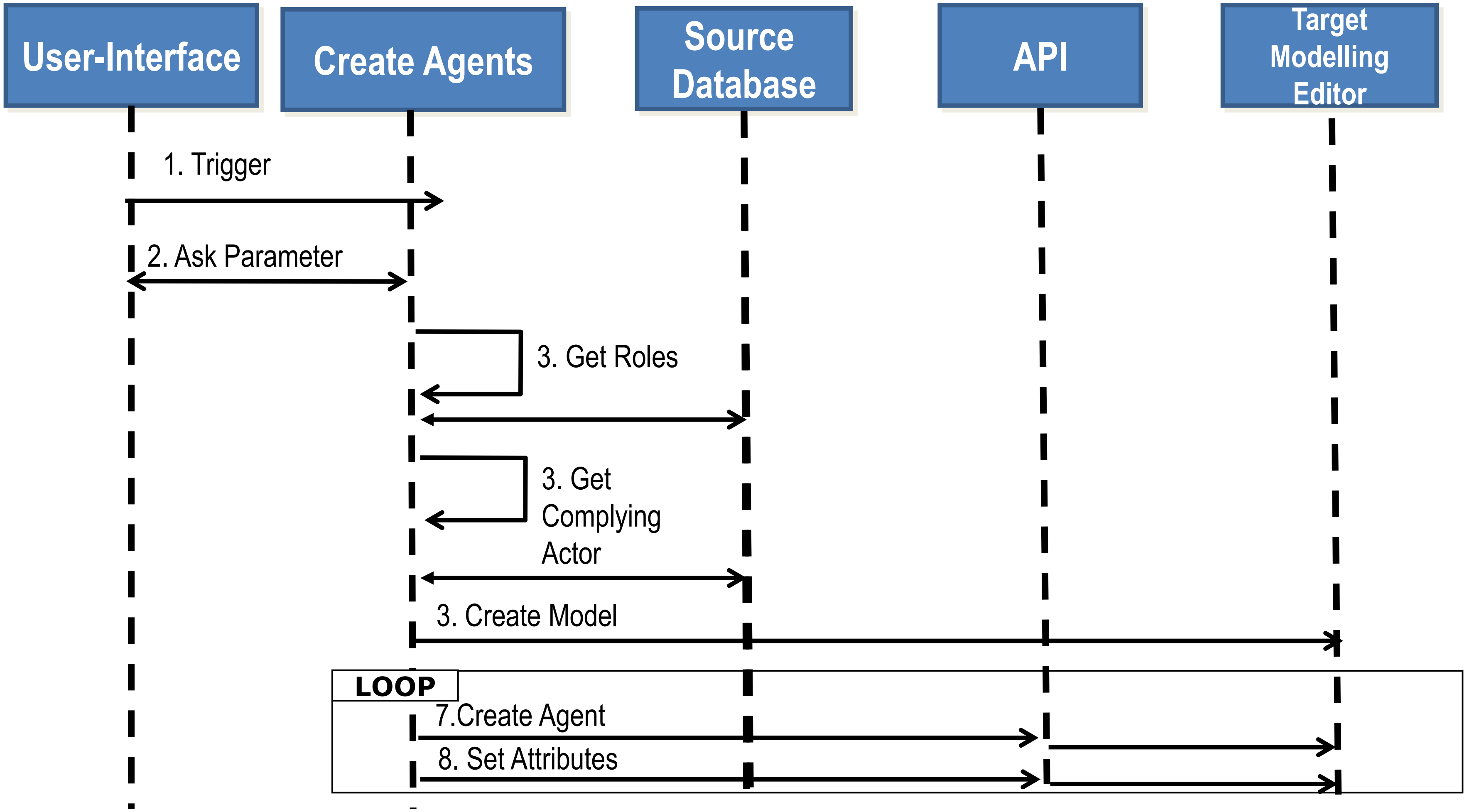
Organisation:
Vedran Space

Role:
Patient

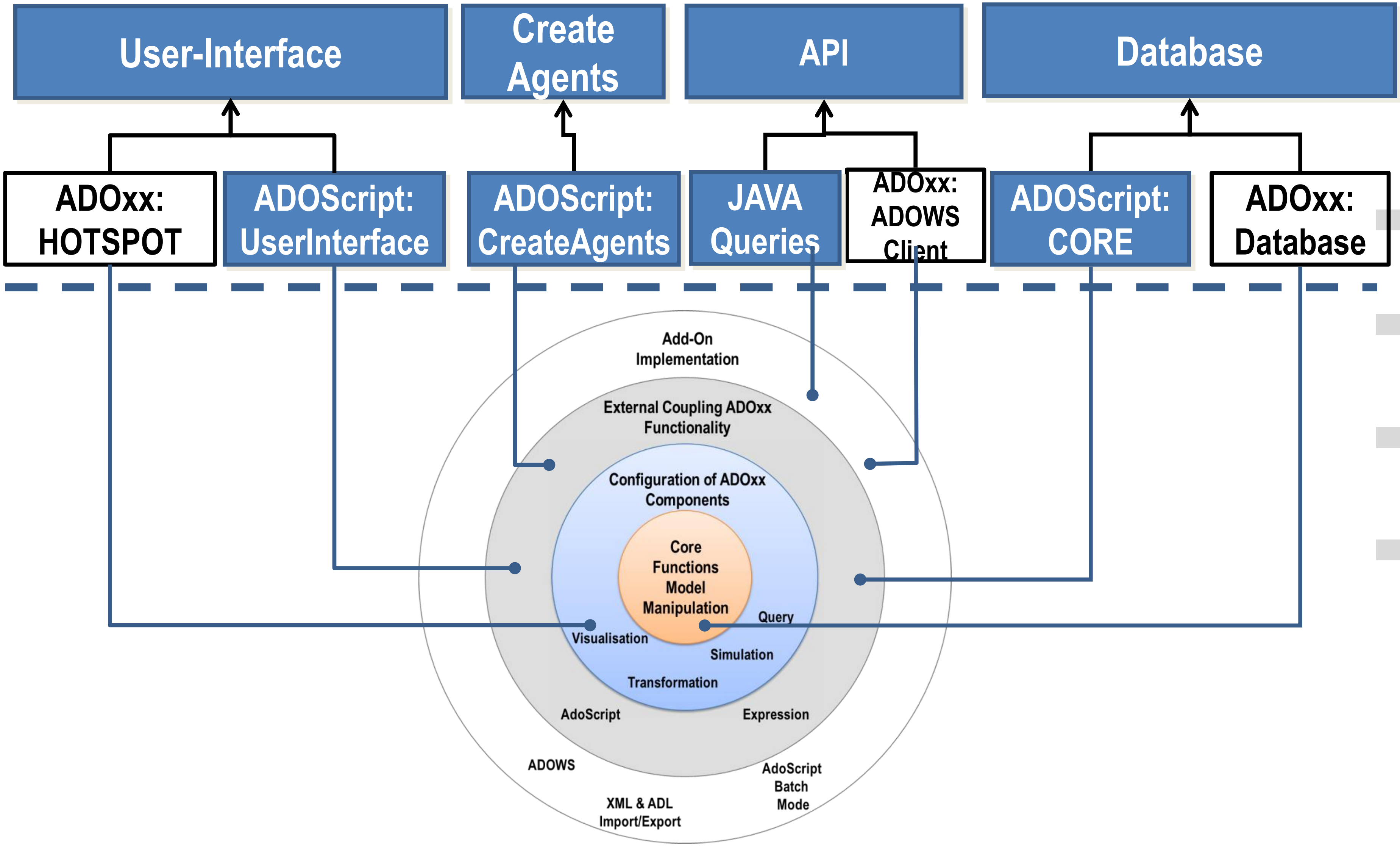
API (java)

