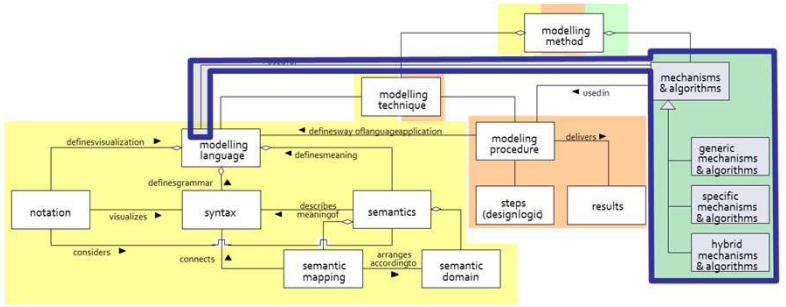


Change Log - SMS

SCENARIO:
SMS NOTIFICATION ABOUT CHANGES MADE BY ANOTHER USERS

Scenario Description

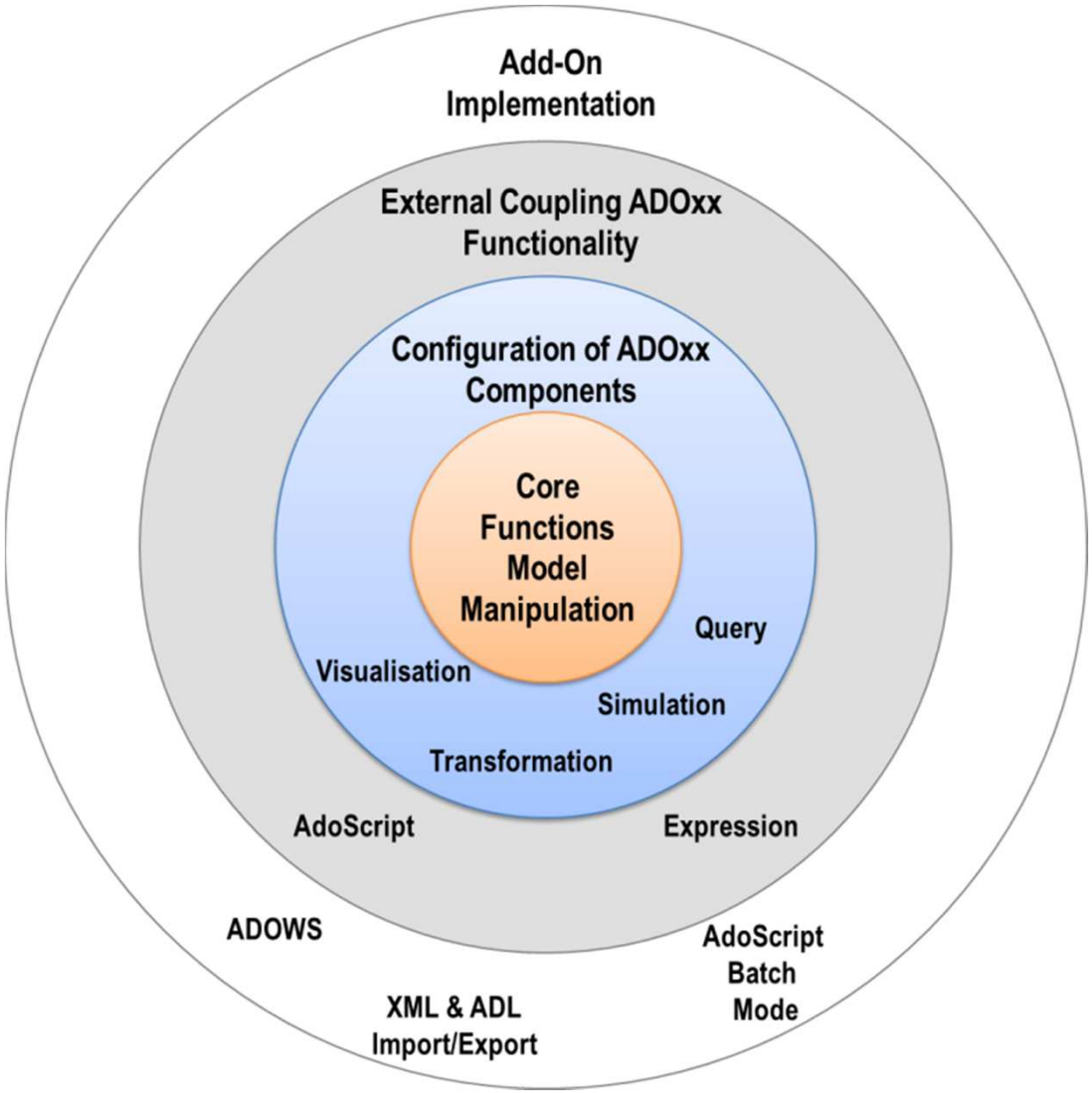
Case:

