

Interact with Google Maps

3. SCENARIO: EXTERNAL SERVICE INTERACTION

Scenario Description



Case: An implementation of a modelling method is extended/enhanced by functionality external to the meta-modelling platform through API calls on WebServices (WS).

GOAL:

- Demonstrate usage of APIs in ADOxx to call external services
- Implement mechanisms for push and pull invocation to external services

Interaction Case:

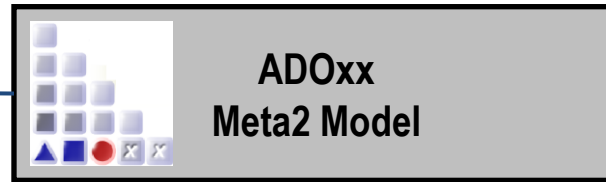
Google Map Interaction: Models defined for the design of supply chain distribution networks are enhanced with geolocation data using the Google Maps WS and OpenStreetMap WS

Meta Modelling Layer: Web-Service Functionality in ADOxx



Component:
Web-Service

provides



developed
in

C++, C#, Java

Specialisation of

execute (AdoScript)

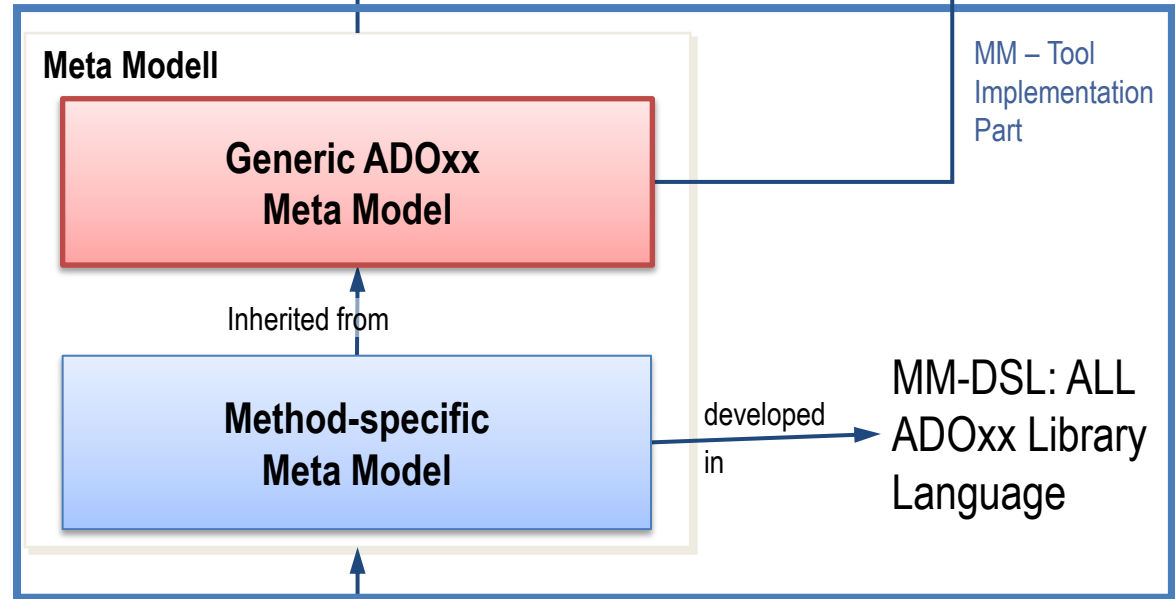
getTOC()
returns:
CC "Documentation" XML_TOC_FOR_USER_ID

getModelImage(String modelid)

getModelXML(String modelid)

getModelImageMap(String modelid)

Instance of



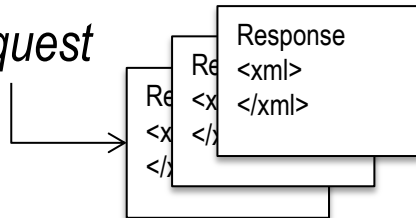
Inherited from

developed
in

MM-DSL: ALL
ADOxx Library
Language

Instance of

Request



Model

described
in

ADL, XML

EVENT HANDLERS



In ADOxx event handlers are used to:

