

Cooperative Attribute-Scripted

SCENARIO:

**Implement Correlation Calculation
Mechanism using Cooperative Attribute**

Scenario Description



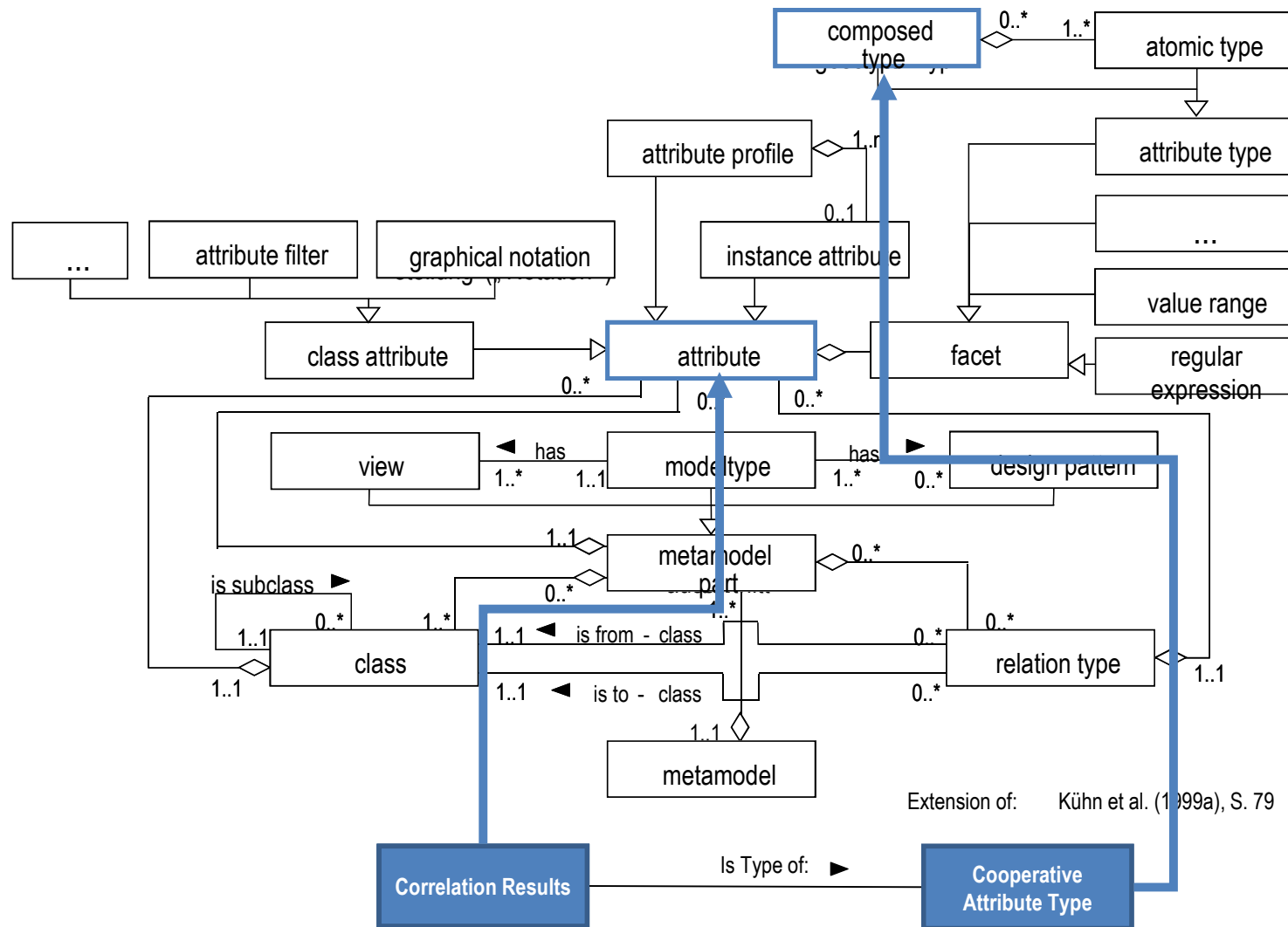
Case:

Realization of Correlation Results attribute which consist of calculated correlation between preferences of two users. The preferences are differentiating by dimensions and the categories under dimensions. The correlation calculation is realized by AdoScript.