Logging creation and deletions of Agents in a Agent Model and notification of user whose phone number is registered in the model

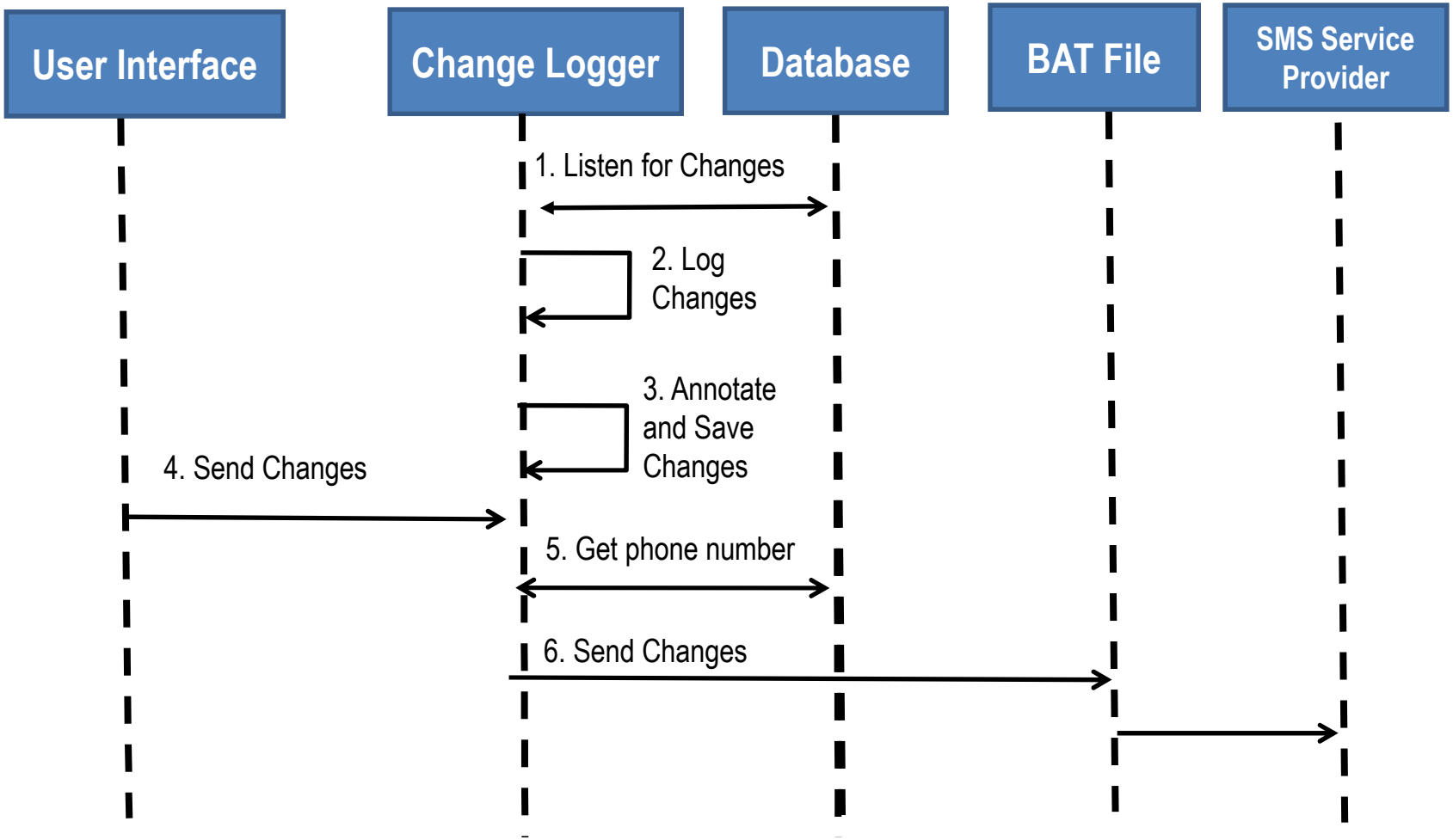
GOAL:

Demonstrate how to log changes made in the models and to notify corresponding users about changes with using SMS