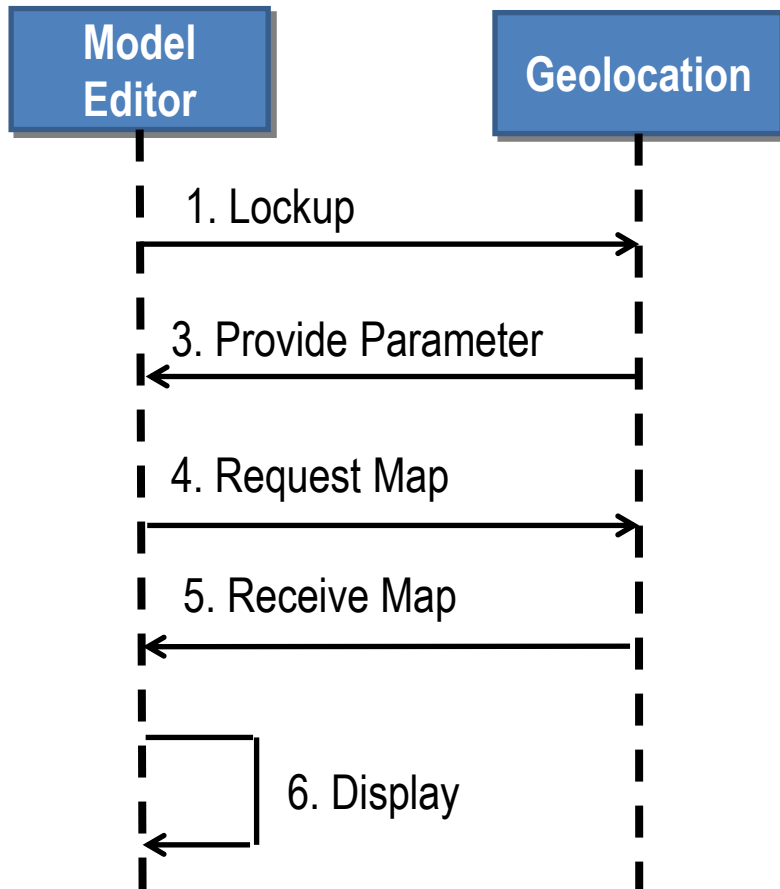
- a) Listen to events that result from the interaction with of the modelling toolkit
- b) Handle/Trigger operations based on the events

Event handlers are realized as an external coupling implementation in the platform, depending on the event, a certain set of parameters/variables are pre-set to be used during the implementation of the actual handler.

Event Category	Number of Events Available
Core	48
Application	3
Modelling	15
Simulation	2
Import/Export	2
Drawing	4

```
# Event implementation to prevent the deletion of instances of a
certain class
ON_EVENT "BeforeDeleteInstance" {
    CC "Core" GET_CLASS_NAME classid:(classid)
    IF (classname = "Information") {
        CC "Core" GET_ATTR_ID classid:(classid) attrname:"Allow
deletion"
        CC "Core" GET_ATTR_VAL objid:(instid) attrid:(attrid)
        IF (val = "no") {
            CC "AdoScript" ERRORBOX "Deletion not allowed!"
            EXIT -1
            # -1 means, that the deletion is aborted, but no error
            # message will appear. That's what we want here, as an
            # error box has already been shown by this event handler.
        }
    }
    # the following statement is redundant (no EXIT means EXIT 0)
    EXIT 0
}
```