ADOWS

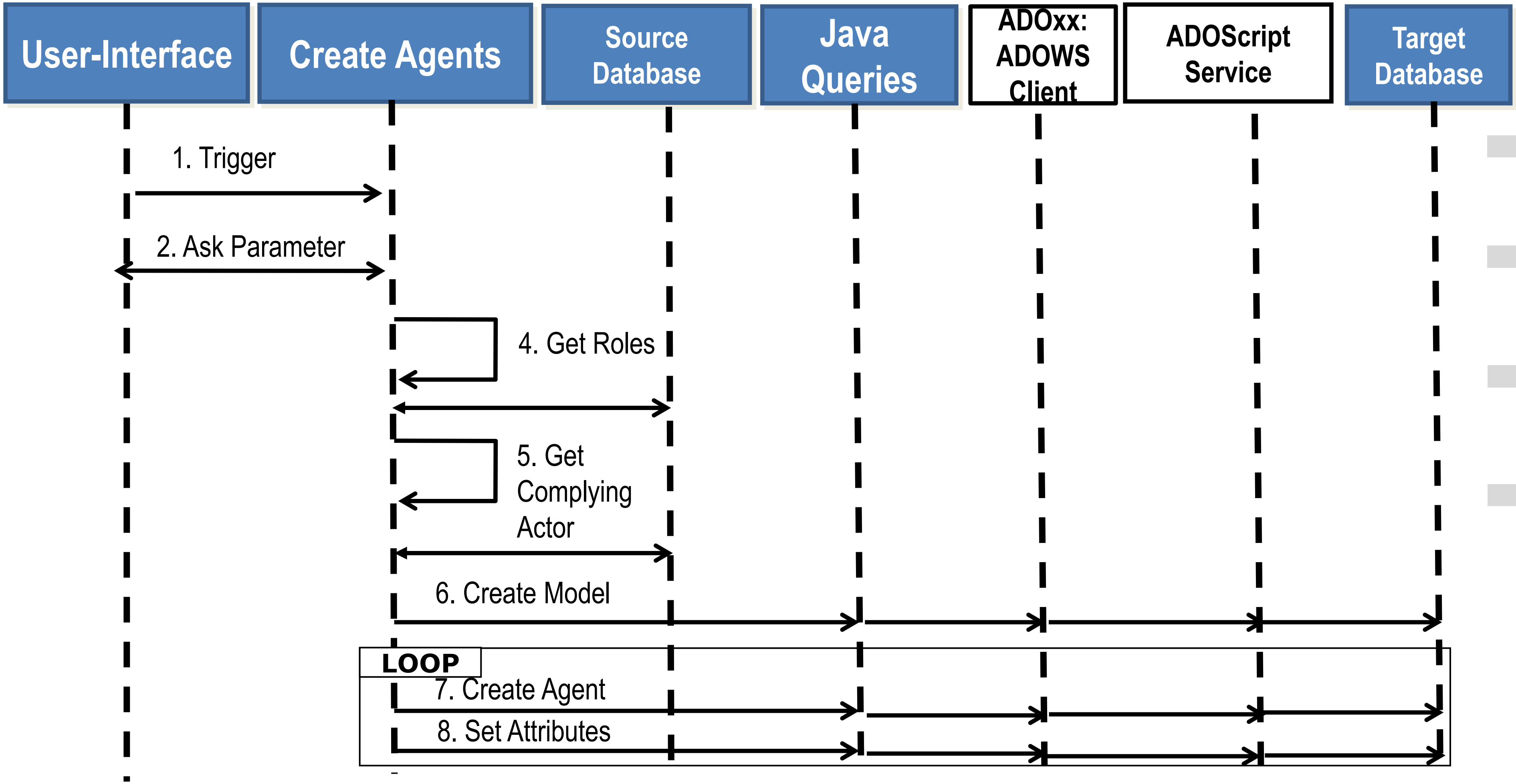
Description of Algorithm



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

•**ADOScript Service:** ADOScript Service: ADOScript Service listens a certain port to get and interpret requests

•**ADOWS Client:** is provide by ADOxx.org in order to achieve ADOWS and JAVA Integration

ADOxx Realisation Hands-On

1. **Source Modelling Language**
    1. Model Types “Space Model”, “Process Model”, “Organizational Chart”
    2. New class “Activity”, “Process Start”, ”Process End”, “Space”, “Role”, “Actor”, “Interaction Process”
2. **Configure ADOxx**
    1. Configure Model Graphrep with Hotspots
3. **Implement API with Java**
    1. Implement Queries
    2. Export Queries with ADOWS Client as runable JAR file
4. **Implement Algorithm with ADOscript**
    1. ADOscript User Interface
    2. Retrieve required information from models
    3. Invoking API
5. **Target Modelling Language**
    1. Model Types “Agent Model”
    2. New class “Agent”

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

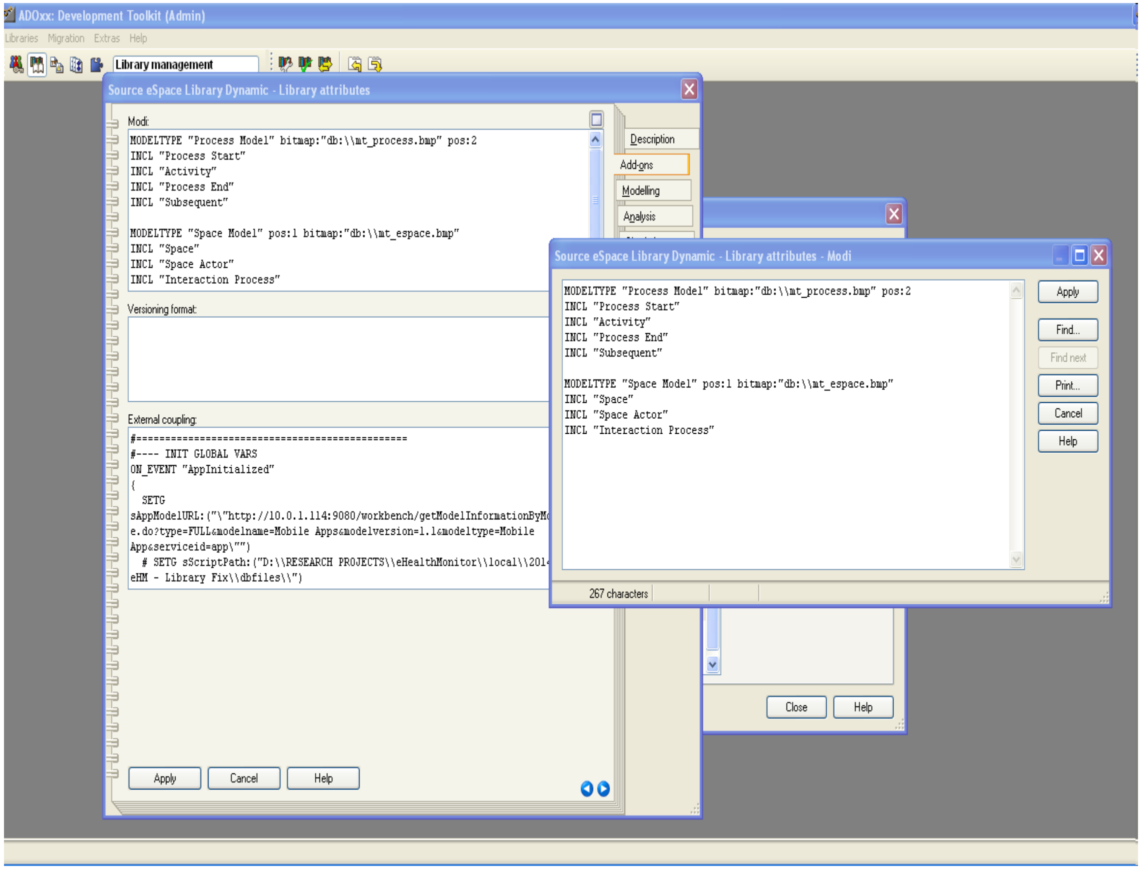
- Mechanisms & Algorithms Implementation
  - **Core Functions for Model Manipulation**
    - Database
    - **Visualisation** ✓
    - Query
    - Transformation
  - Configuration of ADOxx Components
    - Visualisation
    - Query
  - **External Coupling ADOxx Functionality** ✓
    - **ADOscript Triggers**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

RDF Tunnel Direct Invocation

SCENARIO:
TRANSFORMATION OF CONCEPTS

Define new Modeltypes Space Model", "Process Model", "Organizational Chart Model"


ADOxx: Development Toolkit (Admin)
Libraries Migration Extras Help
Library management
Source eSpace Library Dynamic - Library attributes
Description
Add-ons
Modelling
Analysis
Apply
Cancel
Help
Close
Help
Source eSpace Library Dynamic - Library attributes - Modi
Apply
Find...
Find next
Print...
Cancel
Help
267 characters

