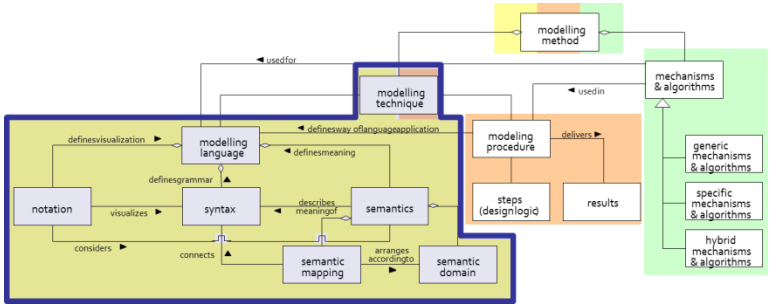


MODELLING METHOD INTEGRATION:

SEMANTIC LIFTING APPROACH

SCENARIO: SUPPLY CHAIN MANAGEMENT

Scenario Description

Case:

Integration of Modelling Methods with Semantic Lifting in Supply Chain Management

GOAL:

Realization of Possible Semantic Lifting Solutions with Extending Modelling Language or with External Coupling

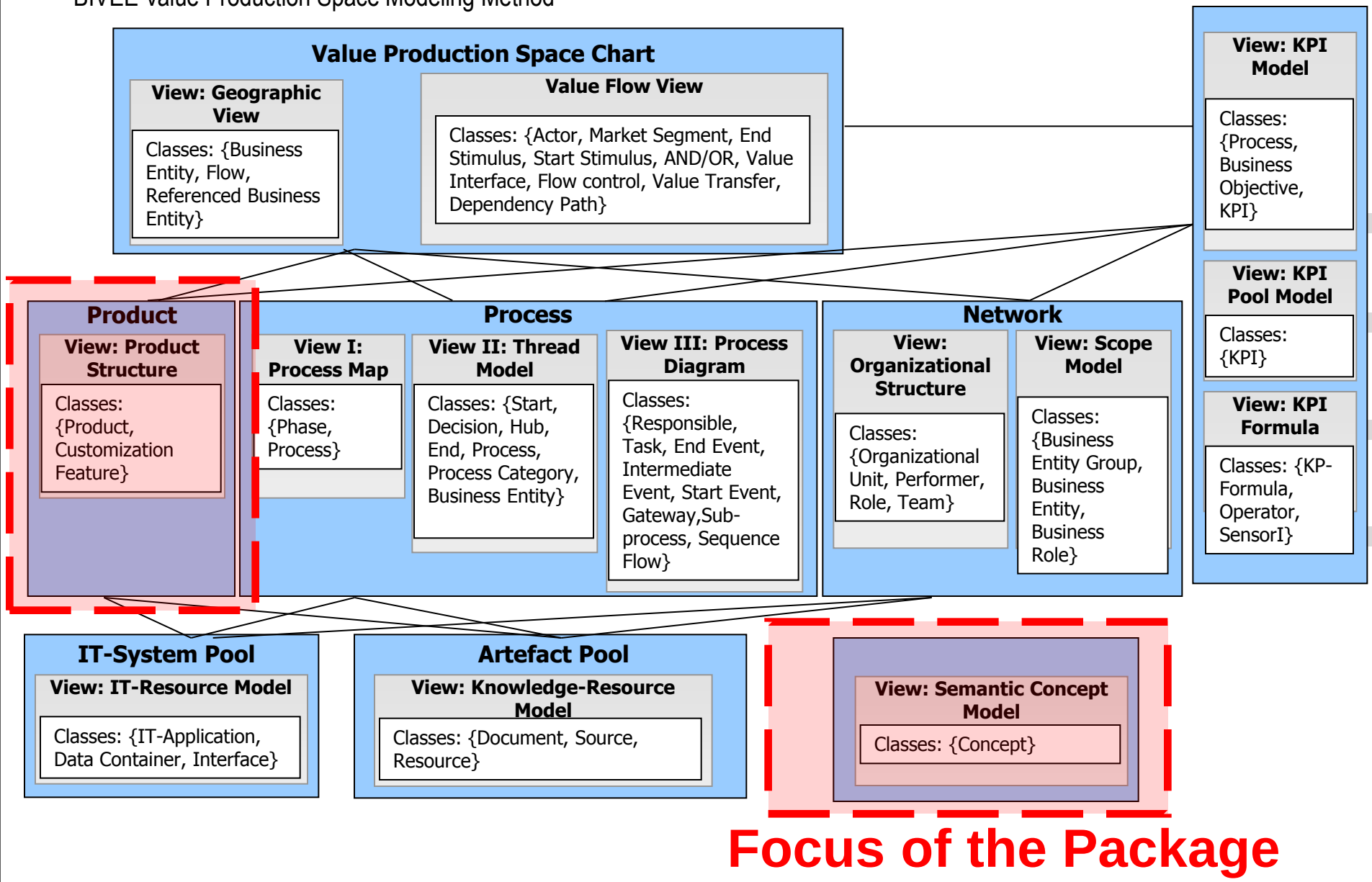
Supply Chain Management Scenario

Characteristics of SCM Models

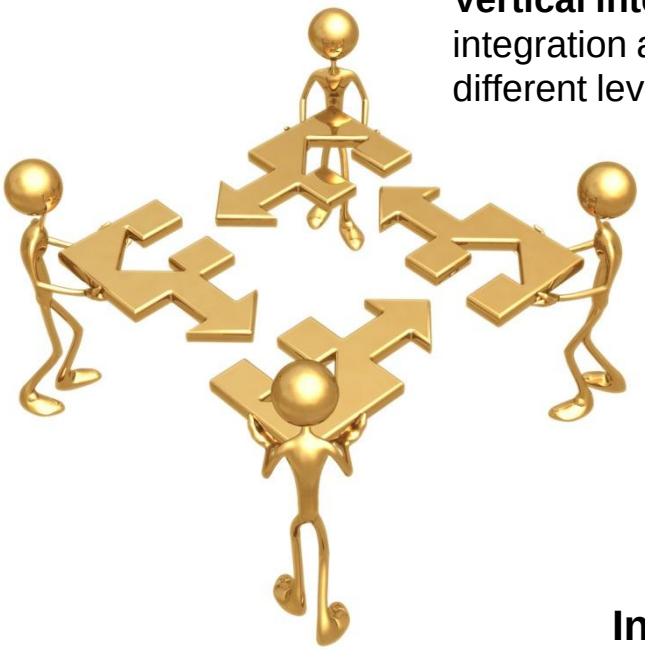
- Interplay of different actors:
  - directly involved (e.g. producers, suppliers)
  - semi-directly involved (e.g. customers)