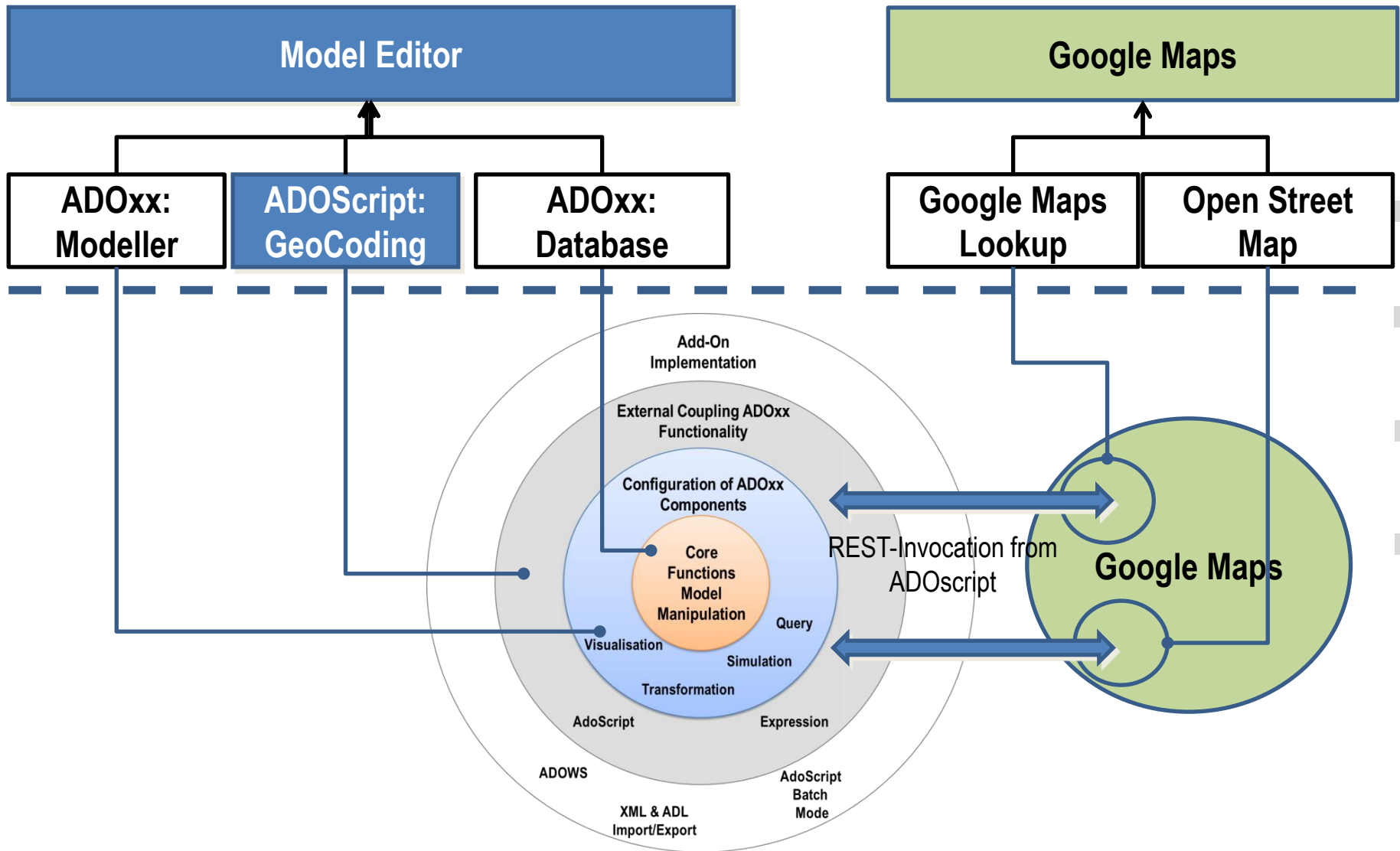
Description of GeoCoding Invocation



Additional Details:

1. Define map (location by name, zoom factor)
2. Request LONG/LAT options by location name through ReST call
3. Select center location from options
4. Request map image through ReST call

Mapping ADOxx Functionality





- **ADOxx Constructs for Modelling Language Extensions**
 - Define a new *attribute of type* “**PROGRAMCALL**” to invoke the map service calls
 - Update the *representation* of the modeltype to represent the map as a background
- **ADOxx Constructs for Mechanism and Algorithms Development**
 - Use AdoScript *External Call Operations to call and invoke* the GoogleMaps/OpenStreetMap API for map information
 - Establish *basic UI elements* for selection of LONG/LAT options of model
 - Define the *event handler* “Applnitialized” also to define the proxy settings.