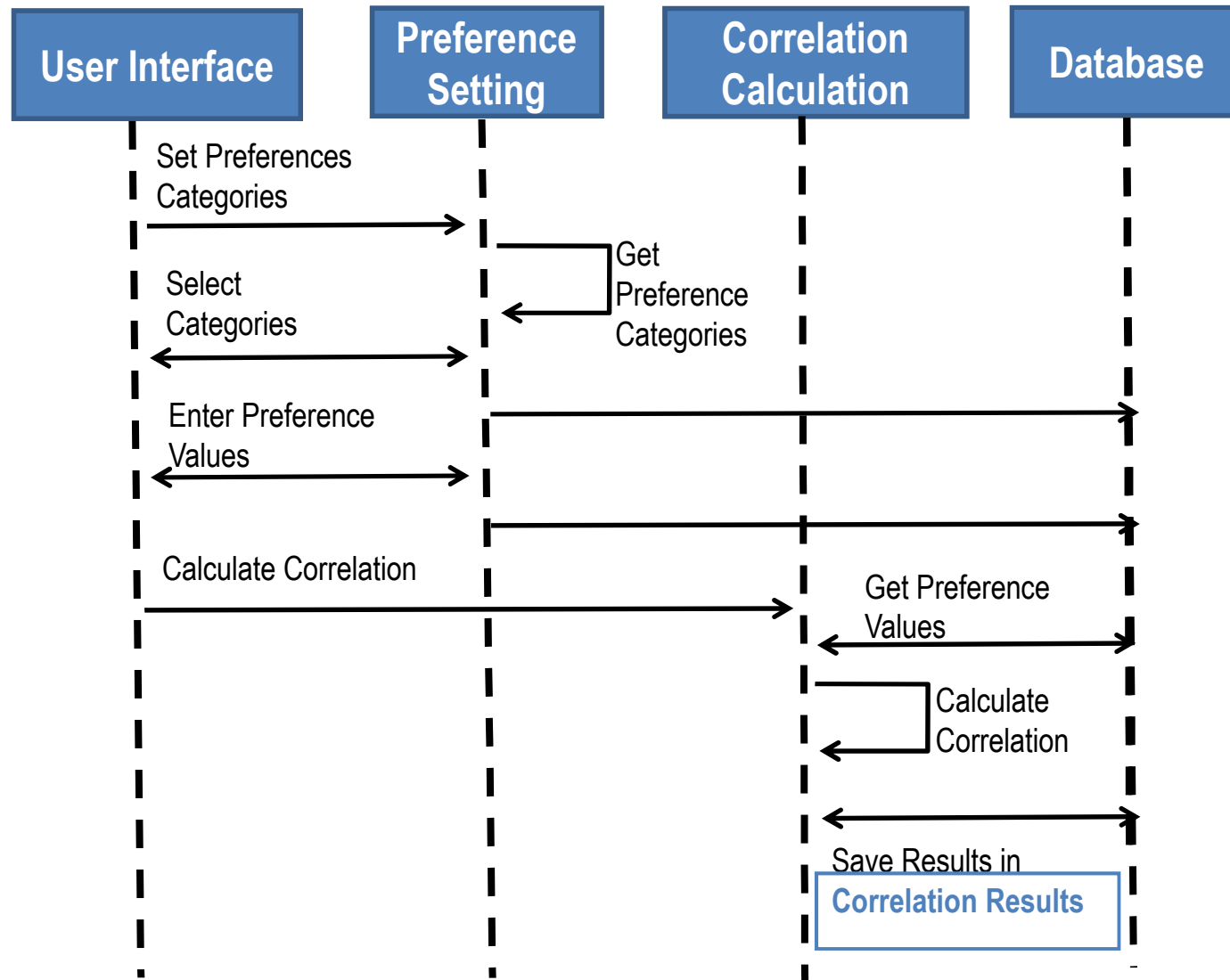
GOAL:

Demonstrate how a cooperative attribute can be realised consisted of calculated common preferences.