Modi:

```
MODELTYPE "Process Model" bitmap:"db:\\mt_process.bmp" pos:2
INCL "Process Start"
INCL "Activity"
INCL "Process End"
INCL "Subsequent"

MODELTYPE "Space Model" pos:1 bitmap:"db:\\mt_espace.bmp"
INCL "Space"
INCL "Space Actor"
INCL "Interaction Process"
```

Versioning format:

External coupling:

```
#==============================================
#---- INIT GLOBAL VARS
ON_EVENT "AppInitialized"
{
  SETG
sAppModelURL:("\"http://10.0.1.114:9080/workbench/getModelInformationByMo
e.do?type=FULL&modelname=Mobile Apps&modelversion=1.1&modeltype=Mobile
App&serviceid=app\"")
  # SETG sScriptPath:("D:\\RESEARCH PROJECTS\\eHealthMonitor\\local\\2014
eHM - Library Fix\\dbfiles\\")
```

```
MODELTYPE "Process Model" bitmap:"db:\\mt_process.bmp" pos:2
INCL "Process Start"
INCL "Activity"
INCL "Process End"
INCL "Subsequent"

MODELTYPE "Space Model" pos:1 bitmap:"db:\\mt_espace.bmp"
INCL "Space"
INCL "Space Actor"
INCL "Interaction Process"
```

New Modeltypes:

•Select "Source eSpace Library Dynamic" and open Library attributes.

•Got to Add Ons

•Add the Modeltypes "Space Model" and "Process Model" in the Modi attribute

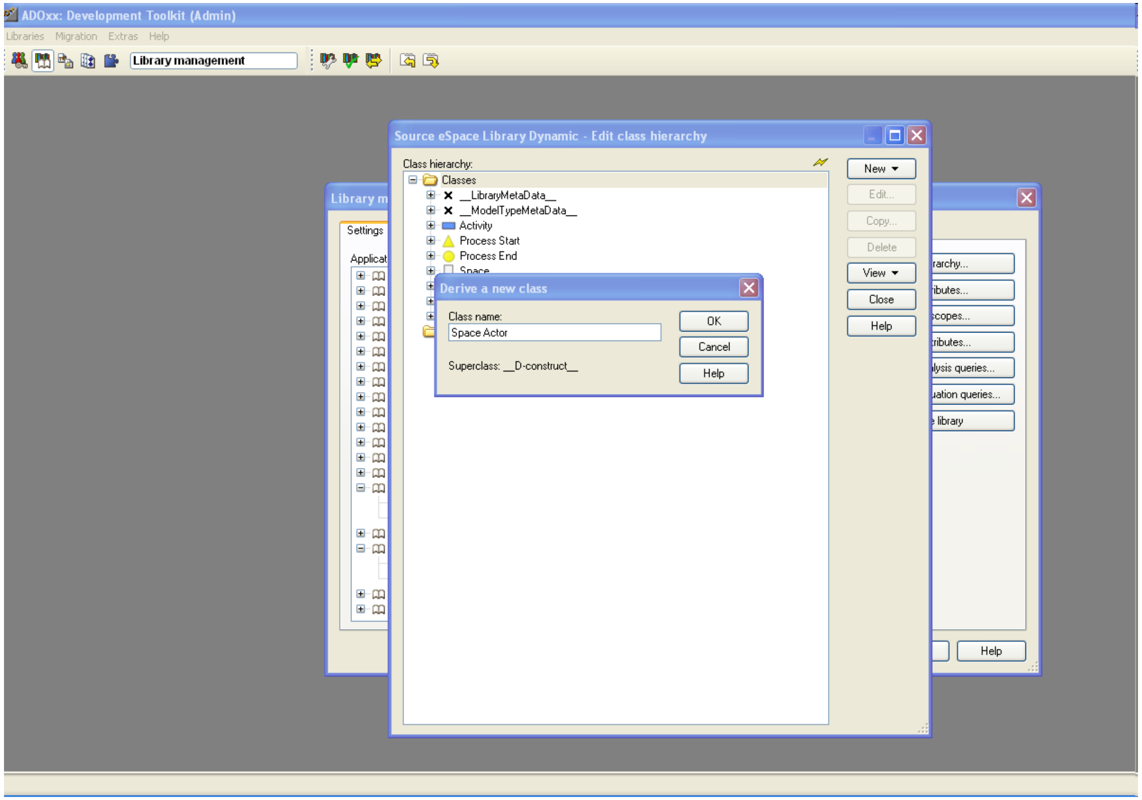
•Select "Source eSpace Library Static" and open Library attributes.

•Got to Add Ons

•Add the Modeltype "Organizational Chart Model" in the Modi attribute

•When the classes are defined, you need to INCLUDE them

Create New Classes



Class hierarchy:
- Classes
  - __LibraryMetaData__
  - __ModelTypeMetaData__
  - Activity
  - Process Start
  - Process End
  - Space

Class name: Space Actor

Superclass: __D-construct__

Create New Classes

•Select “Source eSpace 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: “Activity”, “Process Start”, ”Process End”, “Space Actor”, “Interaction Process”

•Activity”, “Process Start”, ”Process End”, “Space”, “Space Actor”, “Interaction Process” are now sub-classes of __D-construct__

•Select __D_aggregation__

•Create class ““Space”

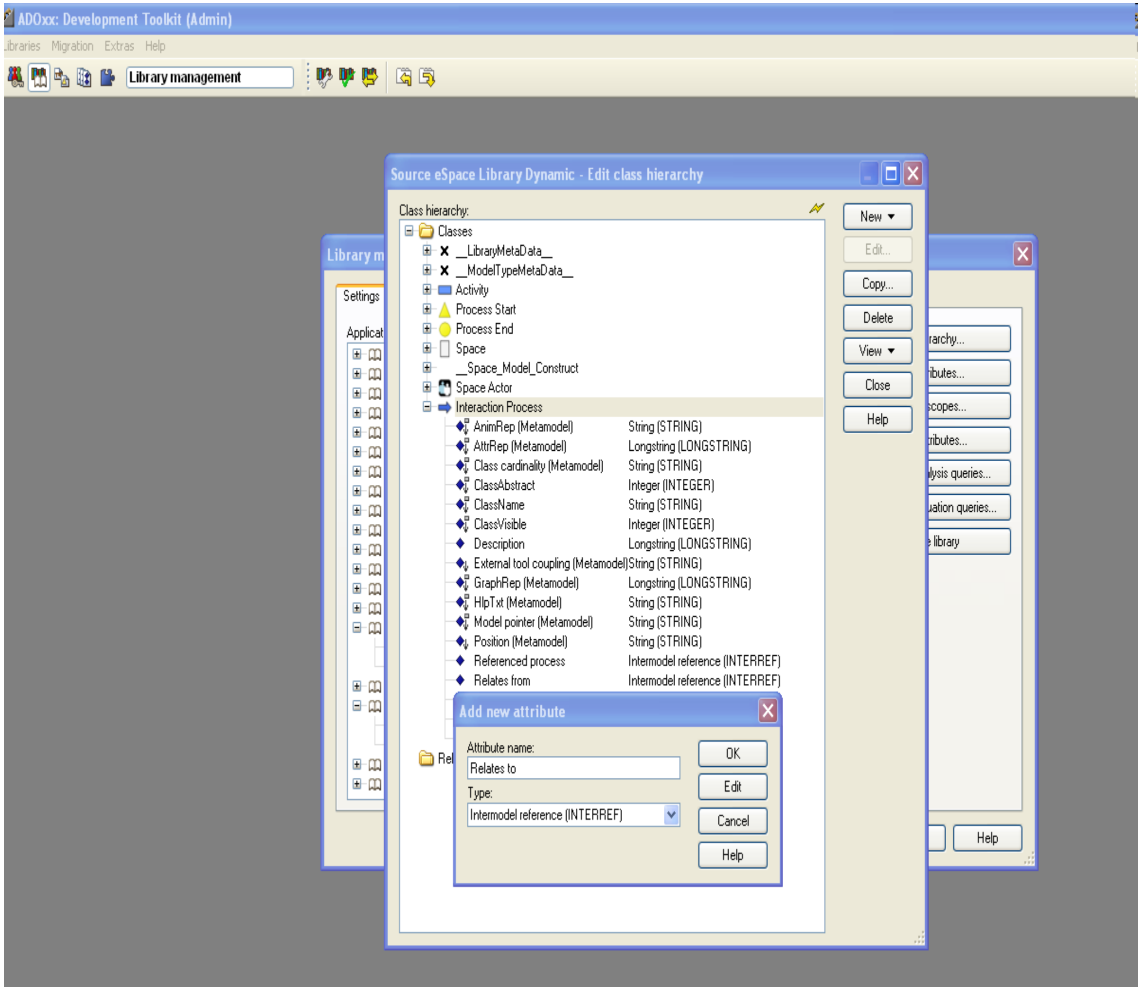
•Select “Source eSpace Static Library” and open Library attributes.