Pseudo Code: PULL Invocation



```
ITEM Notebook Button "Update map" {
  #preset by button: modelid
  locationName = getModelAttribute('locationname')
  mapZoom = getModelAttribute('zoom')
  List locations = CallAPIGeoLookUpLocation (locationName )
  locationSelectionBox = buildListBox (parse(locations))
  locationSelectionBox.show(modal)
  If (endbutton = cancel) {
    EXIT
  }
  ELSE {
    File map = CallAPIGeoStaticMapService(selectedLonLat)
    setModelAttribute ('mapfile', map)
    triggerModelRepresentationUpdate(modelid)
  }
}
```

...Core Operations
...Invocation Operations
...Basic UI Operations

Implementation Result: Maps in Modelling Editor



The screenshot shows the Modelling Editor interface with a map of Europe. Several callout boxes are overlaid on the map, each describing a step in the implementation process:

- Import map into ADOxx**: Located in the top left, pointing to the map area.
- Auto-connect nodes**: Located in the top center, pointing to the map area.
- Calculate weather, altitude**: Located in the top right, pointing to the map area.
- Calculate routes**: Located in the middle right, pointing to the map area.
- Mass-add nodes**: Located in the bottom right, pointing to the map area.
- Calculate distance matrix**: Located in the bottom center, pointing to the map area.
- Find position from long/lat**: Located in the bottom left, pointing to the map area.
- Find position from ADOxx nodes**: Located in the middle left, pointing to the map area.

The map shows several cities marked with blue boxes and connected by lines, representing a network. The cities include Madrid, Barcelona, Valencia, Rome, Venedig, and Vienna. The interface also shows a menu bar (Model, Edit, View, Geolocation tools, Extras, Window, Help), a toolbar, and an Explorer window on the left showing 'Models' and 'My Map'.

Used ADOxx Functionality: API / Web-Service Invocation



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

ADOscript Batch Mode





HANDS-ON

Interact with Google Maps

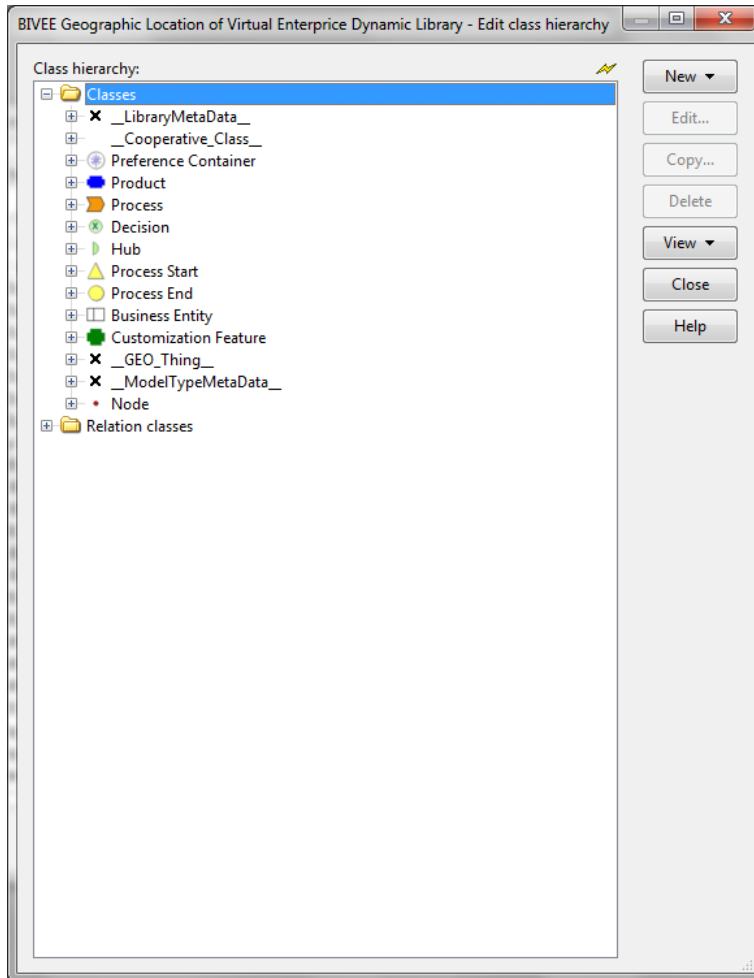
3. SCENARIO: EXTERNAL SERVICE INTERACTION