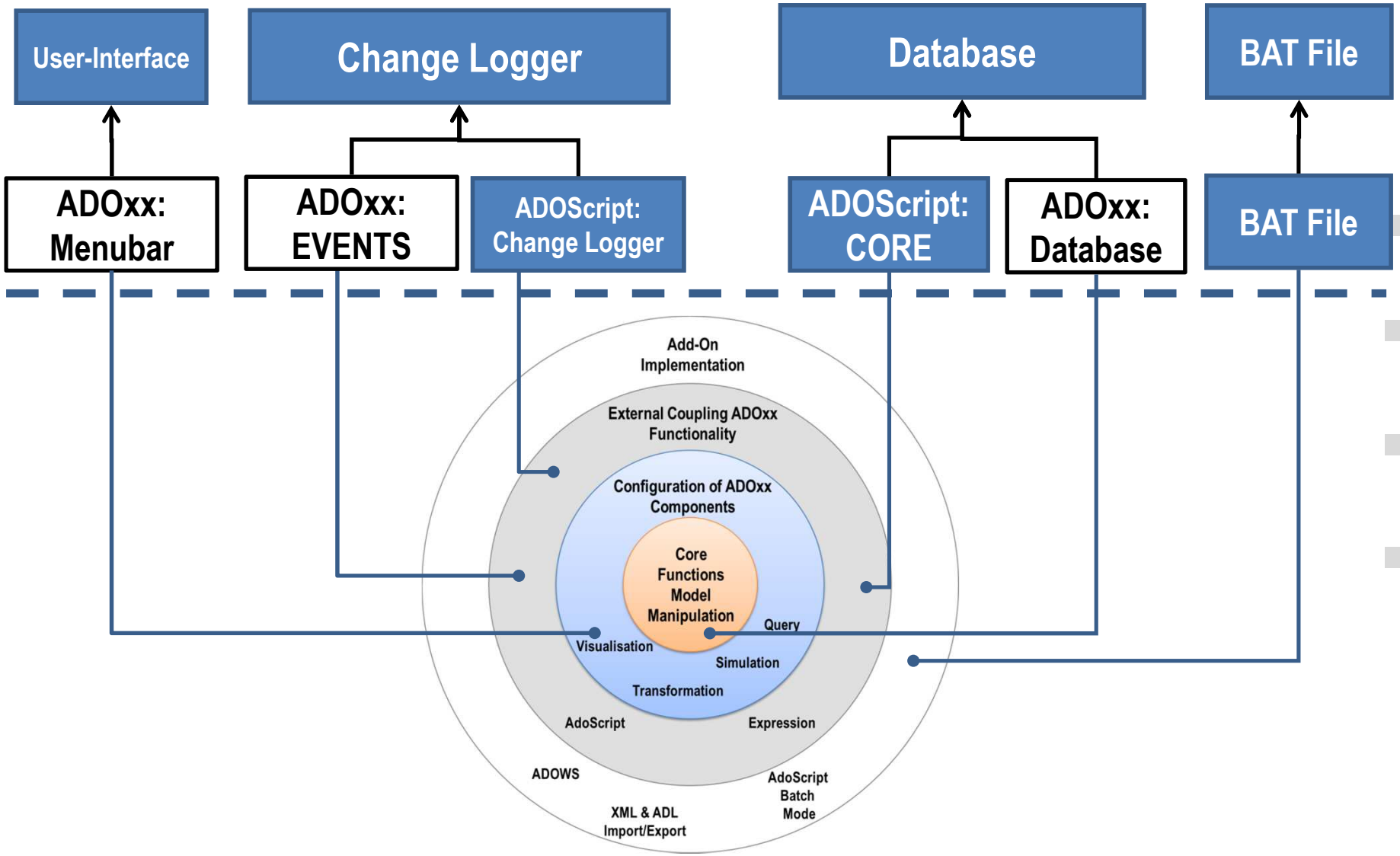
ADOxx Functionality on Meta Level



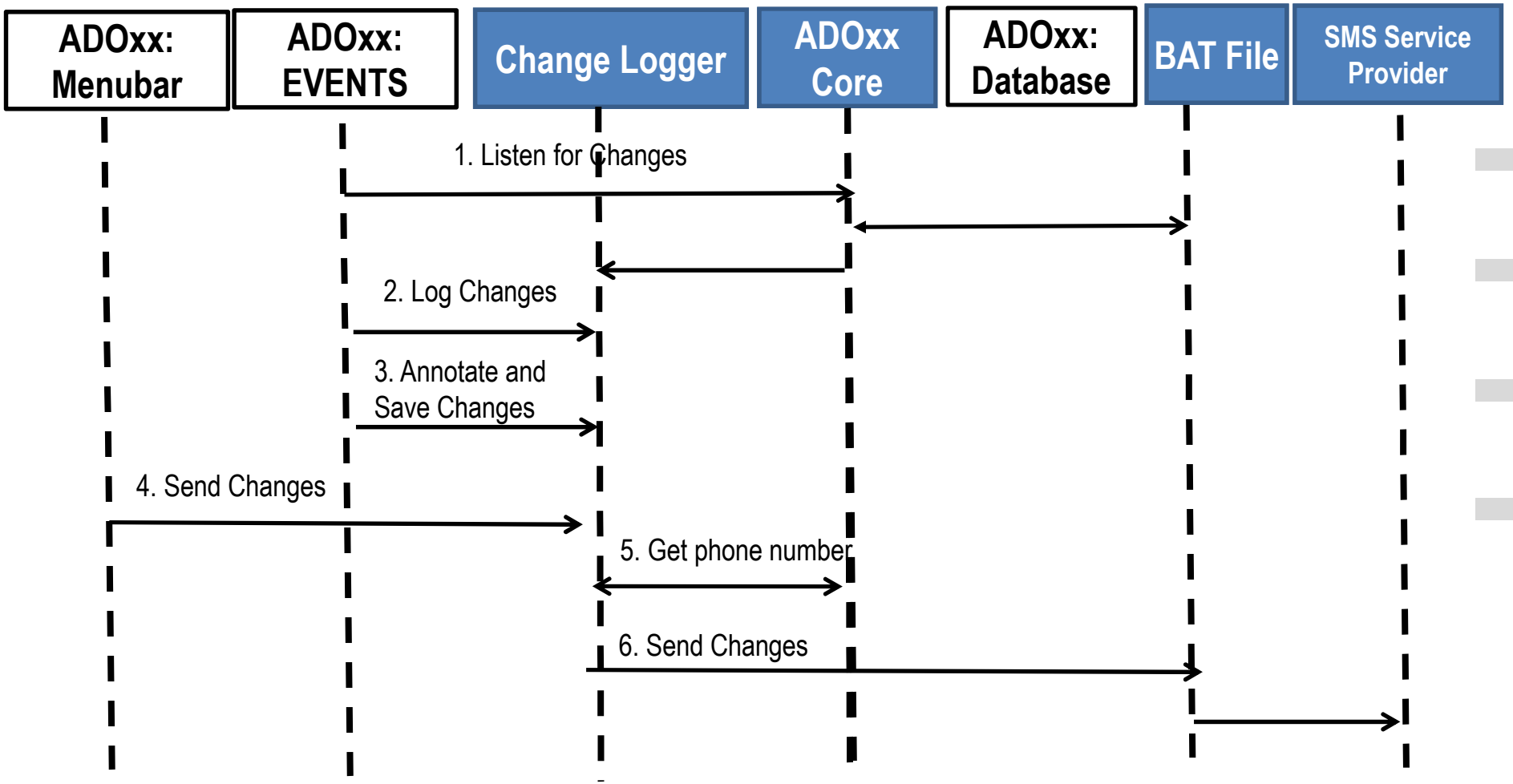
Description of Algorithm



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 Events:** are provided by the platform, which are listening certain events.

HANDS-ON

Change Log - SMS

SMS NOTIFICATION ABOUT CHANGES MADE BY ANOTHER USERS

ADOxx Realisation Hands-On