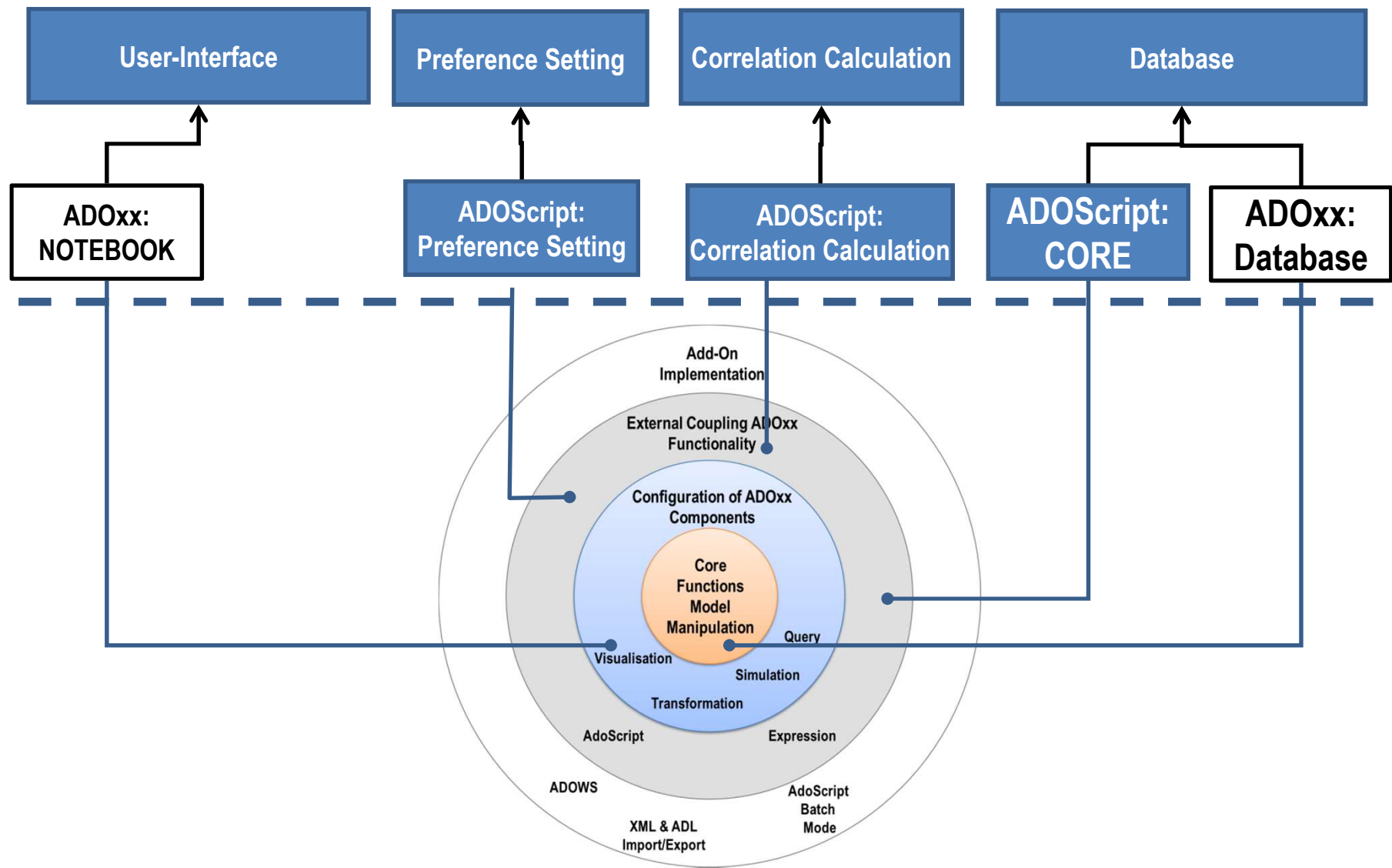
Meta Model of Meta Modelling Language



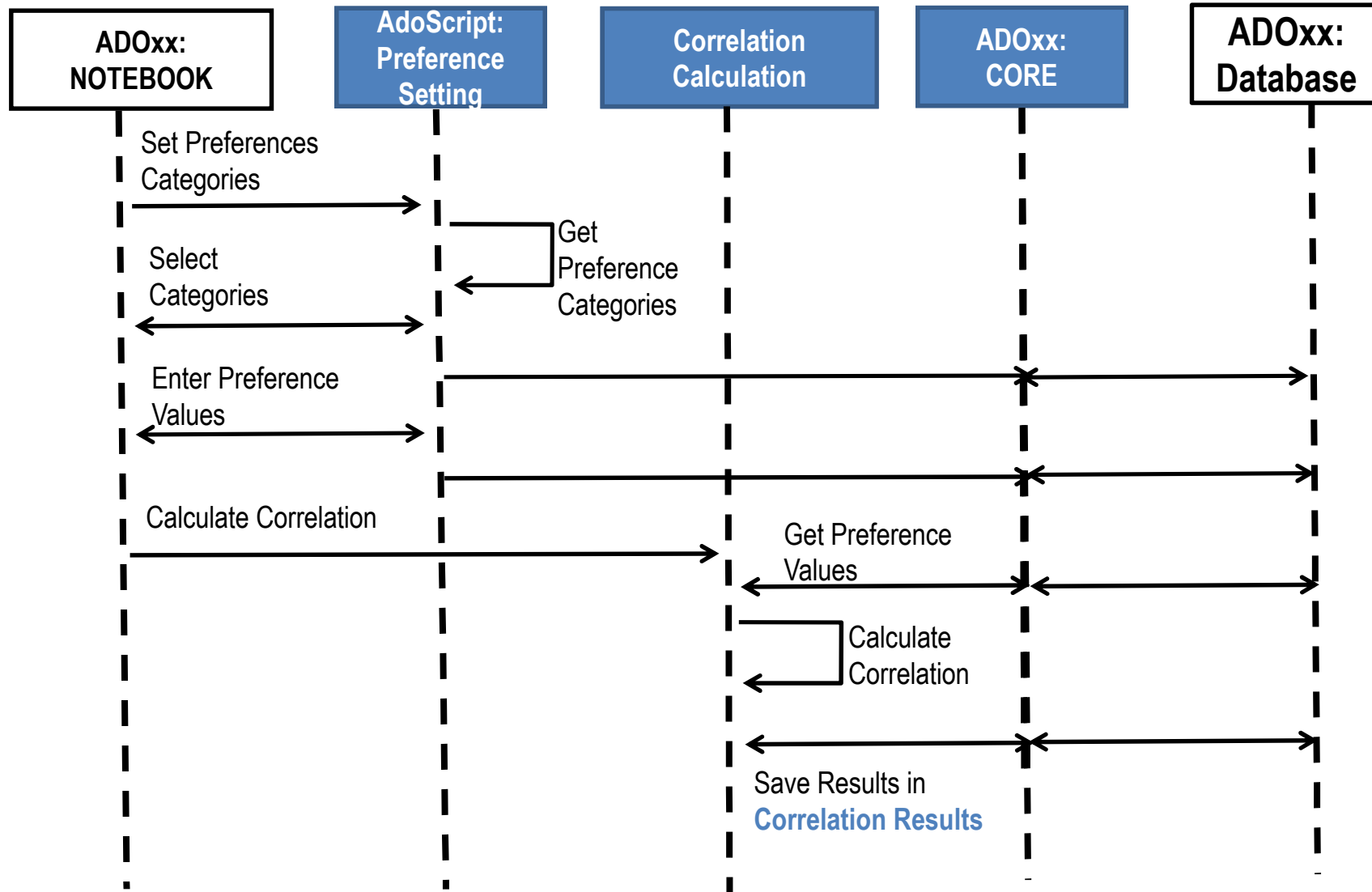
Description of Algorithm



Mapping ADOxx Functionality



ADOxx Realisation Approach



Added Value of Metamodelling Platform



Used meta-modelling functionality for realisation of the scenario:

- **Complex Attribute Types (Record Class)**
- **AttrRep (NOTEBOOK):**
- **Attribute Type: PROGRAMCALL**
- **AdoScript:**