Prerequisites for SMS service

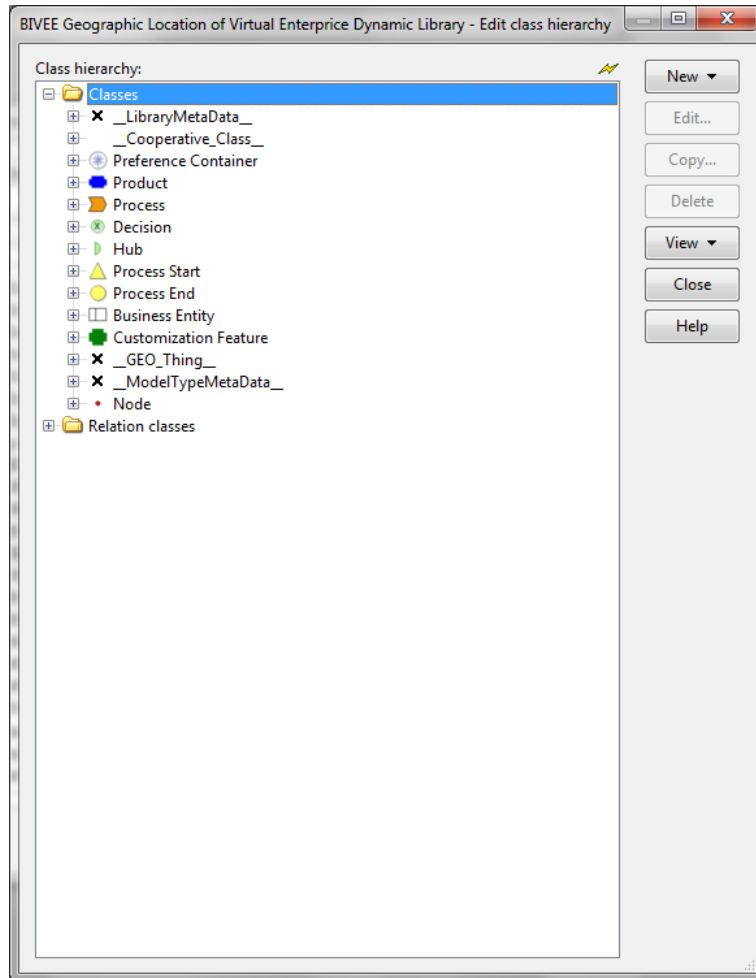
- ADOxx v 1.3
download at www.adoxx.org
- GnuWin32 Wget
download at <http://downloads.sourceforge.net/gnuwin32/wget-1.11.4-1-setup.exe>
- Availability of geolocation web services, services used:
OpenStreetMap
Google Maps API
WeatherOnline

Definition of Modelling Language on ADOxx



1. Define ATTRIBUTE in “__D-construct__”
 - GUID
2. Create “__GEO_Thing__” : “__D-construct__”
 - Absolute X ; Absolute X (Map)
 - Absolute Y ; Absolute Y (Map)
 - ADOxx X ; ADOxx X (Map)
 - ADOxx Y ; ADOxx Y (Map)
 - Altitude ; Altitude data
 - Cloudcover
 - GeoLocation Data
 - Humidity
 - Latitude ; Latitude (Map)
 - Location display name
 - Location name
 - Longitude ; Longitude (Map)
 - Observation time
 - Out of bounds
 - Precipitation
 - Pressure
 - Relative X ; Relative X (Map)
 - Relative Y ; Relative Y (Map)
 - Reposition on map
 - Temperature
 - Tile X ; Tile X (Map)
 - Tile Y ; Tile Y (Map)
 - Update altitude
 - Update node position
 - Update position from map
 - Weather code
 - Weather data

