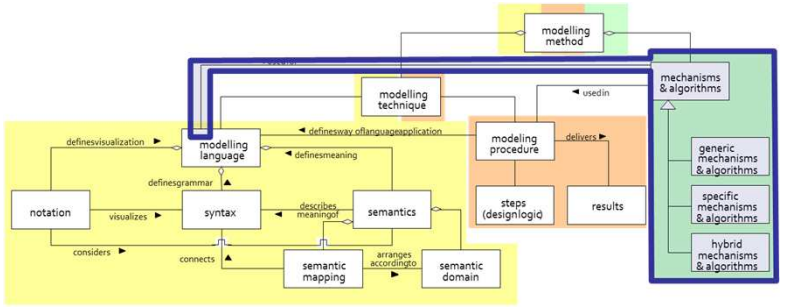


Change Log – Graphical Representation

SCENARIO:
NOTIFICATION ABOUT CHANGES MADE BY ANOTHER USERS

Scenario Description

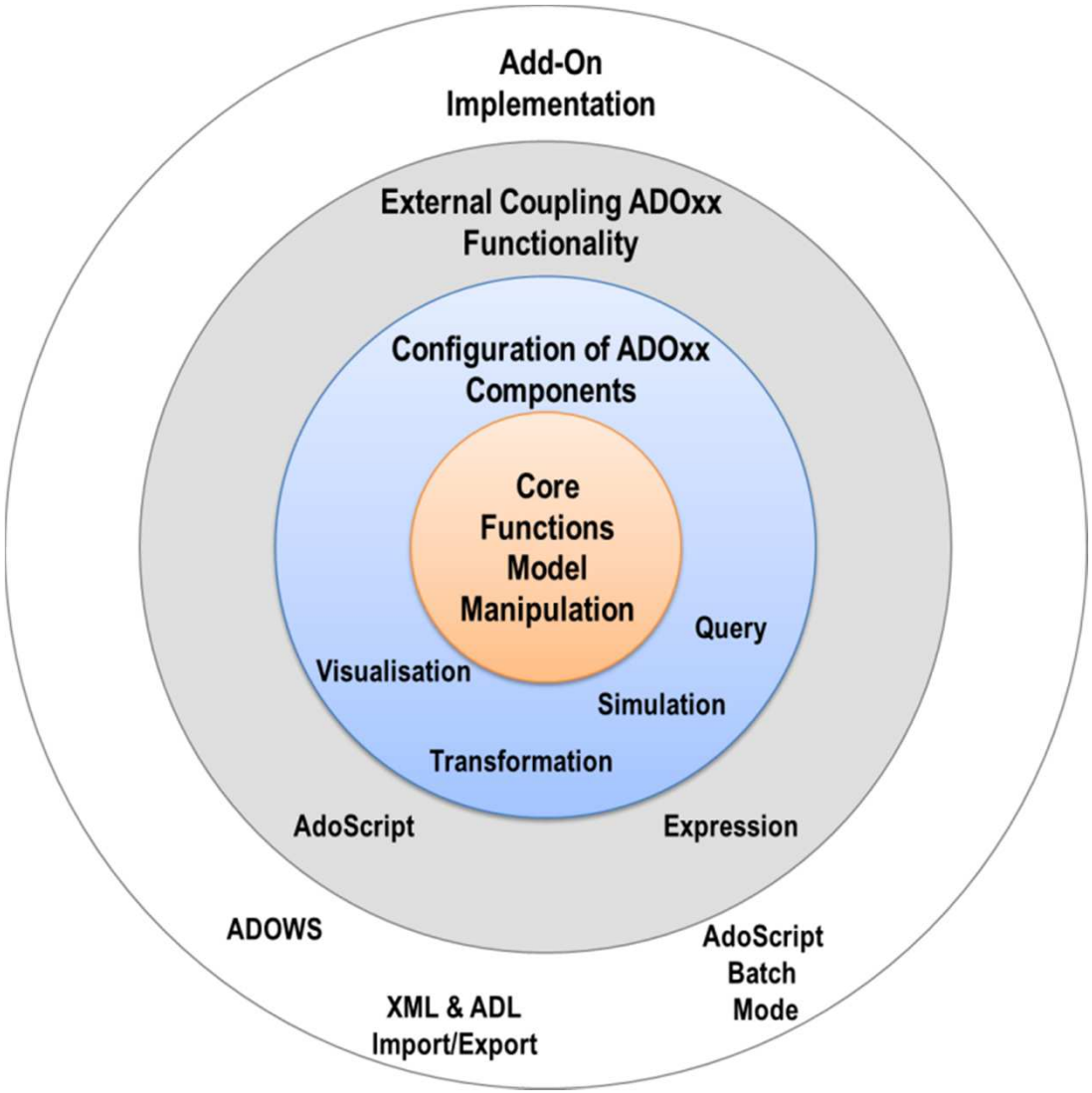
Case:

Logging creation and deletions of Agents in a Agent Model and notification of users through the dynamic Model GraphRep