1. Modelling Language

1. Model Type “Agent Model”
2. New classes “Agent”, “__ModelTypeMetaData__”
3. Add Attributes

2. Configure ADOxx

1. Configure Events
2. Configure Menubar

3. Implement Algorithm with ADOscript

1. ADOscript User Interface
2. Retrieve required information from models
3. Invoking SMS Service

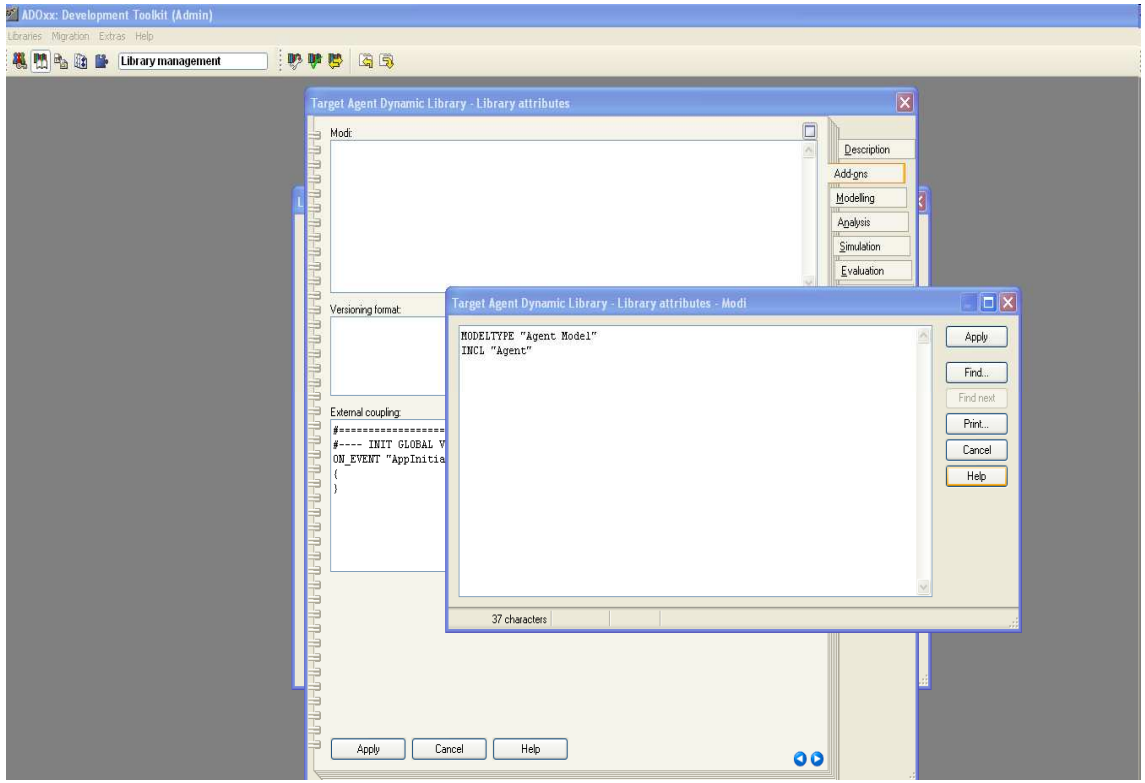
Used ADOxx Functionality: Implementing an Algorithm