Definition of Modelling Language on ADOxx



- Weather description
- Weather forecast
- Weather icon URL
- Wind direction (16point)
- Wind direction (Degree)
- Windspeed (Kilometer/h)
- Windspeed (Miles)

3. Create “__ModelTypeMetaData__” : “__LibraryMetaData__”

Define ATTRIBUTES

- "State",
- "Map center", "Map zoom", "Map width", "Map height"
- "Update map"
- "Map data"
- "Display name"
- "Map longitude"
- "Map latitude"
- "GeoLocation Data", "GeoLocation URL"
- "Map URL"
- "Tile X"
- "Tile Y"
- "Distance matrix"
- "Distance matrix nodeset"
- "Distance matrix URL"
- "Update distance matrix nodeset"
- "Distance matrix (JSON)"
- "Download distance matrix"

4. Create “Business Entity” : “__GEO_Thing__”

ADD NEW ATTRIBUTE TO NOTEBOOK OF MODELTYPE



Class hierarchy:

- Classes
 - LibraryMetaData__
 - Cooperative_Class__
 - Preference Container
 - Product
 - Process
 - Decision
 - Hub
 - Process Start
 - Process End
 - Customization Feature
 - GEO_Thing__
 - ModelTypeMetaData__
 - APListChangeCounter__ Integer (INTEGER)
 - ModellistChangeCounter__ Integer (INTEGER)
 - AnimRep (Metamodel) String (STRING)
 - AttrRep (Metamodel) Longstring (LONGSTRING)
 - Class cardinality (Metamodel) String (STRING)
 - ClassAbstract Integer (INTEGER)
 - ClassName String (STRING)
 - homeid String (STRING)
 - Map center String (STRING)
 - Map height Integer (INTEGER)
 - Map latitude String (STRING)
 - Map longitude String (STRING)
 - Map URL Expression (EXPRESSION)
 - Map width Integer (INTEGER)
 - Map zoom Enumeration (ENUMERATION)
 - Model pointer (Metamodel) String (STRING)
 - Position (Metamodel) String (STRING)
 - Supply Chain Location Attrrep Longstring (LONGSTRING)
 - Supply Chain Location Graphrep Longstring (LONGSTRING)
 - TESTDOCSETTING String (STRING)
 - Tile X Expression (EXPRESSION)
 - Tile Y Expression (EXPRESSION)
 - Update distance matrix nodeset Programcall (PROGRAMCALL)
 - Update map Programcall (PROGRAMCALL)
 - VisibleAttrs (Metamodel) String (STRING)
 - WF_Trans (Metamodel) String (STRING)
- Node
- Business Entity
- Relation classes

1 Select class attribute „Supply Chain Location Attrrep“ of class " __ModelTypeMetaData__ "

Click on Edit ...

- New
- Edit...**
- Copy...
- Delete
- View

Supply Chain Location Attrrep - Edit facets

Standard value:

```
NOTEBOOK
CHAPTER "Map configuration"
ATTR "State"
ATTR "Map center"
ATTR "Map zoom"
ATTR "Map width"
ATTR "Map height"
ATTR "Update map" no-param push-button
CHAPTER "Map data"
ATTR "Display name" write-protected
ATTR "Map longitude" write-protected
ATTR "Map latitude" write-protected
ATTR "GeoLocation Data" write-protected
ATTR "GeoLocation URL" write-protected
ATTR "Map URL" write-protected
```

Attribute type:
Longstring (LONGSTRING)

Predefined value
Facets

2

Add attribute to notebook by adding the line
ATTR „State“
ATTR „Map center“
...
in chapter „Map configuration“

Confirm with Close to save the change.

Close

UPDATE OF MODEL GRAPHREP



1

Select class attribute „Supply Chain Location GraphRep“ of class “__ModelTypeMetaData__”

Click on Edit ...