•Open Class hierarchy, view “Metamodel” and “Class hierarchy” in the View button, select __S-construct__ and click new class.

•Name new classes: “Role”

•“Roel” are now sub-classes of __D-construct__

Add Attributes for Classes



Class hierarchy:

- Classes
  - __LibraryMetaData__
  - __ModelTypeMetaData__
  - Activity
  - Process Start
  - Process End
  - Space
  - __Space_Model_Construct
  - Space Actor
  - Interaction Process

AnimRep (Metamodel)	String (STRING)
AttrRep (Metamodel)	Longstring (LONGSTRING)
Class cardinality (Metamodel)	String (STRING)
ClassAbstract	Integer (INTEGER)
ClassName	String (STRING)
ClassVisible	Integer (INTEGER)
Description	Longstring (LONGSTRING)
External tool coupling (Metamodel)	String (STRING)
GraphRep (Metamodel)	Longstring (LONGSTRING)
HlpTxt (Metamodel)	String (STRING)
Model pointer (Metamodel)	String (STRING)
Position (Metamodel)	String (STRING)
Referenced process	Intermodel reference (INTERREF)
Relates from	Intermodel reference (INTERREF)

Attribute name: Relates to

Type: Intermodel reference (INTERREF)

Add Attributes

•Select “Space Actor” and click Newattribute.

•Make “Referenced Role” as type INTERREF.

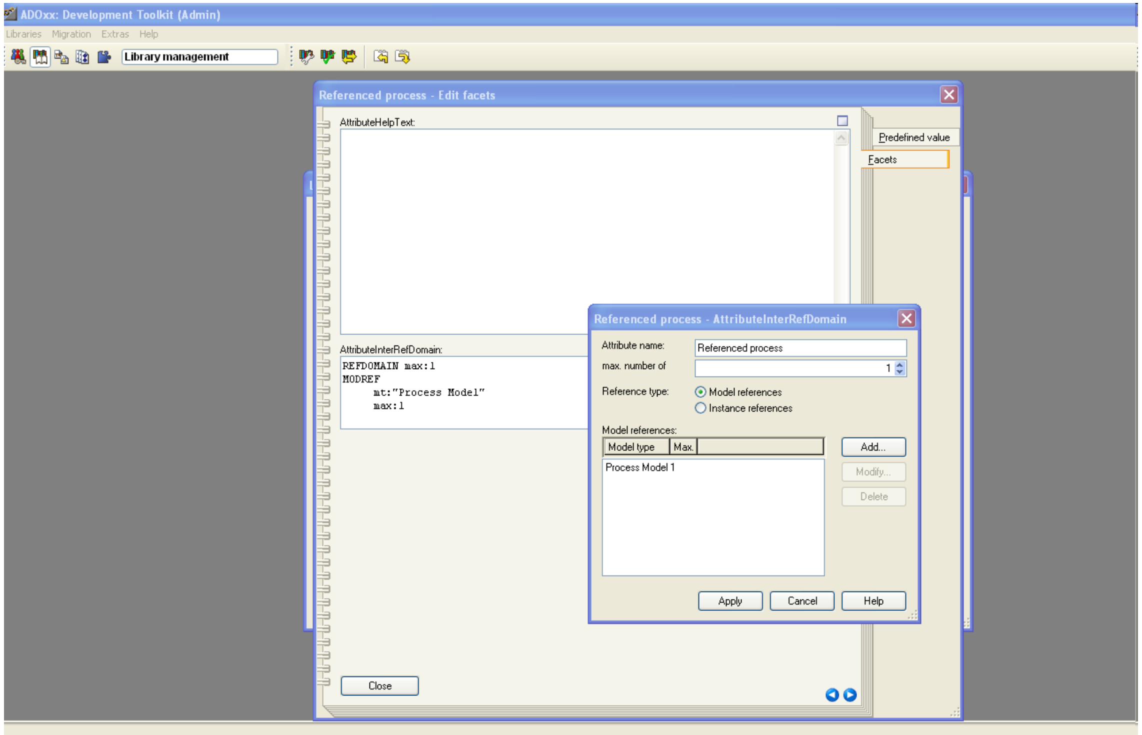
•Select “Interaction Process” and click New, attribute.

•Make “Referenced Process” “Relates from” and “Relates to” as type INTERREF.

•Select “Activity” and click New, attribute.

•Make “Referenced Role” as type INTERREF

Edit INTERREF


ADOxx: Development Toolkit (Admin)
Libraries Migration Extras Help
Library management
Referenced process - Edit facets
Predefined value
Facets
Close

AttributeHelpText:

AttributeInterRefDomain:

```
REFDOMAIN max:1
MODREF
     mt:"Process Model"
     max:1
```

Referenced process - AttributeInterRefDomain

Attribute name: Referenced process

max. number of: 1

Reference type: ● Model references ○ Instance references

Model references:

Model type	Max.
Process Model 1	

Add...
Modify...
Delete
Apply
Cancel
Help

Specification of INTERREF" Referenced Process"

- •EDIT Facet
- •Select AttributeInterrefDomain
- •Select "Model reference"
- •Max number of references is 1
- •Select Process Model
- •Max number of references is 1

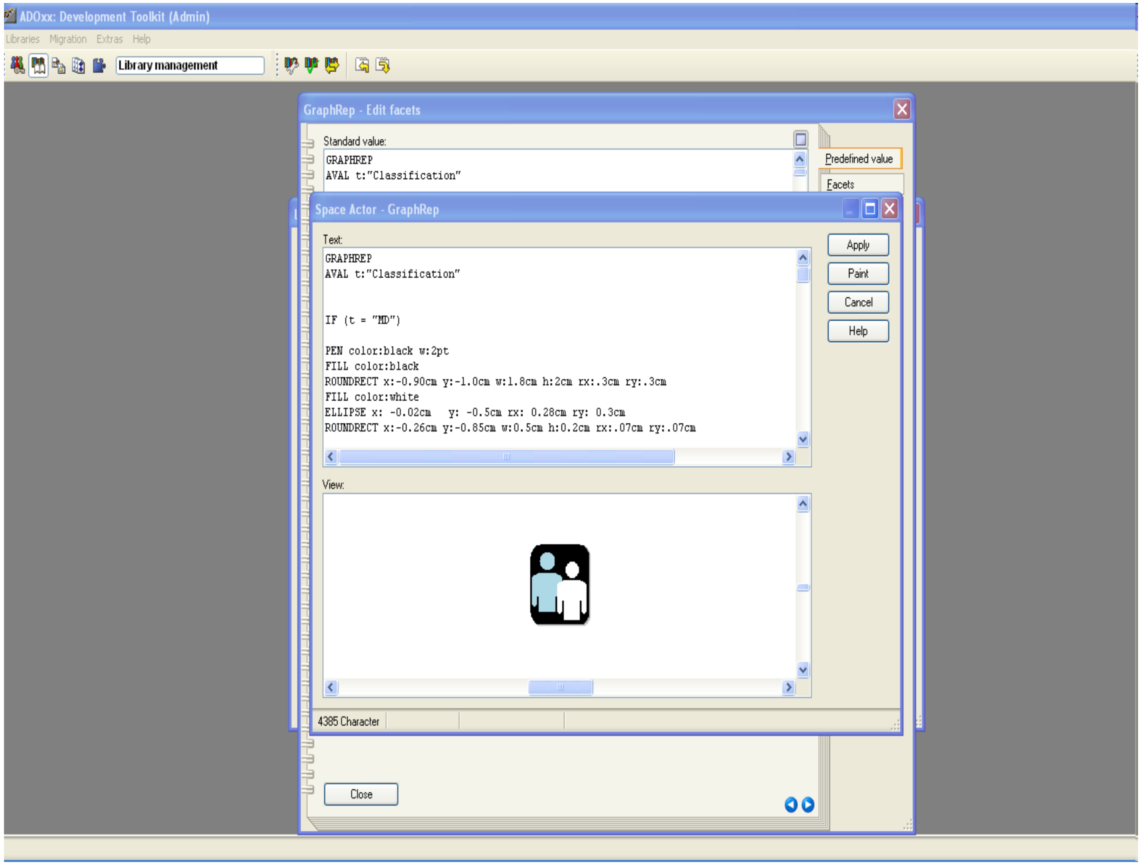
Specification of INTERREF" Relates from" and "Relates to"