- 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

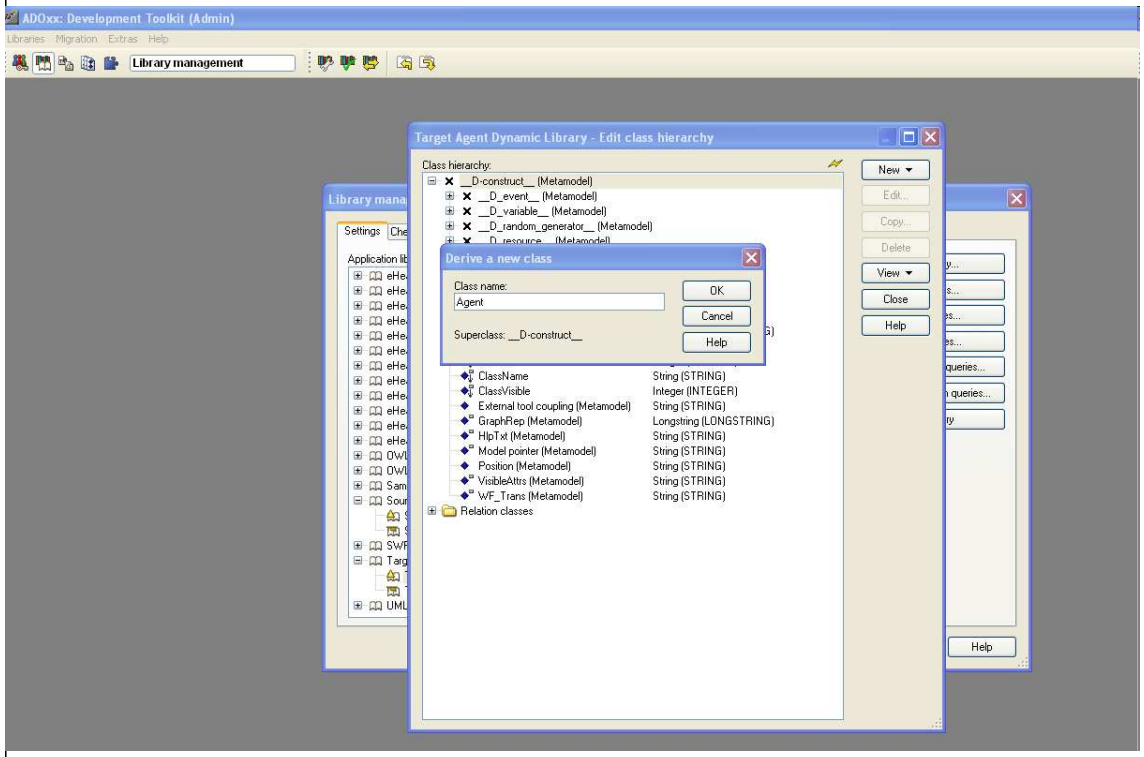
Define new Modeltype “Agent Model”



New Modeltypes:

- Select “Chang Loging 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 “Agent Class”