ADOxx Realisation Hands-On

1. Realisation of Modelling Language

1. Define Model Types “Space Model”, “Preference Model”
2. New class “Interaction Process”, “Preference”
3. Add Complex Attribute Type
4. Add Attributes

2. Implement Algorithm with ADOscript

1. Preference Setting
2. Correlation Calculation

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	



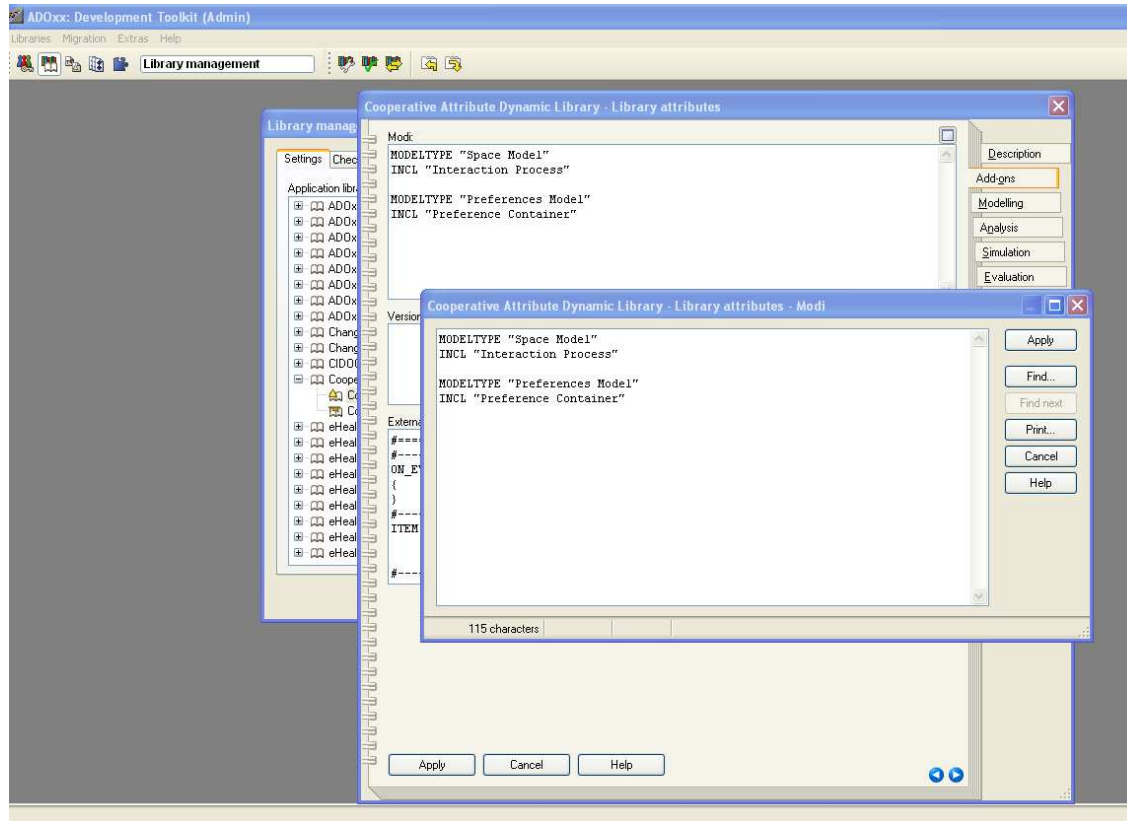
HANDS-ON

Cooperative Attribute-Scripted

SCENARIO:

**Implement Correlation Calculation
Mechanism using Cooperative Attribute**

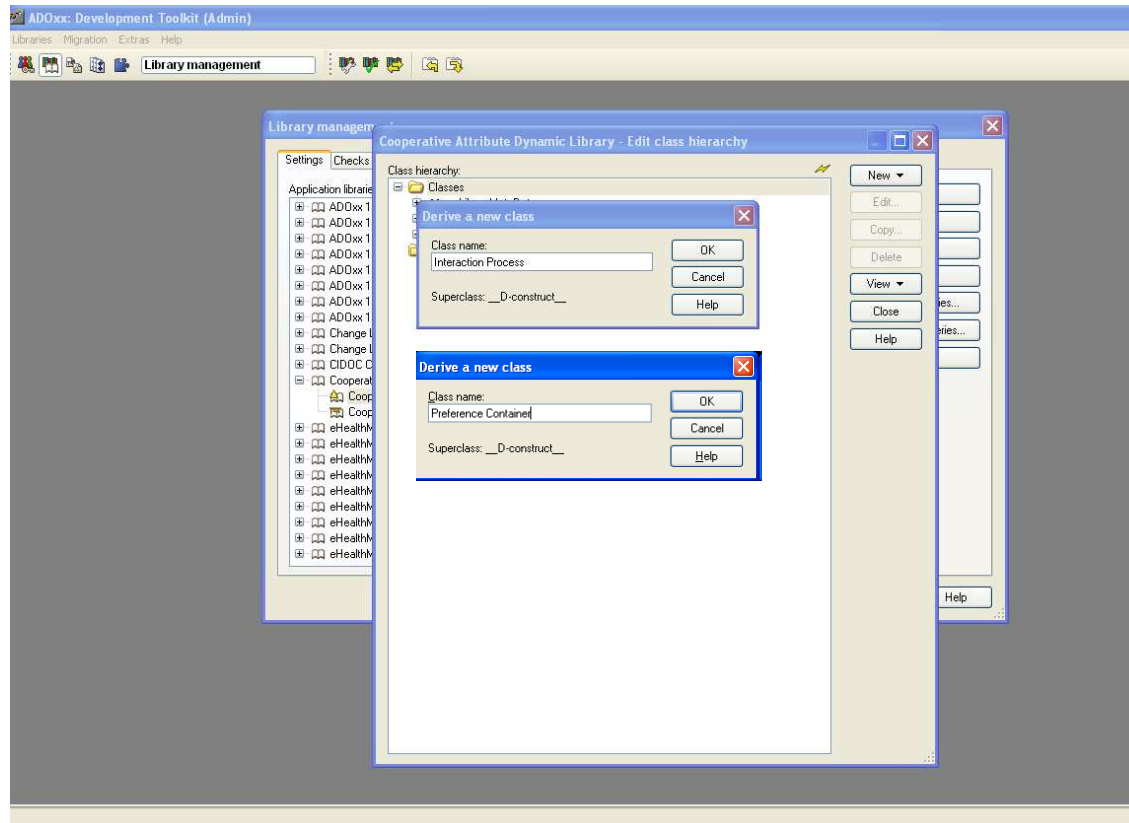
Define Model Types “Space Model” and “Preference Pool Model”



New Modeltypes:

- Select “Cooperative Attribute Dynamic Library” and open Library attributes.
- Got to Add Ons
- Add the Modeltypes “Space Model” and “Prefences Pool Model” in the Modi attribute
- When the classes are defined, you need to INCLUDE “Interaction Process” under “Space Model” and “Preferences Container” under “Preferences Pool Model”