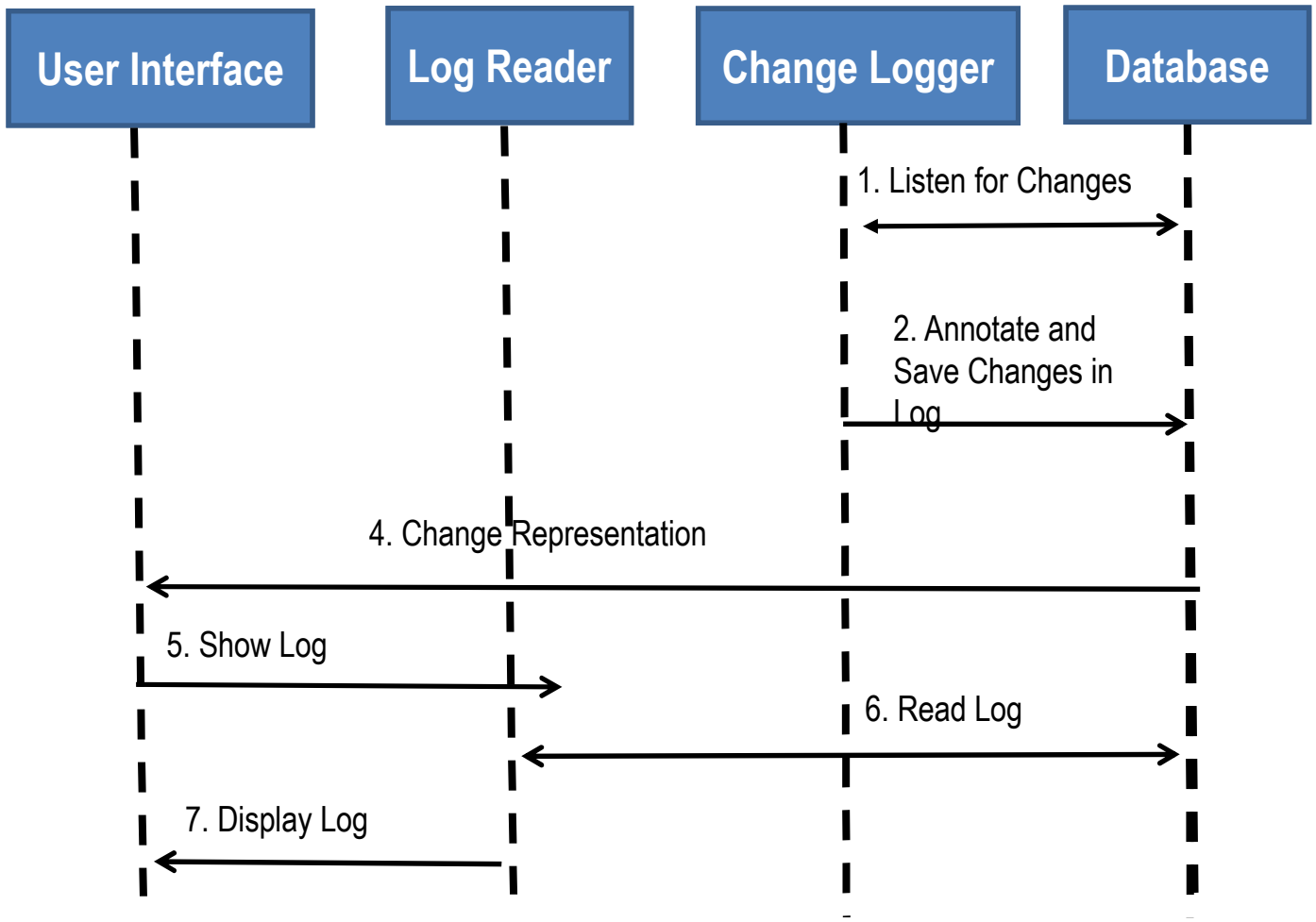
GOAL:

Demonstrate how to log changes made in the models and to notify corresponding users about changes with using dynamic Model Type GraphRep