- •EDIT Facet
- •Select AttributeInterrefDomain
- •Select "Instance References" and Space Model
- •Max number of references is 1
- •Select Space
- •Max number of references is 1

Specification of INTERREF" Referenced Role"

- •EDIT Facet
- •Select AttributeInterrefDomain
- •Select "Instance References" and Organizational Chart
- •Max number of references is 1
- •Select Role
- •Max number of references is 1

Add GRAPHREP



Standard value:

```
GRAPHREP
AVAL t:"Classification"
```

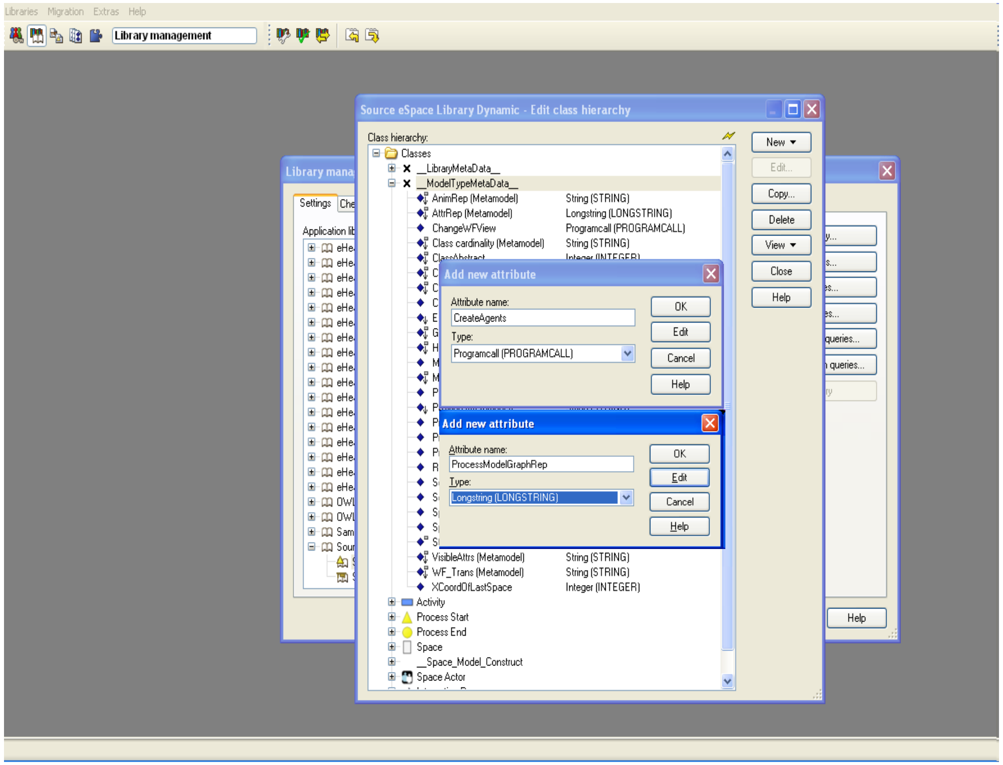
Text:

```
GRAPHREP
AVAL t:"Classification"



IF (t = "MD")

PEN color:black w:2pt
FILL color:black
ROUNDRECT x:-0.90cm y:-1.0cm w:1.8cm h:2cm rx:.3cm ry:.3cm
FILL color:white
ELLIPSE x: -0.02cm   y: -0.5cm rx: 0.28cm ry: 0.3cm
ROUNDRECT x:-0.26cm y:-0.85cm w:0.5cm h:0.2cm rx:.07cm ry:.07cm
```

Specification of GRAPHREP

- Select “Space Actor”
- Click on Attribute “GraphRep”
- Open the GraphRep Editor
- Enter text, paint it and apply.
- Repeat the steps for class “Interaction Process”, “Activity” “Process Start”, “Process End” and “Role”

Add Hotspot



Source eSpace Library Dynamic - Edit class hierarchy

Class hierarchy:

- Classes
  - __LibraryMetaData__
  - __ModelTypeMetaData__
    - AnimRep (Metamodel) String (STRING)
    - AttrRep (Metamodel) Longstring (LONGSTRING)
    - ChangeWFView Programcall (PROGRAMCALL)
    - Class cardinality (Metamodel) String (STRING)
    - ClassAbstract Integer (INTEGER)
    - VisibleAttrs (Metamodel) String (STRING)
    - WF_Trans (Metamodel) String (STRING)
    - XCoordOfLastSpace Integer (INTEGER)
  - Activity
  - Process Start
  - Process End
  - Space
  - __Space_Model_Construct
  - Space Actor

Add new attribute

Attribute name: CreateAgents

Type: Programcall (PROGRAMCALL)

Add new attribute

Attribute name: ProcessModelGraphRep

Type: Longstring (LONGSTRING)

Add Hotspot

- Select "__ModelTypeMetaData__"
- Add New Attributes "CreateAgents" as type PROGRAMCALL and "ProcessModelGraphRep" as type LONGSTRING
- Copy GraphRep code into standard value of "ProcessModelGraphRep"
- Select "CreateAgents" and enter "CreateAgents" into Standard value, open facets the and copy programcall code into EnumarationDomain

Programcall

```
ITEM "CreateAgents"
EXECUTE file:("db:\\DirectInvocationScript.asc")
```

GraphRep Code

```
GRAPHREP layer:-1
GRADIENT_RECT x:6.6cm y:.6cm w:3.2cm h:1.2cm style:diagcross color2:lightgray color1:gray color3:lightgray color4:gray
GRADIENT_RECT x:6.8cm y:.7cm w:2.8cm h:1.0cm style:vert color2:lightgray color1:white
PEN w:0.05cm color:darkgray
FILL style:null
RECTANGLE x:6.6cm y:.6cm w:3.2cm h:1.2cm
FONT color:("darkslategray") bold h:9pt
TEXT "Create Agents" x:8.3cm y:.8cm line-break:words w:c:2.0cm
FONT "Wingdings 2" bold h:18pt color:("darkslategray")
TEXT "9" x:7.2cm y:1.1cm w:c h:c
FONT color:("darkslategray")
HOTSPOT "CreateAgents" x:6.6cm y:.6cm w:3.2cm h:1.2cm
```