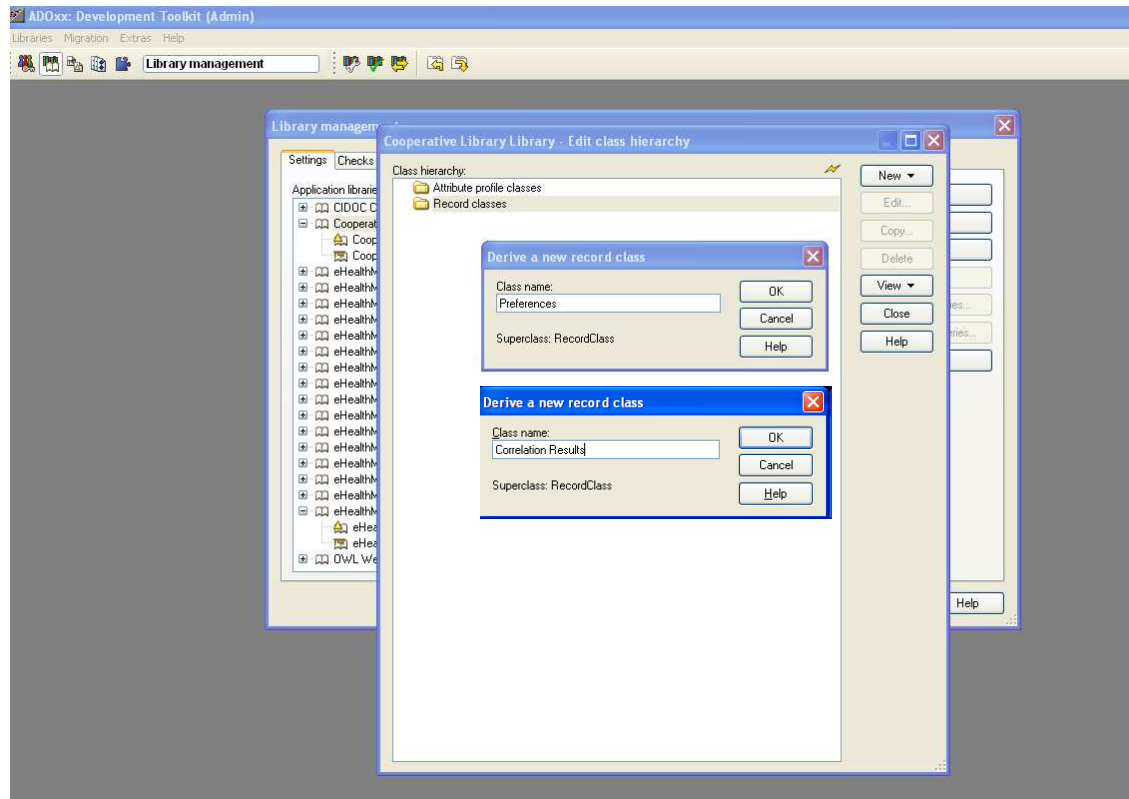
Create New Classes



Create New Classes

- Select “Cooperative Attribute 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:
“Interaction Process”, “Preference Container”, “Roel” are now sub-classes of __D-construct__

Add and Configure Complex Attribute Types



Add Complex Attribute Types

- Select Cooperative Attribute Library, open Class Hierarchy, select Record Classes.
- Make Recordclass "Preferences" and "Correlation Results".
- Add under class "Preferences" attributes "Preference", "Dimension" as type STRING, and "Weight" as type ENUMERATION and value range {-3@-2@-1@0@1@2@3}
- Add under class "Correlation Results" attributes "Preference", "Dimension" as type STRING and "Correlation Results" as type "DOUBLE"
- Configure Attrep Attributes of recordclasses like,

Preferences AttrRep

NOTEBOOK

CHAPTER "Description"