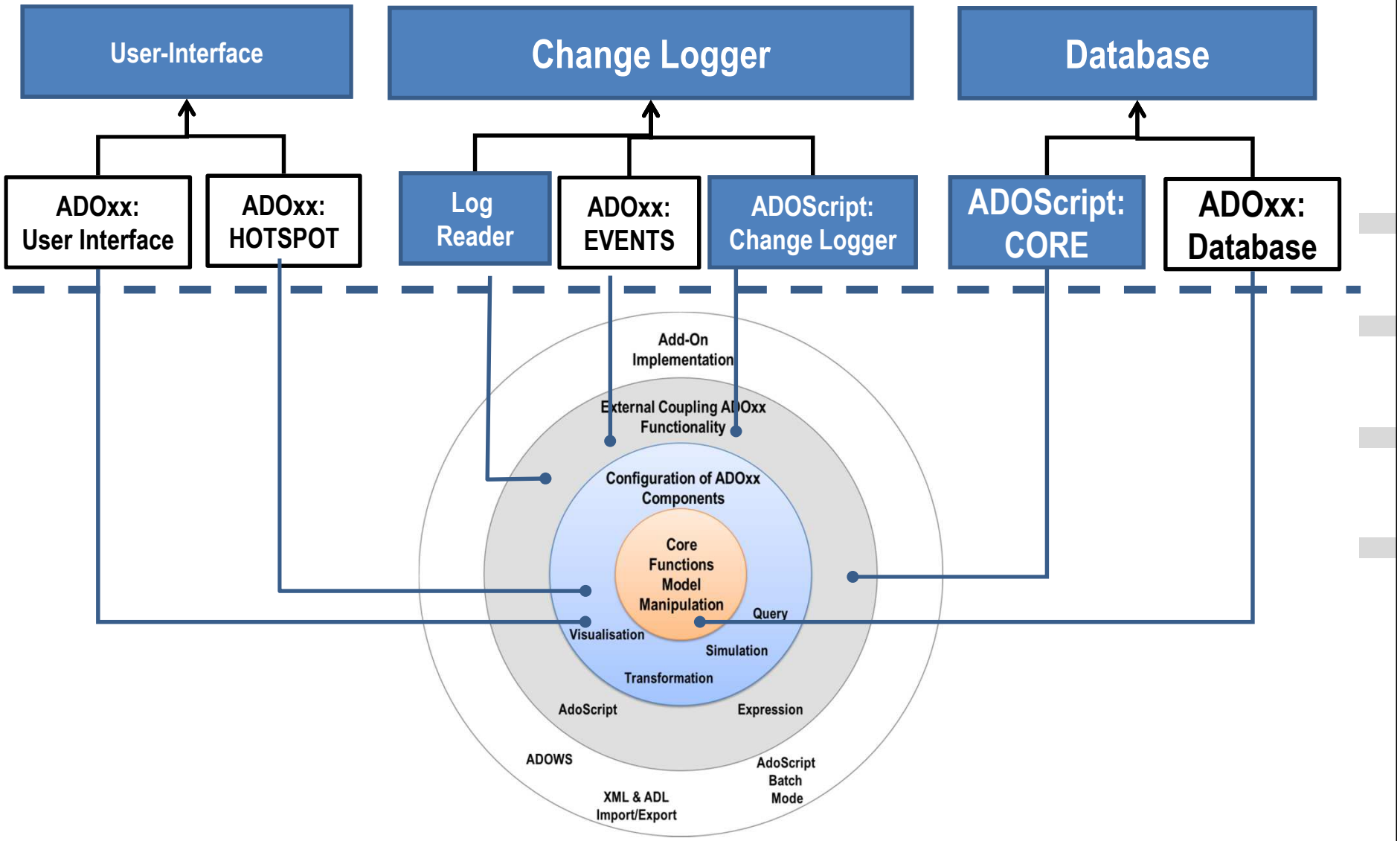
ADOxx Functionality on Meta Level



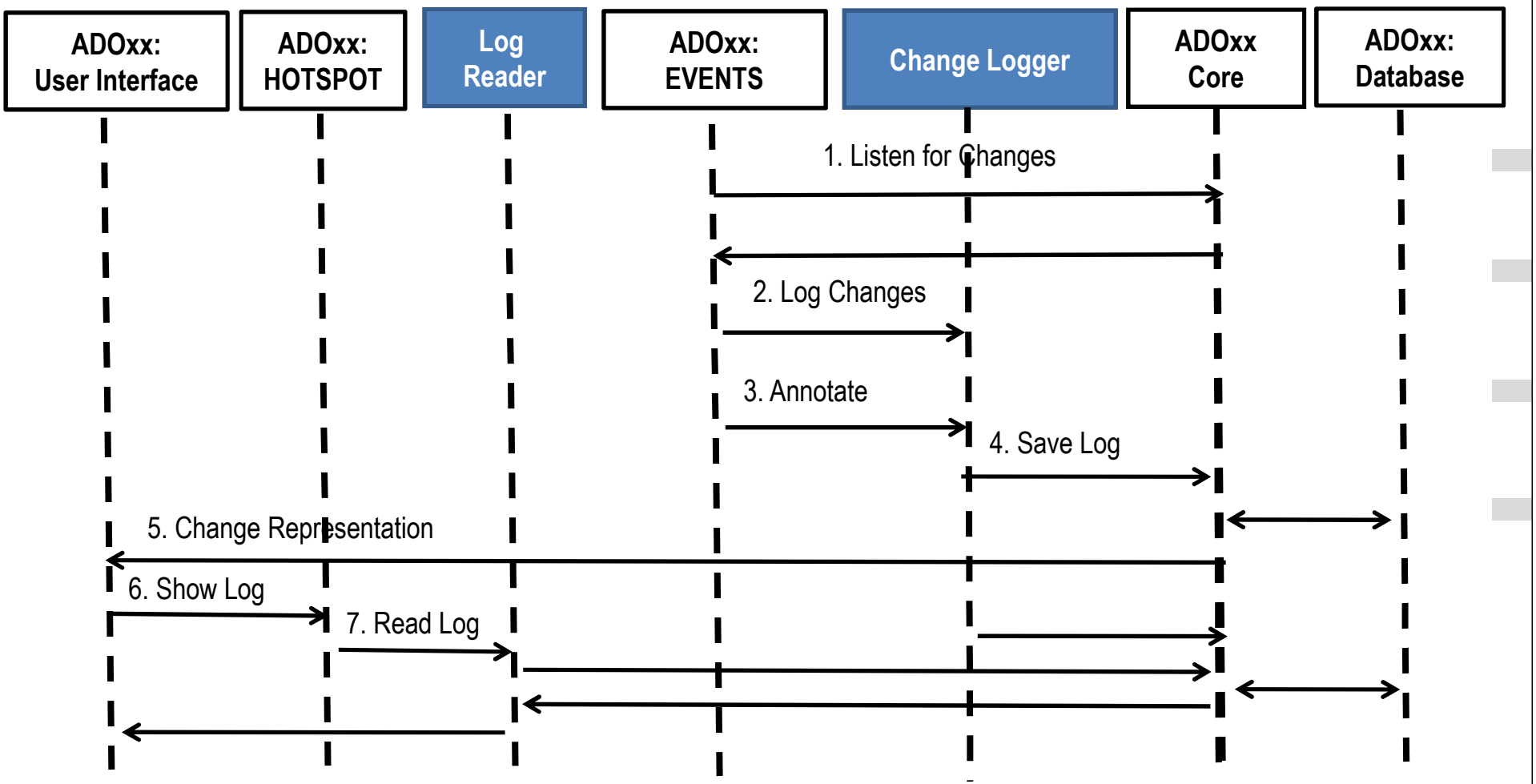
Description of Algorithm



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

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 and integrate Change Logger AdoScript