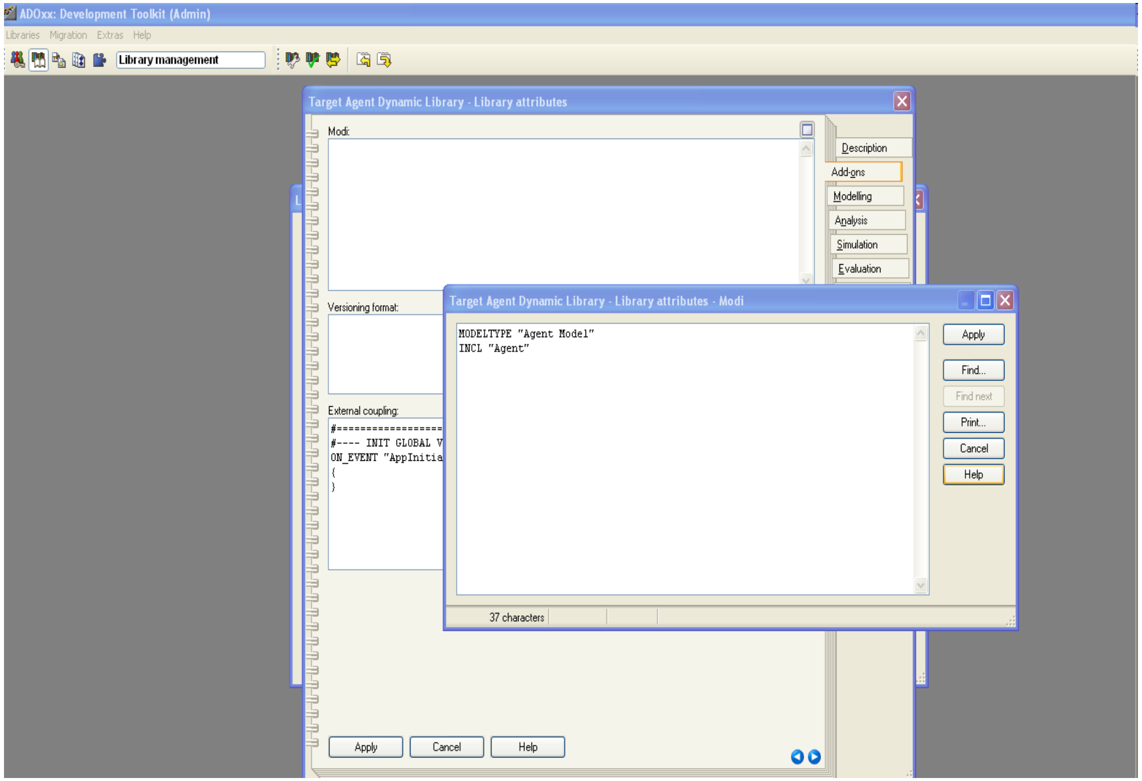
Implement and Import API

```
package org.adoxx.adows.client;

/**
 * Samples of AdoScripts summarized in a static class
 *
 *
 */
public class QueryScripts {

    private static final String RESULTVAR = "result";

/**
 * Sample AdoScript implementation to query a model of a specific type for all instances
 *
 * @param modelname
 * @param modeltype
 * @param resultName
 * @return objids (as String, space-seperated)
 */
public static String getAllInstancesOfModelByName(String modelname, String modeltype, String resultName)
{
StringBuffer buffer = new StringBuffer();
buffer.append("CC \"Core\" GET_MODEL_ID modelname:\""+modelname+"\" modeltype:\""+modeltype+"\"\r\n");
buffer.append("CC \"Core\" LOAD_MODEL modelid:(modelid) read-access\r\n");
buffer.append("CC \"Core\" GET_ALL_OBJS modelid:(modelid)\r\n");
buffer.append("CC \"Core\" DISCARD_MODEL modelid:(modelid)\r\n");
buffer.append("SETG "+resultName+":(objids)");
return buffer.toString();

}
.. .
```

Implement and Import ADOscript File into Database

```
SET sTempAPIResultsFile:("API_results.txt")
CC "Modeling" GET_ACT_MODEL
# --> modelid: intValue.
SET actModelId:(modelid)

CC "Core" GET_CLASS_ID classname:("Interaction Process")
SET n_intproc_classid:(classid)

CC "Core" GET_ATTR_ID classid:(n_intproc_classid) attrname:("Relates from")
SET n_space_relfrom_attrid:(attrid)

CC "Core" GET_ATTR_ID classid:(n_intproc_classid) attrname:("Relates to")
SET n_space_relto_attrid:(attrid)

CC "CoreUI"   MODEL_SELECT_BOX  modeltype:("Space Model") title:("Select Space Model") boxtext:("Please select a Space
Model") oktext:("Select")
SET s_selected_space_modelid:(modelids)
SET s_model_selection:(endbutton)

IF (s_model_selection ="ok")
{
                CC "Core" LOAD_MODEL modelid:(VAL s_selected_space_modelid)

                CC "AdoScript" PERCWIN_CREATE title:"Please wait! Agents are being created..."
                ####################in MAS####################################
                #CC "Core"  GET_CLASS_ID classname:("Agent") bp-library
                #--> RESULT ecode: intValue classid: intValue isrel: intValue
                #SET n_agent_classId:(classid)

                SYSTEM ("cmd /c java -jar tools\\DirectInvocationAPI.jar \"getClassIdByName\" \"Agent\"")
                CC "AdoScript"  FREAD file: (sTempAPIResultsFile) binary: 0 base64: 0
                SET n_agent_classId:(text)

}

...
```

Define new Modeltype “Agent Model”



Modi:

```
MODELTYPE "Agent Model"
INCL "Agent"
```

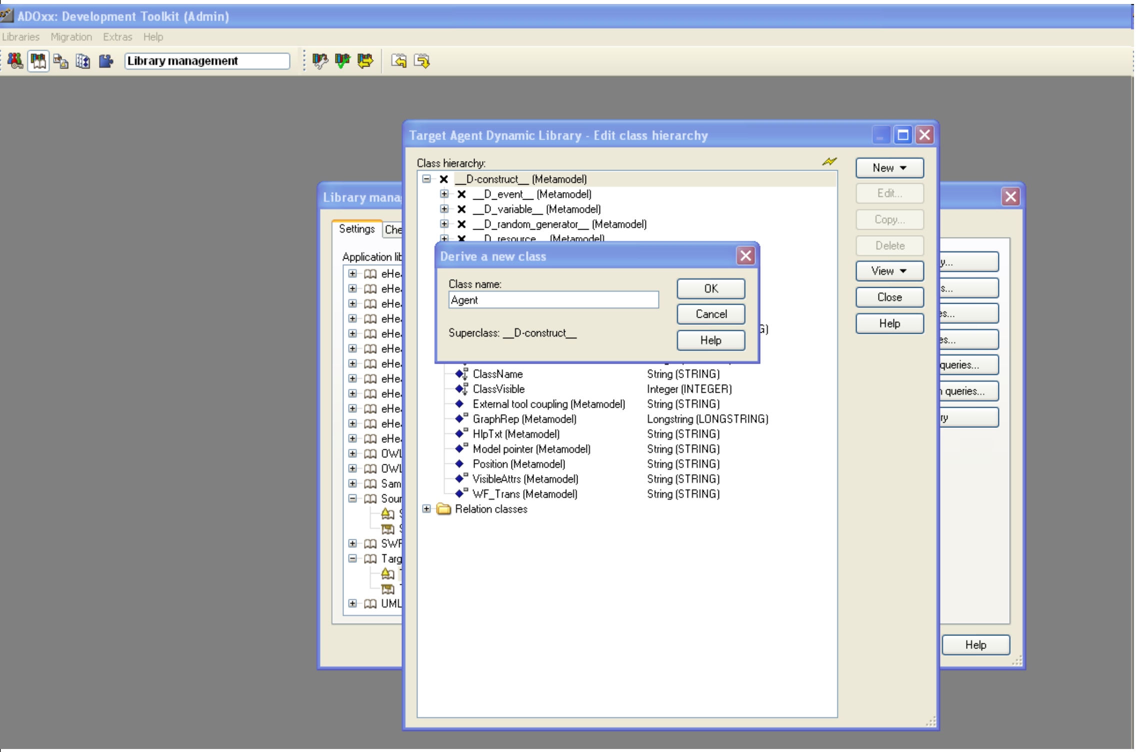
External coupling:

```
#==================
#---- INIT GLOBAL V
ON_EVENT "AppInitia
{
}
```

New Modeltypes:

- Select “Target Agent Library Dynamic” and open Library attributes.
- Go to Add Ons
- Add the Modeltype “Agent Model” in the Modi attribute
- When the classes are defined, you need to INCLUDE them

Create New Classes



Target Agent Dynamic Library - Edit class hierarchy

Class hierarchy:

- __D-construct__ (Metamodel)
  - __D_event__ (Metamodel)
  - __D_variable__ (Metamodel)
  - __D_random_generator__ (Metamodel)
  - __D_resource__ (Metamodel)

ClassName	String (STRING)
ClassVisible	Integer (INTEGER)
External tool coupling (Metamodel)	String (STRING)
GraphRep (Metamodel)	Longstring (LONGSTRING)
HlpTxt (Metamodel)	String (STRING)
Model pointer (Metamodel)	String (STRING)
Position (Metamodel)	String (STRING)
VisibleAttrs (Metamodel)	String (STRING)
WF_Trans (Metamodel)	String (STRING)