2

GRAPHREP
AVAL filename:"GeoLocation URL"
BITMAPINFO (filename)
BITMAP (filename) x:0cm y:0cm w:(bmpwidth / 96 * 2.54cm) h:(bmpheight / 96 * 2.54cm)

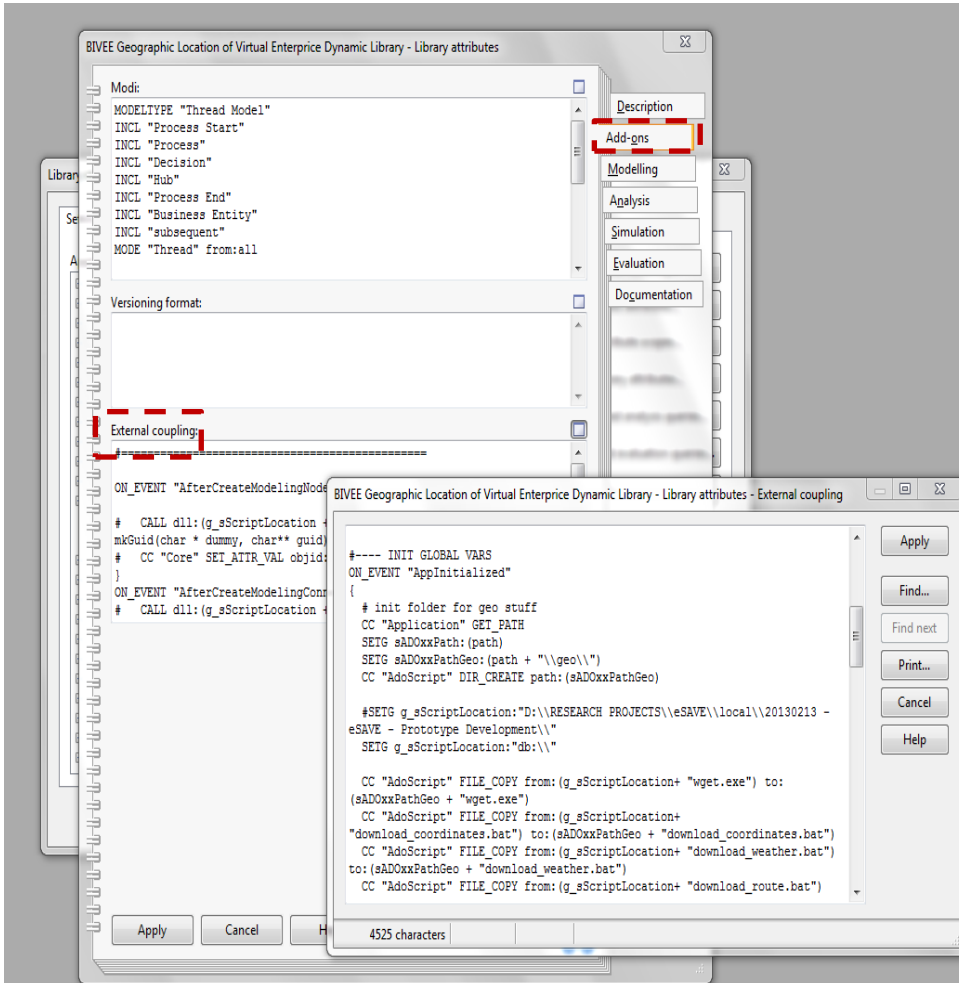
UPDATE MODELTYPE



MODELTYPE "Scope Model" from:none plural:"Supply Chain Location Networks" pos:1 attrrep:"Supply Chain Location Attrep" graphrep:"Supply Chain Location Graphrep"
INCL "Business Entity"
INCL "connects"

```
ON_EVENT "AfterCreateModelingNode" {  
# CALL dll:(g_sScriptLocation + "MkGuidDll.dll") function:"long  
mkGuid(char * dummy, char** guid)"  
# CC "Core" SET_ATTR_VAL objid:(objid) attrname:"GUID" val:(guid)  
}  
ON_EVENT "AfterCreateModelingConnector" {  
# CALL dll:(g_sScriptLocation + "MkGuidDll.dll") function:"long"
```