2. Configure Model GraphRep

3. Implement and Configure ADOscript

1. Implement and Configure Log Reader AdoScript

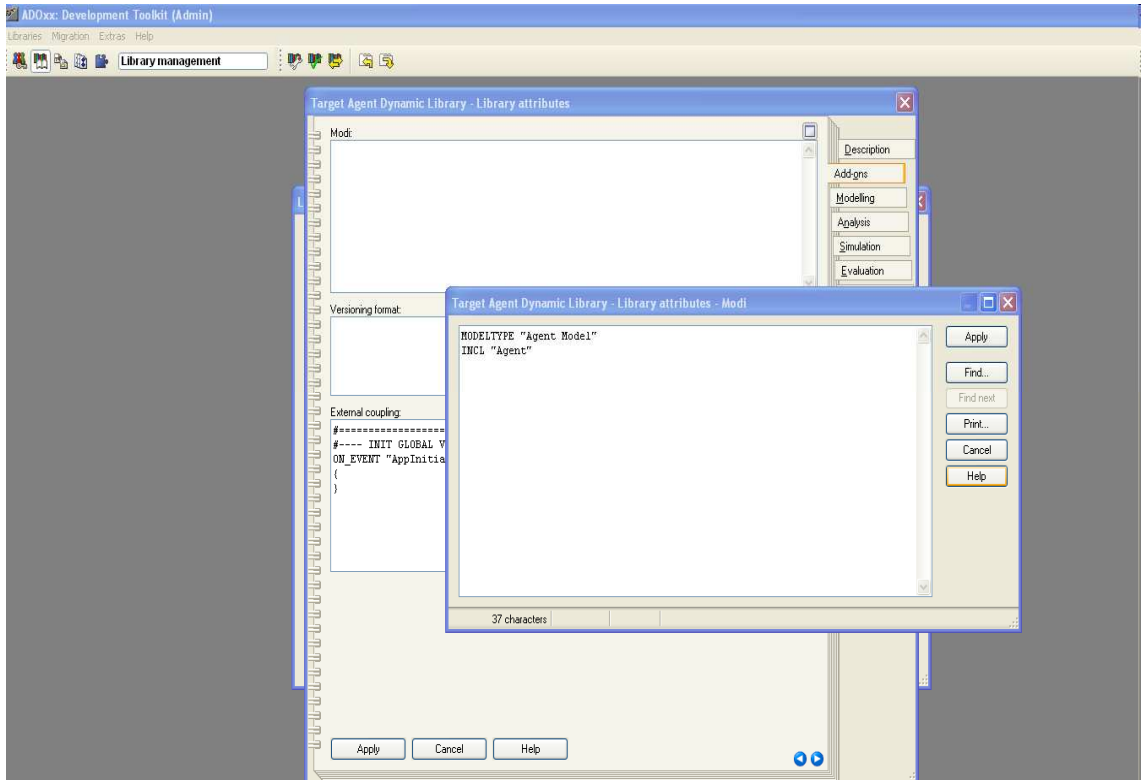
Used ADOxx Functionality: Implementing an Algorithm

- 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 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