  - indirectly involved (e.g. government)
- Obeying to different rules:
  - imposed rules (e.g. law – privacy issues)
  - “lived” rules (e.g. management decisions)
- Importance of Interoperability
  - many different standards for all aspects (e.g. SCOR, VRM)
  - importance of “leveled” semantics (e.g. understanding current status)

Supply Chain Management Scenario

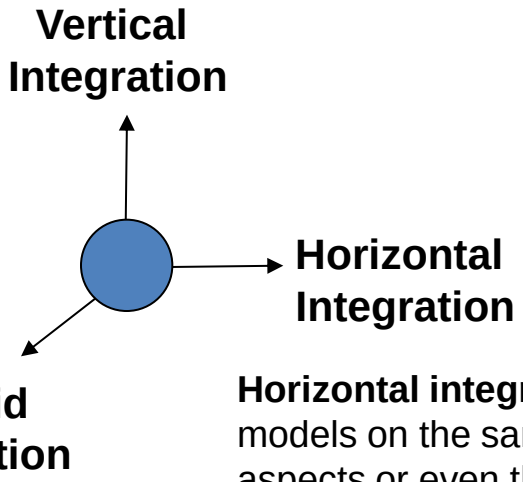
BIVEE Value Production Space Modeling Method



Hybrid Integration Approaches



Vertical integration represents a top-down or bottom-up integration approach. Metamodels and models with different level of details are integrated.