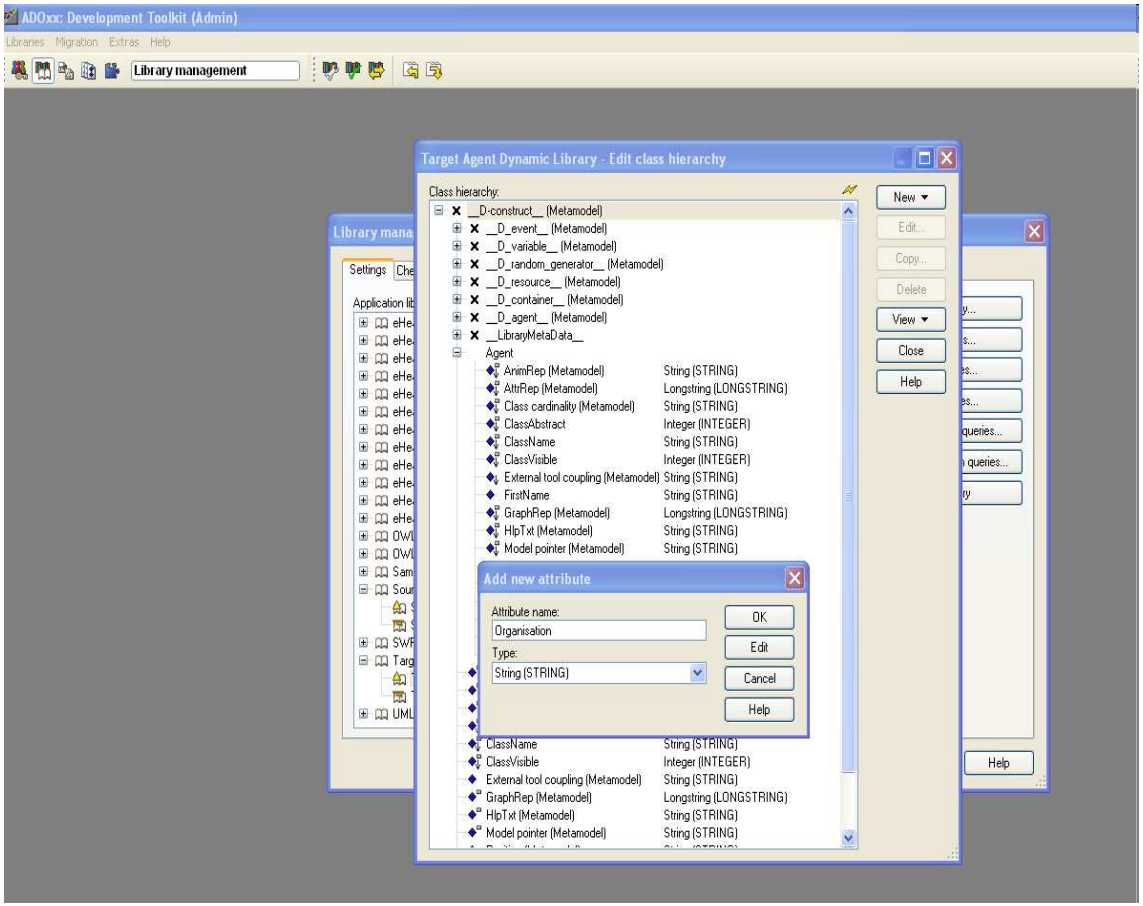
Create New Classes



Create New Classes

- Select “Change Logging 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: ”Agent” and “__ModelTypeMetaData__”
- “Agent” and “__ModelTypeMetaData__” are now sub-clas of __D-construct__