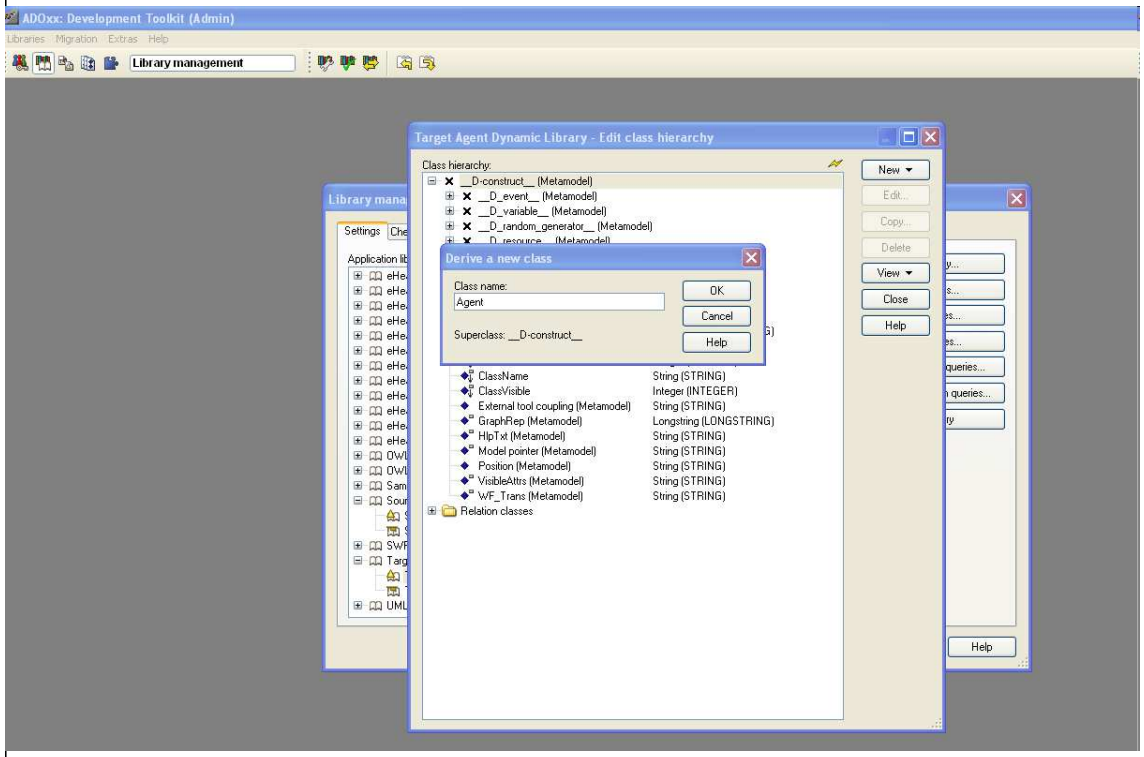
Define new Modeltype “Agent Model”



New Modeltypes:

- Select “Chang Loging GraphRep Dynamic Library” 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