Relation classes

Derive a new class

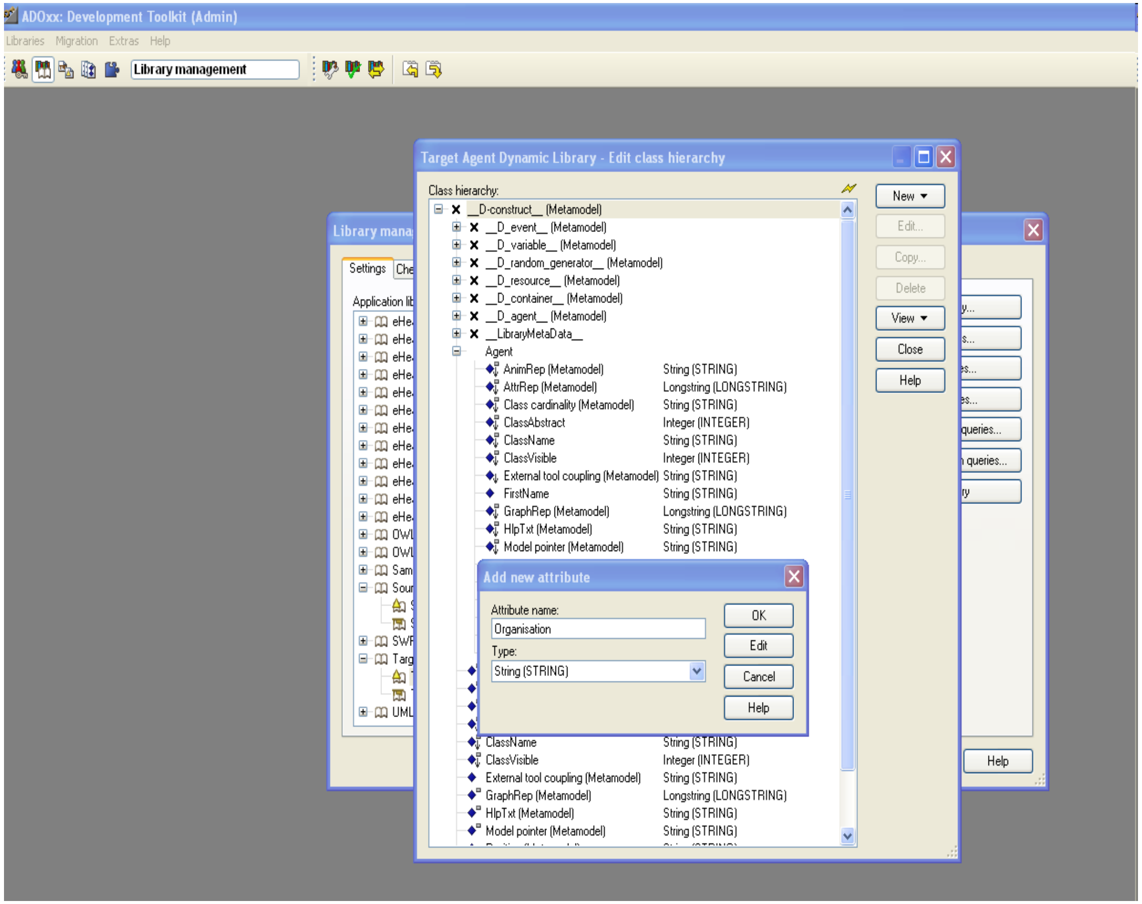
Class name: Agent

Superclass: __D-construct__

Create New Classes

- Select “Target Agent 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 class: ”Agent”
- Agent is now sub-clas of __D-construct__

Add Attributes for Classes



Class hierarchy:

```
__D-construct__ (Metamodel)
  __D_event__ (Metamodel)
  __D_variable__ (Metamodel)
  __D_random_generator__ (Metamodel)
  __D_resource__ (Metamodel)
  __D_container__ (Metamodel)
  __D_agent__ (Metamodel)
  __LibraryMetaData__
  Agent
    AnimRep (Metamodel)                 String (STRING)
    AttrRep (Metamodel)                 Longstring (LONGSTRING)
    Class cardinality (Metamodel)       String (STRING)
    ClassAbstract                       Integer (INTEGER)
    ClassName                           String (STRING)
    ClassVisible                        Integer (INTEGER)
    External tool coupling (Metamodel)  String (STRING)
    FirstName                           String (STRING)
    GraphRep (Metamodel)                Longstring (LONGSTRING)
    HlpTxt (Metamodel)                  String (STRING)
    Model pointer (Metamodel)           String (STRING)
  ClassName                             String (STRING)
  ClassVisible                          Integer (INTEGER)
  External tool coupling (Metamodel)    String (STRING)
  GraphRep (Metamodel)                  Longstring (LONGSTRING)
  HlpTxt (Metamodel)                    String (STRING)
  Model pointer (Metamodel)             String (STRING)
```

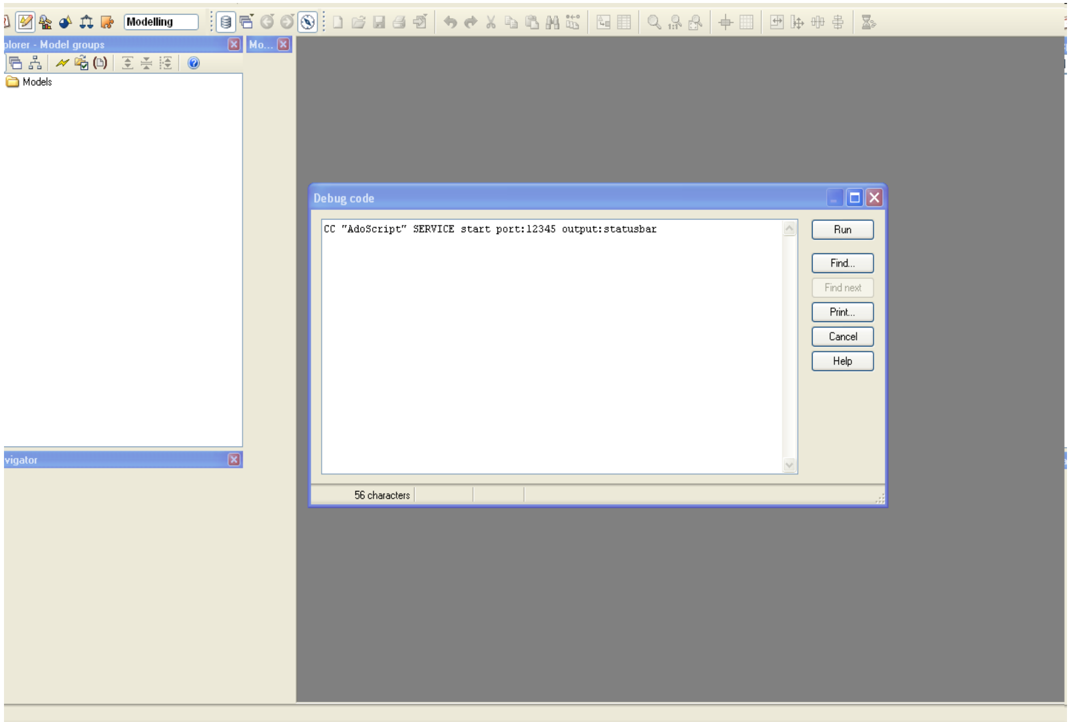
Attribute name: Organisation

Type: String (STRING)

Add Attributes

•Select “Agent” and click Newattribute.
•Make “FirstName”, ”SecondName”, ”Role”, “Organisation” as type String.