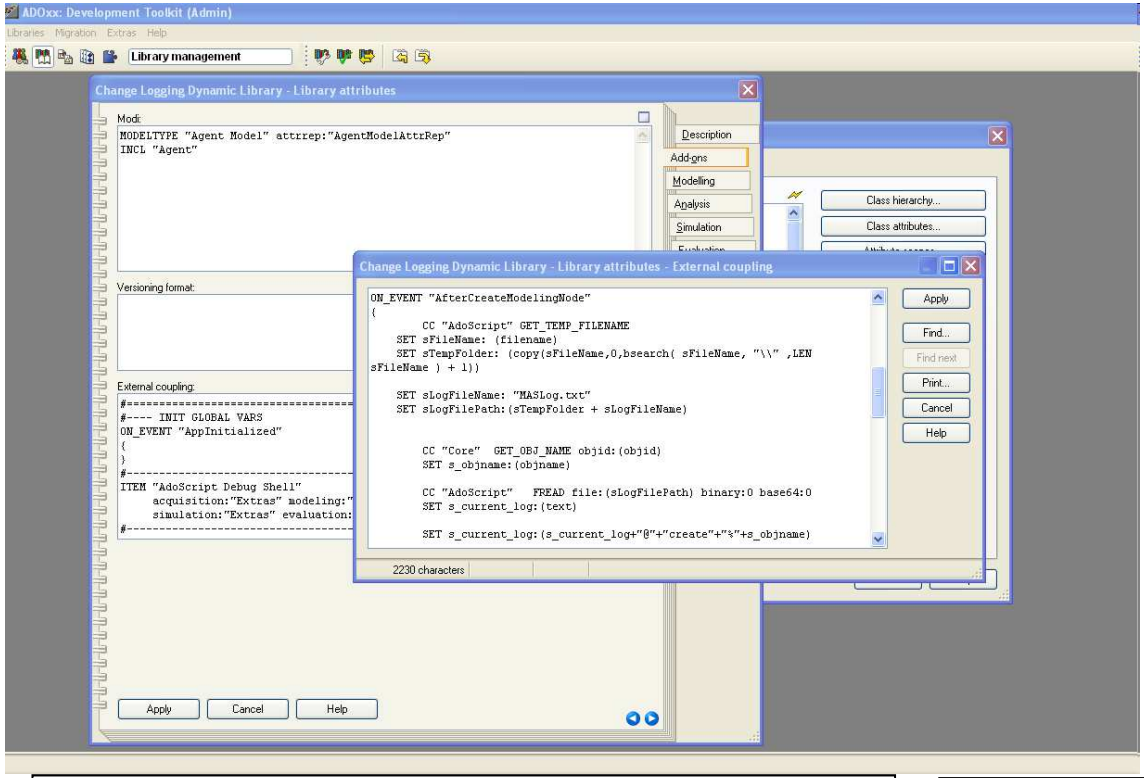
Add Attributes for Classes



Add Attributes

- Select “__ModelTypeMetaData__” and click Newattribute.
- Make “AgentModelAttrRep”, as type LONGSTRING and “Phone number” as type STRING.

Configure Events



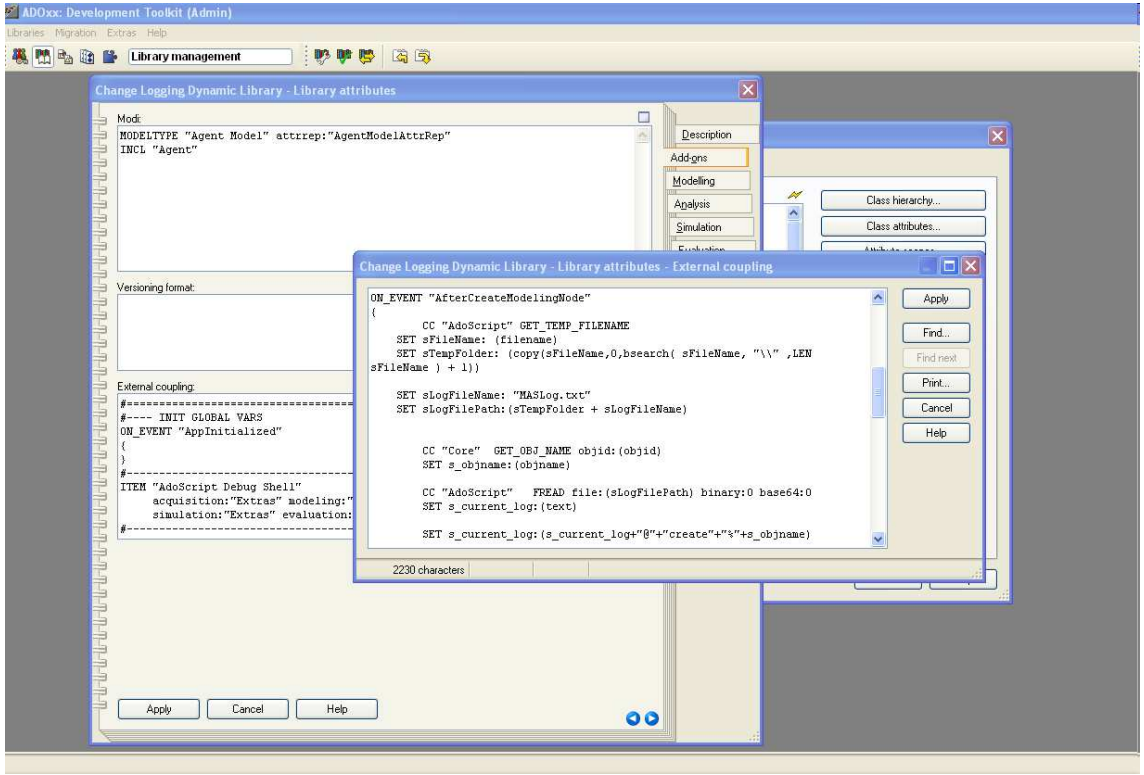
Configure Events:

- Select “Change Loggng Library Dynamic” and open Library attributes.
- Go to Add Ons
- Add the events “AfterCreateModelingNode” and “BeforeDeleteInstance” in the External Coupling attribute and configure them like in given AdoScript codes

```
ON_EVENT "AfterCreateModelingNode"
{
   CC "AdoScript" GET_TEMP_FILENAME
   SET sFileName: (filename)
   SET sTempFolder: (copy(sFileName,0,bsearch( sFileName, "\\"
,LEN sFileName ) + 1))

   SET sLogFileName: "MASLog.txt“
…
```

```
ON_EVENT "BeforeDeleteInstance"
{
                CC "AdoScript" GET_TEMP_FILENAME
   SET sFileName: (filename)
   SET sTempFolder: (copy(sFileName,0,bsearch( sFileName, "\\"
,LEN sFileName ) + 1))

   SET sLogFileName: "MASLog.txt"
…
```

Configure Menubar



Configure Menubar:

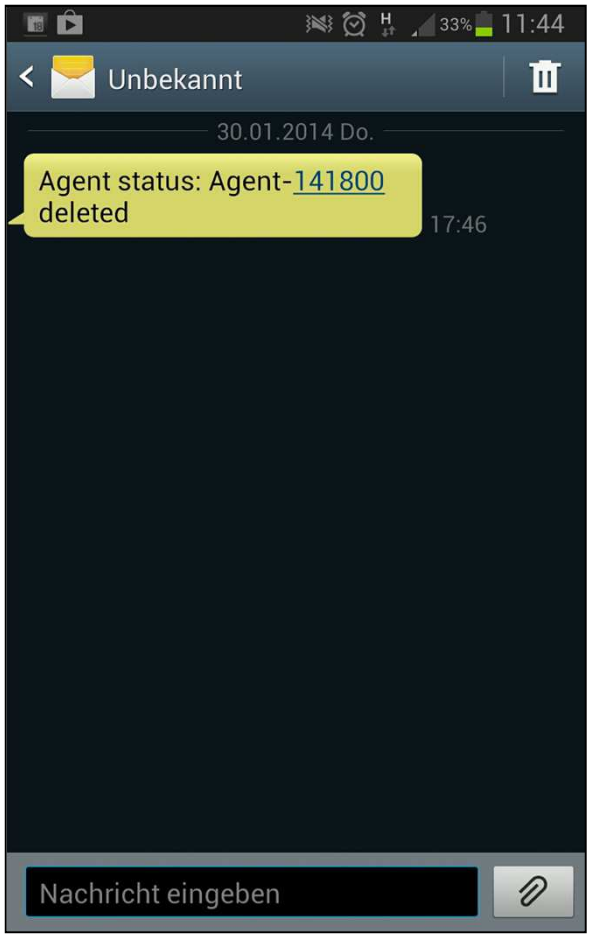
- Select “Change Logging Library Dynamic” and open Library attributes.
- Go to Add Ons
- Add the following code in the External Coupling attribute

```
ITEM "Send Changes by SMS"
    acquisition:"Extras" modeling:"Extras" analysis:"Extras“  simulation:"Extras" evaluation:"Extras“
importexport:"Extras"

EXECUTE file:("db:\\sendChangeReport.asc")
```

Implement and Import ADOscript File into Database

```
CC "Modeling" GET_ACT_MODEL
SET sActualModel: ( modelid )

CC "AdoScript" GET_TEMP_FILENAME
SET sFileName: (filename)
SET sTempFolder: (copy(sFileName,0,bsearch( sFileName, "\\" ,LEN sFileName ) + 1))

SET sBatfileName: "SendSMS.bat"
SET sBatfilePath: (sTempFolder + sBatfileName)
CC "AdoScript"  FILE_COPY from: ("db:\\" + sBatfileName) to: (sBatfilePath)
SET sLogFileName: "MASLog.txt"
SET sLogFilePath:(sTempFolder + sLogFileName)

CC "AdoScript"   FREAD file:(sLogFilePath) binary:0 base64:0
SET s_current_log:(text)
SET t2:("Agent status: number of agents unchanged.")
SET t3:("Agent status: ")
FOR s_item in:(s_current_log) sep:("@")
{
        SET s_item_annotation:(token(s_item,0,("%")))
        SET s_item_objname:(token(s_item,1,("%")))
        IF (s_item_annotation ="create")
        {
                SETL s_message:(s_item_objname+" created ")
                SET t3:(t3+s_message)
                SET t2:(t3)
        }

        IF (s_item_annotation ="delete")
        {
                SETL s_message:(s_item_objname+" deleted ")
                SET t3:(t3+s_message)
                SET t2:(t3)
        }
        IF
}
...
```

Result



Description

•As the user click on the item “Send Changes by SMS” in the menubar, mechanism sends last changes to the phone number of the user to whom model belongs

Further Questions?

www.adoxx.org

tutorial@adoxx.org