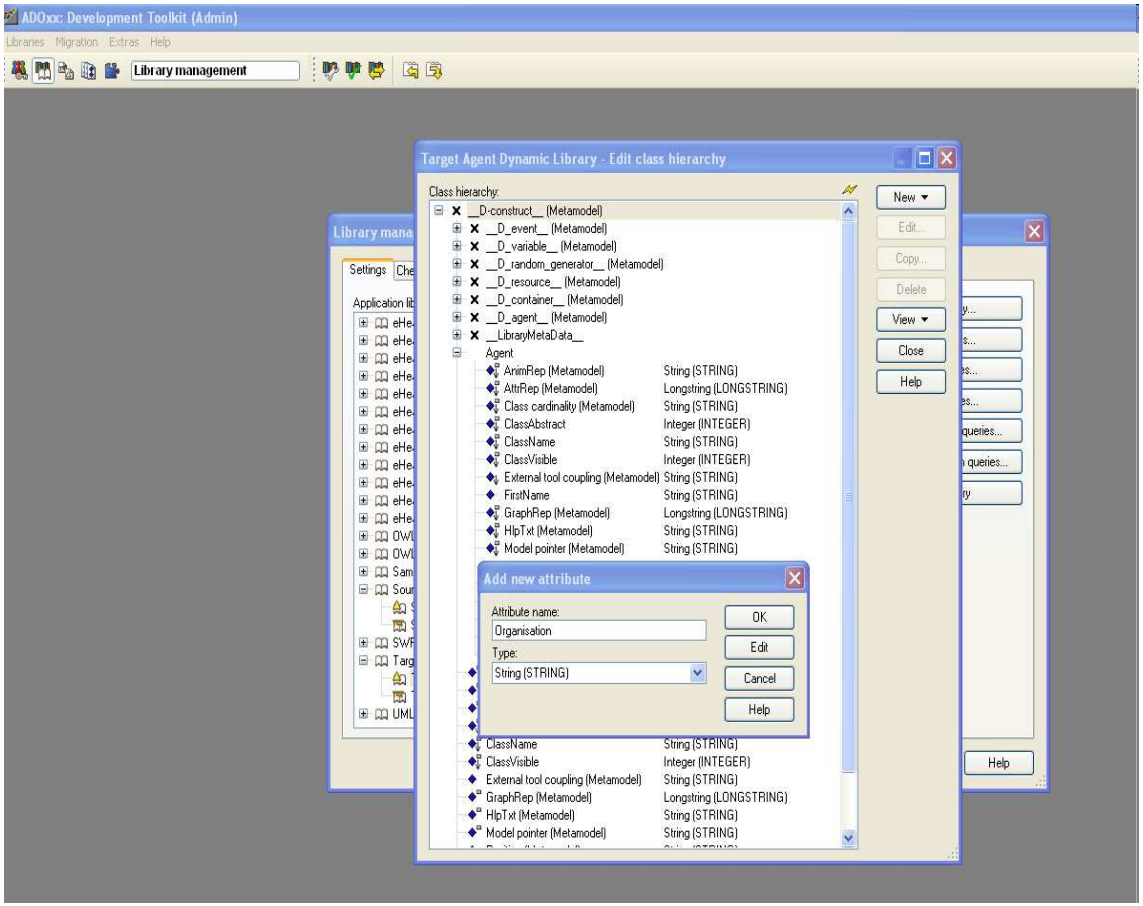
Create New Classes



Create New Classes

- Select “Change Logging GraphRep 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__

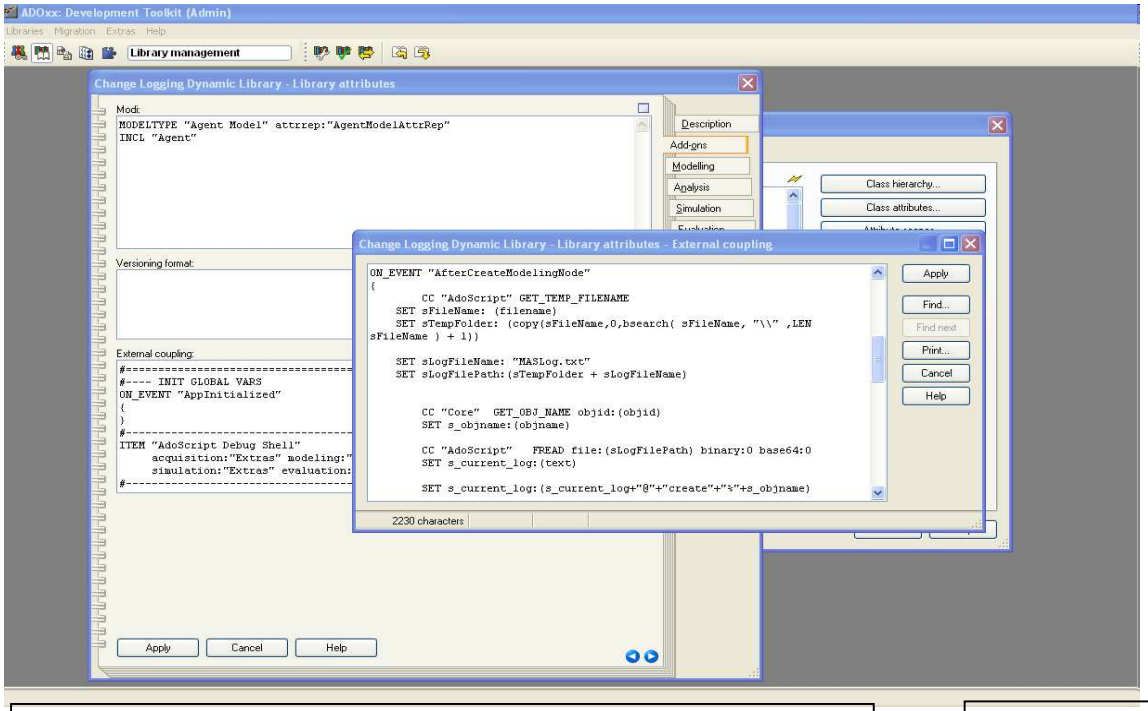
Add Attributes for Classes



Add Attributes

- Select “__ModelTypeMetaData__” and click Newattribute.
- Make “AgentModelAttrRep”, as type LONGSTRING and “ReadLog” as type PROGRAMCALL.

Configure Events



Configure Events:

•Select “Change Loggng Library Dynamic” and open Library attributes.