ATTR "Preference" width:5.0 write-protected

ATTR "Dimension" width:5.0 write-protected

ATTR "Weight" width:5.0

Correlation Results AttrRep

NOTEBOOK

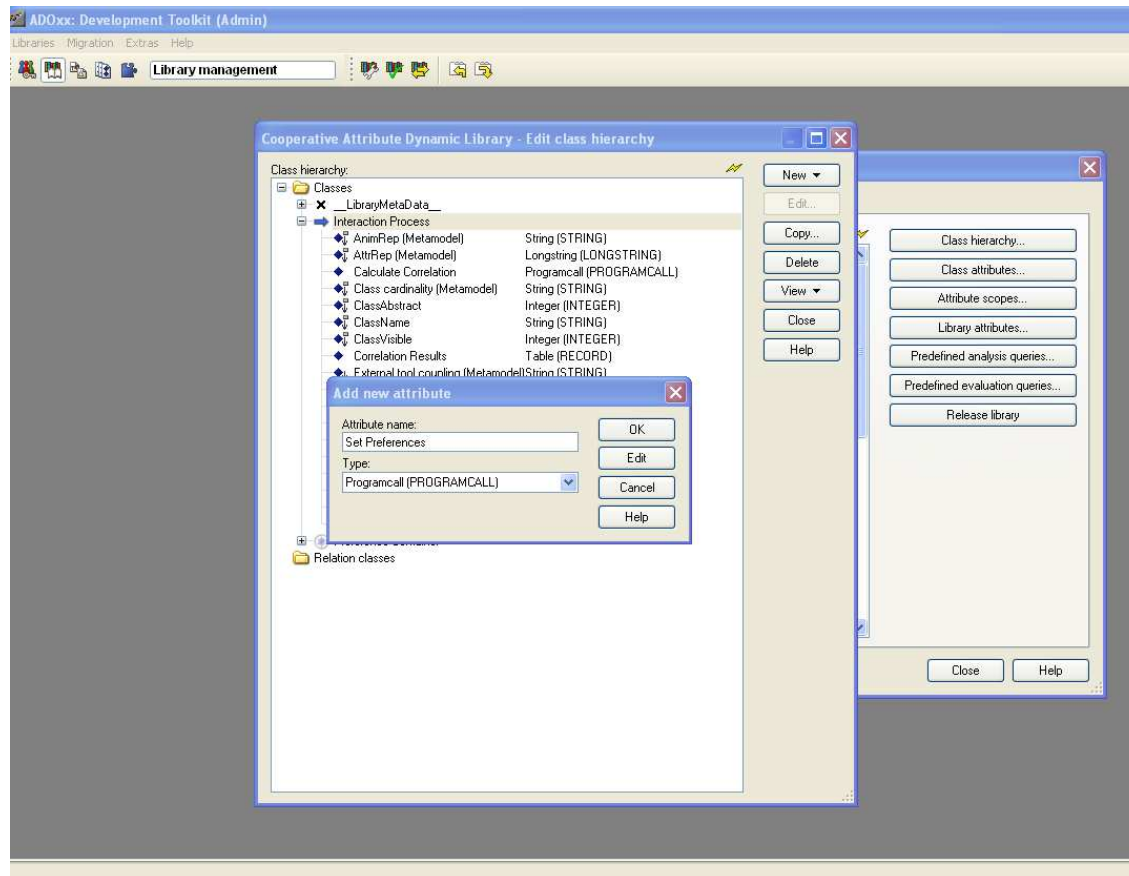
CHAPTER "Description"

ATTR "Preference" width:5.0 write-protected

ATTR "Dimension" width:5.0 write-protected

ATTR "Correlation" width:5.0 write-protected

Add Attributes



Add Attributes

- Select "Interaction Process" and click Newattribute.
- Make "Set Preferences" and "Calculate Correlation" as type PROGRAMCALL, set their Standart values "Set Preferences" and "Calculate Correlation" respectively and configure their EnumerationDomain attributes as given below.
- Make "First User Preferences" and "Second User Preferences" as TABLE and select "Preferences" as Referenced record class
- Make "Correlation Results" as TABLE and select "Correlation Results" as References record class.
- Select "Preference Container" and click New, attribute.
- Make "Dimension" as type ENUMERION with value range {Reliability@Availibility@Cost}.

Set Preferences

ITEM "Set Preferences"

EXECUTE file:("db:\\setPreferenceInInteractionProcess.asc")

Calculate Correlation

ITEM "Calculate Correlation"

EXECUTE file:("db:\\correlationCalculation.asc")

Implement and Import ADOscripts File into Database

setPreferenceInInteractionProcess.asc

```
CC "CoreUI" MODEL_SELECT_BOX modeltype:("Preferences Pool Model") title:("Select Preferences Pool Model")
boxtext:("Please select a Preferences Pool Model") oktext:("Select")
SET n_selected_prefpool_modelid:(modelids)
IF (endbutton = "ok")
{
    CC "Core" LOAD_MODEL modelid:(VAL n_selected_prefpool_modelid)

    CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL n_selected_prefpool_modelid) classname:("Preference
Container")
    #--> RESULT ecode: intValue objids: list .
    SET s_prefcont_objids:(objids)
    SET a_listof_objnames:(array(tokcnt(s_prefcont_objids," "))
    SET a_listof_objdimension:(array(tokcnt(s_prefcont_objids," "))
    SET counter:0

    FOR s_prefcont_objid in: (s_prefcont_objids)
    {
        CC "Core" GET_OBJ_NAME objid:(VAL s_prefcont_objid)
        SET a_listof_objnames[counter]:(objname)

        CC "Core" GET_ATTR_VAL objid:(VAL s_prefcont_objid) attrname:("Dimension")
        SET a_listof_objdimension[counter]:(val)
        SET counter:(counter+1)
    }

    CC "Modeling" GET_SELECTED
    #--> RESULT ecode: intValue objids: strValue classid: id .
    SET s_selected_objid:(objids)
```

...