Start Adoxx Web Service in Target Library



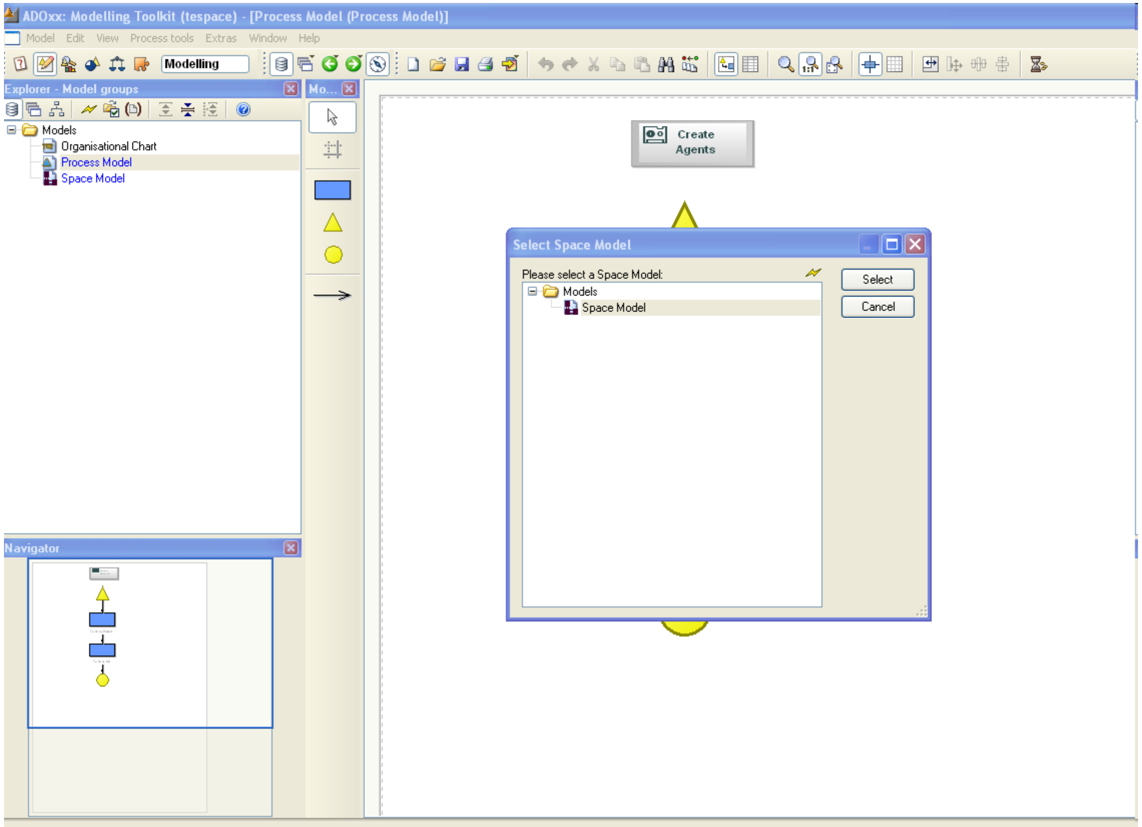
```
CC "AdoScript" SERVICE start port:12345 output:statusbar
```

Start ADOWS

- With using ADOScript Debug Shell enter AdoScript code:

CC "AdoScript" SERVICE start port:12345 output:statusbar

Results



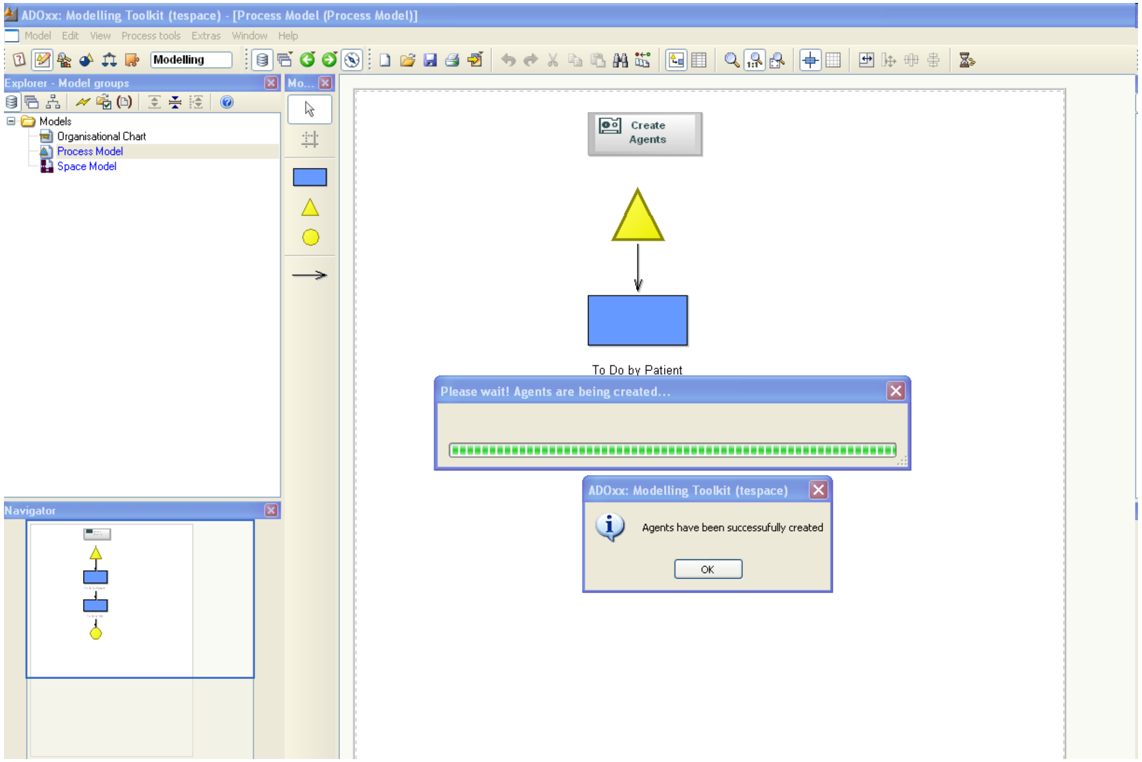
Please select a Space Model:

Models
Space Model

Start Creating Agents

- •Click “Create Agents” Button
- •Select corresponding Space Model

Results


ADOxx: Modelling Toolkit (tespace) - [Process Model (Process Model)]

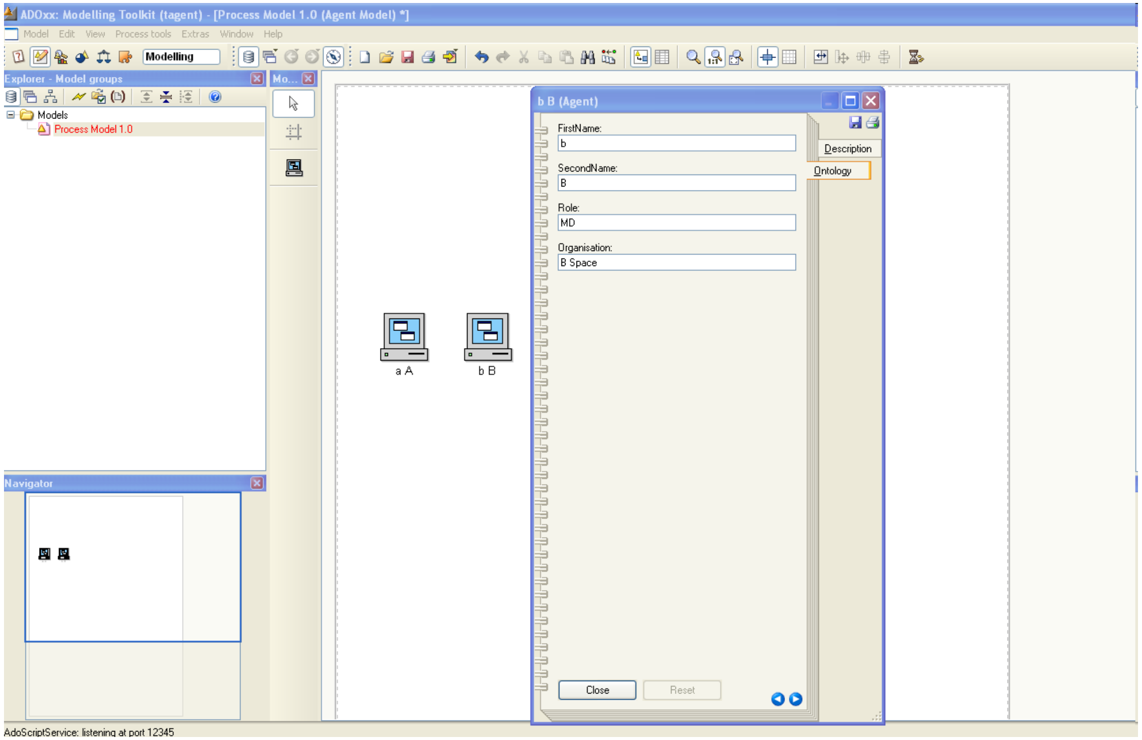
Please wait! Agents are being created...

ADOxx: Modelling Toolkit (tespace)

Agents have been successufully created

- AdoScript retrieves required information from three models and invokes API

Results


ADOxx: Modelling Toolkit (tagent) - [Process Model 1.0 (Agent Model) *]
Model Edit View Process tools Extras Window Help
Modelling
Explorer - Model groups
Models
Process Model 1.0
Navigator
a A
b B
b B (Agent)
Description
Ontology
Close
Reset

FirstName: b

SecondName: B

Role: MD

Organisation: B Space

AdoScriptService: listening at port 12345

- Agents are created with corresponding information

Further Questions?

www.adoxx.org

tutorial@adoxx.org