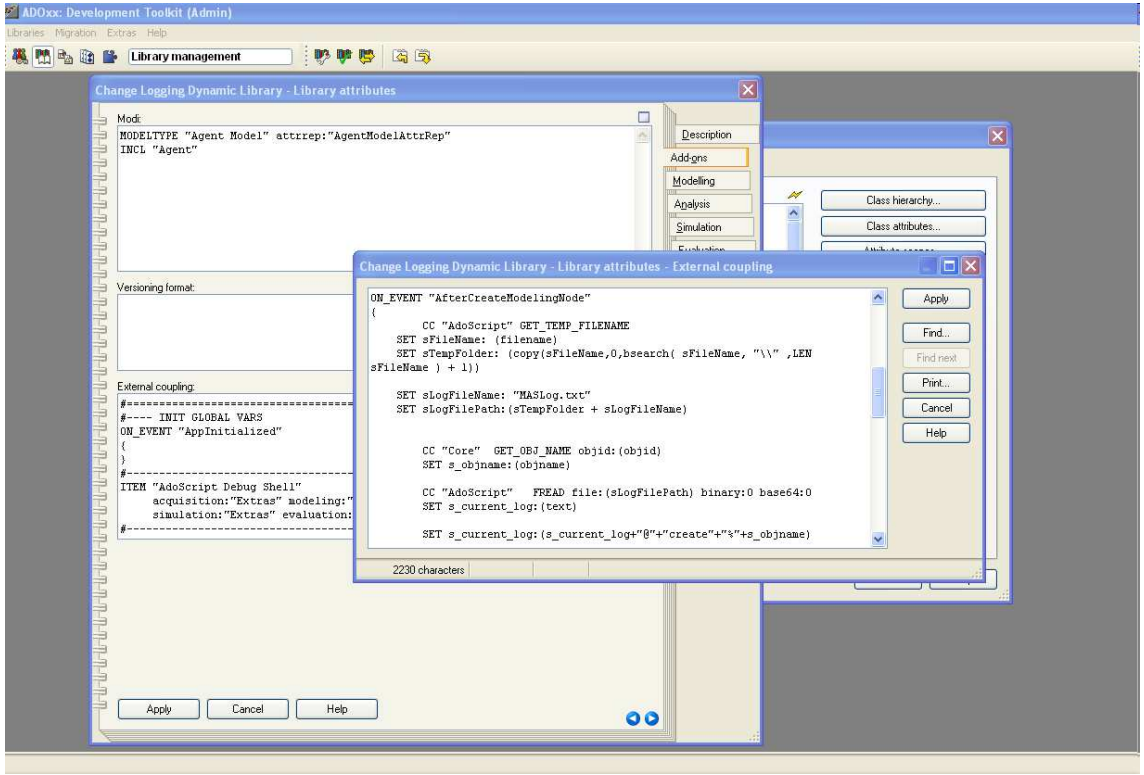
•Go to Add Ons

•Add the events “AfterCreateModelingNode” and “BeforeDeleteInstance” in the External Coupling attribute and configure them and integrate Change Logger AdoScript like;

```
ON_EVENT "AfterCreateModelingNode"
{
    CC "Core"  GET_OBJ_NAME objid:(objid)
    SET s_objname:(objname)
    SET s_objname:(s_objname+" is created")
    CC "Core" GET_ATTR_VAL objid:(modelid) attrname:("Created Objects")
    SET s_temp_co:(val)
    SET s_created_objects:(tokcat(s_temp_co,s_objname,"\n"))
    CC "Core" SET_ATTR_VAL objid:(modelid) attrname:("Created Objects") val:(s_created_objects)
}
```

```
ON_EVENT "BeforeDeleteInstance"
{
  CC "Core"  GET_OBJ_NAME objid:(instid)
  SET s_objname:(objname)
  SET s_objname:(s_objname+" is deleted")
  CC "Core" GET_ATTR_VAL objid:(modelid) attrname:("Deleted Objects")
  SET s_temp_do:(val)
  SET s_deleted_objects:(tokcat(s_temp_do,s_objname,"\n"))
   CC "Core" SET_ATTR_VAL objid:(modelid) attrname:("Deleted Objects") val:(s_deleted_objects)
}
```

Configure Menubar



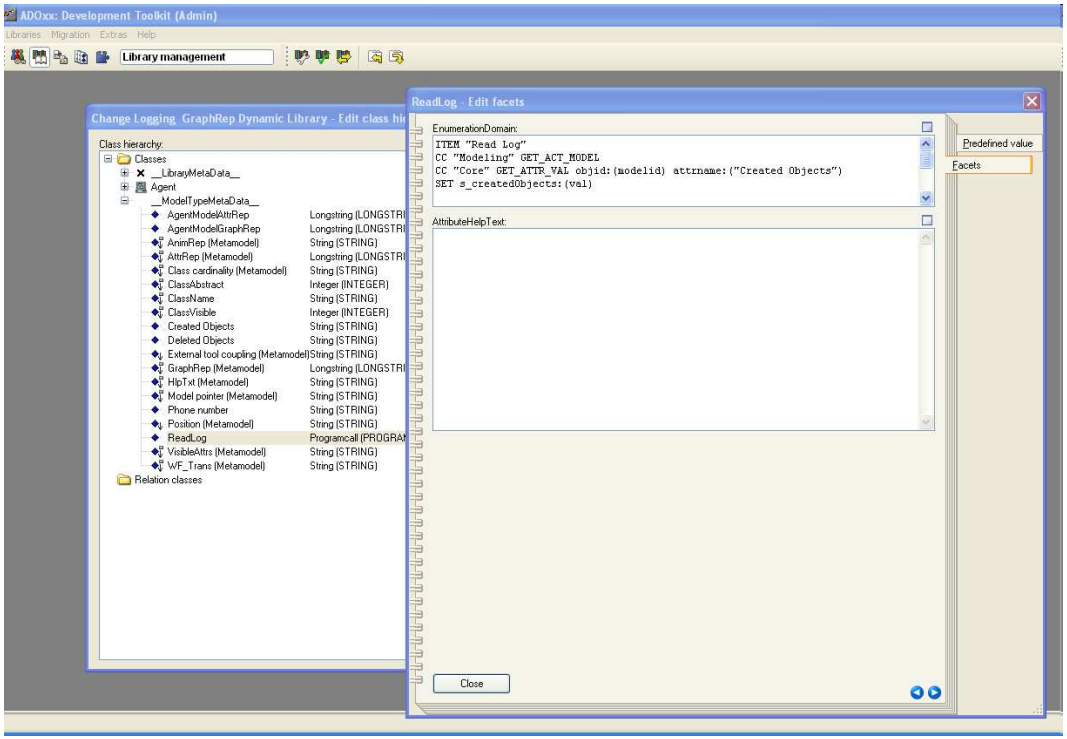
Configure Menubar:

- 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 Configure ADOscript

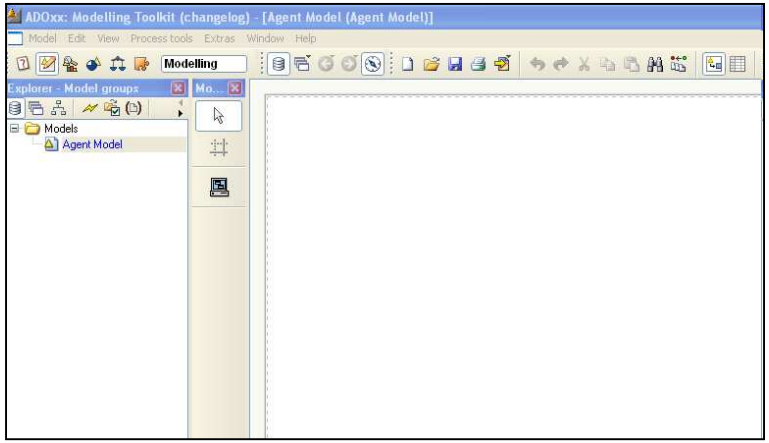


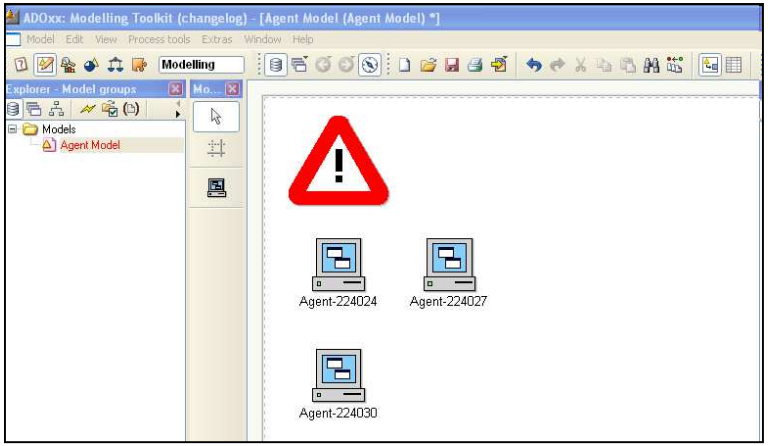
Configure Programcall:

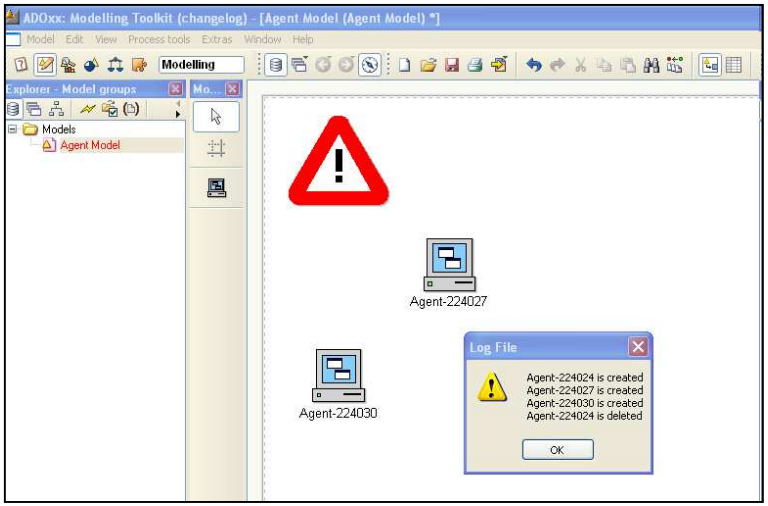
- Select “Change Logging Library Dynamic” and open Class Hierarchy.
- Open Class “__ModelTypeMetaData__”
- Select Attribute “ReadLog” and click Edit button
- Set “Read Log” as standart value
- Open Facets amd add the following code in the External Coupling attribute

```
ITEM "Read Log"
CC "Modeling" GET_ACT_MODEL
CC "Core" GET_ATTR_VAL objid:(modelid) attrname:("Created Objects")
SET s_createdObjects:(val)
CC "Core" GET_ATTR_VAL objid:(modelid) attrname:("Deleted Objects")
SET s_deletedObjects:(val)
SET s_log:(tokcat(s_createdObjects, s_deletedObjects, "\n"))
CC "AdoScript" WARNINGBOX (s_log) title:("Log File")
```

Result







Description:

- In case of creation and/or deletion of agent log and model graph will be update in order to notify users
- If user clicks on notificaiton sign log with list of created and deleted agents is will be displayed.

Further Questions?

www.adoxx.org

tutorial@adoxx.org