Implement and Import ADOscripts File into Database

correlationCalculation.asc

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

CC "AQL" EVAL_AQL_EXPRESSION modelid:(n_act_modelid) expr:("<\\"Interaction Process\\">")
SET s_intprocess_objids:(objids)
CC "Core" GET_CLASS_ID classname:("Interaction Process") bp-library
SET n_intprocess_classid:(classid)

CC "AdoScript" PERCWIN_CREATE title:"Please wait! Correlations are being calculated..."
CC "AdoScript" SLEEP ms:100

SET n_percentage:0
SET n_count:0
FOR s_intprocess_objid in:(s_intprocess_objids)
{
    SET n_num_of_objs:(tokcnt(s_intprocess_objids," "))
    SET n_count:(n_count+1)
    SET n_percentage:((n_count/(n_num_of_objs+4))*100)
    CC "AdoScript" PERCWIN_SET percentage:(n_percentage)
    CC "AdoScript" SLEEP ms:200

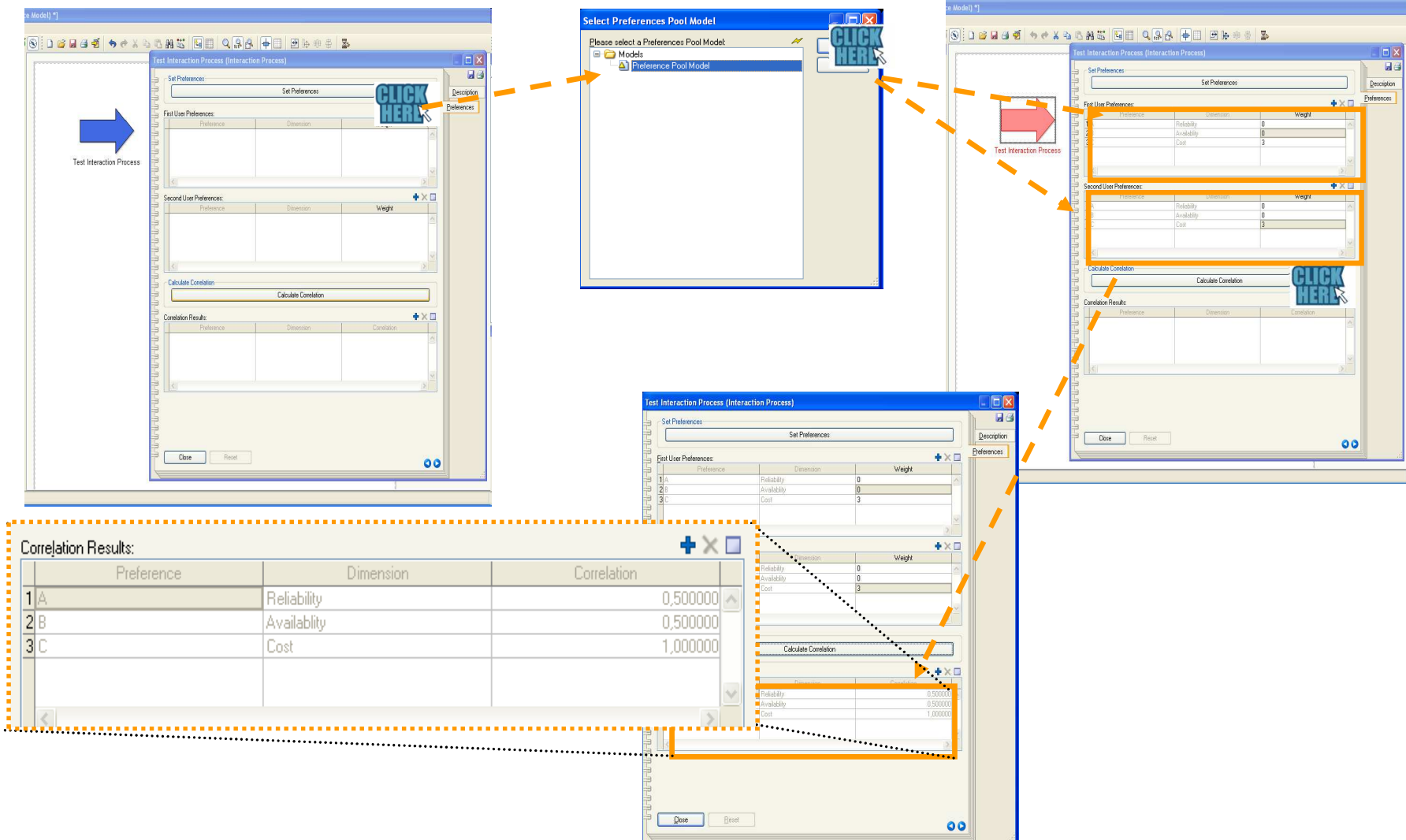
    CC "Core" GET_CLASS_ID classname:("Interaction Process") bp-library
    SET n_intproc_classid:(classid)

    CC "Core" GET_ATTR_ID classid:(n_intproc_classid) attrname:("First User Preferences")
    SET n_intproc_inspacepref_attrid:(attrid)

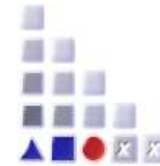
    CC "Core" GET_ATTR_ID classid:(n_intproc_classid) attrname:("Second User Preferences")
    SET n_intproc_outspacepref_attrid:(attrid)

    ...
}
```

Result



Further Questions?



www.adoxx.org

tutorial@adoxx.org