Update External Coupling



```

ON_EVENT "AppInitialized"
{
    # init folder for geo stuff
    CC "Application" GET_PATH
    SETG sADOxxPath:(path)
    SETG sADOxxPathGeo:(path + "\\geo\\")
    CC "AdoScript" DIR_CREATE path:(sADOxxPathGeo)

    #SETG g_sScriptLocation:"D:\\RESEARCH
    PROJECTS\\eSAVE\\local\\20130213 - eSAVE - Prototype
    Development\\"
    SETG g_sScriptLocation:"db\\"

    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+ "wget.exe")
    to:(sADOxxPathGeo + "wget.exe")
    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+
    "download_coordinates.bat") to:(sADOxxPathGeo +
    "download_coordinates.bat")
    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+
    "download_weather.bat") to:(sADOxxPathGeo +
    "download_weather.bat")
    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+
    "download_route.bat") to:(sADOxxPathGeo + "download_route.bat")
    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+
    "download_matrix.bat") to:(sADOxxPathGeo + "download_matrix.bat")
    CC "AdoScript" FILE_COPY from:(g_sScriptLocation+
    "download_map.bat") to:(sADOxxPathGeo + "download_map.bat")
    ...
}
    
```

AdoScript Implementation update_map.asc



```
CC "AdoScript" SET_CWD path:(sADOxxPathGeo)
CC "Modeling" GET_ACT_MODEL
SETL nCurrentModelID:(modelid)

CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map center")
SETL nMapCenterAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map longitude")
SETL nMapLongAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map latitude")
SETL nMapLatAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Display name")
SETL nDisplayNameAttrID:(attrid)

CC "Core" GET_ATTR_ID classid:bp-model attrname:("GeoLocation Data")
SETL nGeoLocationDataAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("GeoLocation URL")
SETL nGeoLocationURLAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map width")
SETL nMapWidthAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map height")
SETL nMapHeightAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map URL")
SETL nMapURLAttrID:(attrid)

CC "Core" GET_ATTR_VAL objid:(nCurrentModelID) attrid:(nMapCenterAttrID)
SETL sMapCenterValue:(val)
...
```

AdoScript Implementation update_position_node.asc



```
CC "AdoScript" SET_CWD path:(sADOxxPathGeo)
# AdoScript to position the marker on the map
SETL nMarkerObjID:(nCurrentObjID)
CC "Modeling" GET_ACT_MODEL
SETL nCurrentModelID:(modelid)

CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map width")
SETL nMapWidthAttrID:(attrid)
CC "Core" GET_ATTR_ID classid:bp-model attrname:("Map height")
SETL nMapHeightAttrID:(attrid)
CC "Core" GET_ATTR_VAL objid:(nMarkerObjID) attrname:("Location name")
CC "Core" SET_ATTR_VAL objid:(nMarkerObjID) attrname:("Out of bounds") val:(0)
CC "Modeling" UNDYE (nMarkerObjID)
SETL sLocationName:(val)
IF (LEN sLocationName <> 0) {
  CC "AdoScript" MSGWIN ("Updating location information for marker: " + sLocationName)
  CC "Core" GET_ATTR_VAL objid:(nMarkerObjID) attrname:("ADOxx X")
  SETL nADOxxX:(val)
  CC "Core" GET_ATTR_VAL objid:(nMarkerObjID) attrname:("ADOxx Y")
  SETL nADOxxY:(val)
  CC "Core" GET_ATTR_VAL objid:(nCurrentModelID) attrid:(nMapHeightAttrID)
  SETL nMapHeight:(val/ 96 * 2.54)
  CC "Core" GET_ATTR_VAL objid:(nCurrentModelID) attrid:(nMapWidthAttrID)
  SETL nMapWidth:(val/ 96 * 2.54)
  IF (nADOxxX > 0 AND nADOxxX < nMapHeight AND nADOxxY > 0 AND nADOxxY < nMapHeight) {
    CC "Modeling" SET_OBJ_POS objid:(nMarkerObjID) x:(CM nADOxxX) y:(CM nADOxxY)
  }
  ...
}
```



Further Questions?



www.adoxx.org

tutorial@adoxx.org