Horizontal integration integrates enterprise models on the same level of details. Different aspects or even the same aspects but from different viewpoints are integrated.

Hybrid integration is the combination of the vertical and horizontal integration approach.

Source: Kühn H. (2004). Methodenintegration im Business Engineering, PhD Thesis, University of Vienna, April 2004.

Heterogeneity Dimensions

- **Syntactical**
  - Heterogeneity of formats
  - Unstructured, semi-structured and structured formats
  - Example: Serialization formats: XML-based, object-oriented, relational formats, etc.

- **Structural**
  - Representational heterogeneity (different modelling primitives available – expressive power of language may vary) - Example: Some languages support multiple inheritance, others are restricted to single parent class.
  - Schematic heterogeneity (concepts are described differently, while seen from different viewpoints) - Example: the concept “Performer” is defined as a simple attribute of a class, whereas the same concept can be described with two classes: the “Worker” and its generalisation “Processor”.”

- **Semantic**
  - differences in the meaning of the concepts under consideration
  - Semantically equivalent concepts (e.g. synonyms)
  - Semantically related concepts (relation types such as “is-a”, “has-a”, “type-of” and “associate” etc)
  - Unrelated concepts (completely orthogonal semantics – homonyms?)

Heterogeneity dimensions occur on all level of metamodelling hierarchy

Source: Kühn H. (2004). Methodenintegration im Business Engineering, PhD Thesis, University of Vienna, April 2004.

Semantic Lifting Realization(1/2)

- **RS1 – Non-supported direct linkage**: in this scenario the meta models that are utilized in the applied setting are not changed or adapted. The stakeholder responsible for performing the semantic lifting achieves the desired result by manually entering the semantic concept in an existing attribute field in the target meta model In case such attribute is not required a minor change of the target meta model may be required. Functional support of the underlying meta modeling platform for the semantic lifting process is not essential.

- **RS2 – Supported direct linkage:** this realization scenario depicts a use case where two meta models – Source, and Target (possibly residing in two different tools/components of the distributed application) are connected using the scripting functionality provided by the underlying meta modeling platform. The linkage is established by selecting a modeling object from the source meta model and establishing a connection to the target meta model through so-called semantic-tunnel. Additional context information may be made available to the process by employing attributes in either hidden or non-instantiated modeling objects. This scenario relies heavily on the functionality provided by the utilized meta modeling platform.

Semantic Lifting Realization(2/2)

- **RS3 – Indirect linkage:** this scenario makes use of so-called Semantic Transit Models (STM). This depicts an approach where objects from source model required for the semantic lifting are introduced in the target platform as STM – thus enabling light version of the strong integration pattern. This gives advantage of a redundant concept storage – which is then managed by e.g. write protecting the STM and allowing only import of new concepts from the source meta model. The complexity introduced by the redundant storage (subset applicable to the specific model that is to be lifted) is handled either by mechanisms of the meta modeling platform or by external tools and services (e.g. notification on changes of the source/target to keep the semantic lifting consistent)

- **RS4 – Loose coupling:** is a specialization of the indirect linkage, given that the referenced are not the modeling objects from target (or vice versa the source) but instead a reference from both is made to a reference ontology. A similar approach is applied when using pivot approach when integrating domain ontologies.

- **RS5 – Direct/Indirect linkage:** is a combination of the two approaches by providing a support in terms of a fixed core set of concepts that can be used for the semantic lifting, as well as allowing the agile approach to allow import of new or changed concepts in the applied set.

Semantic Lifting Solutions

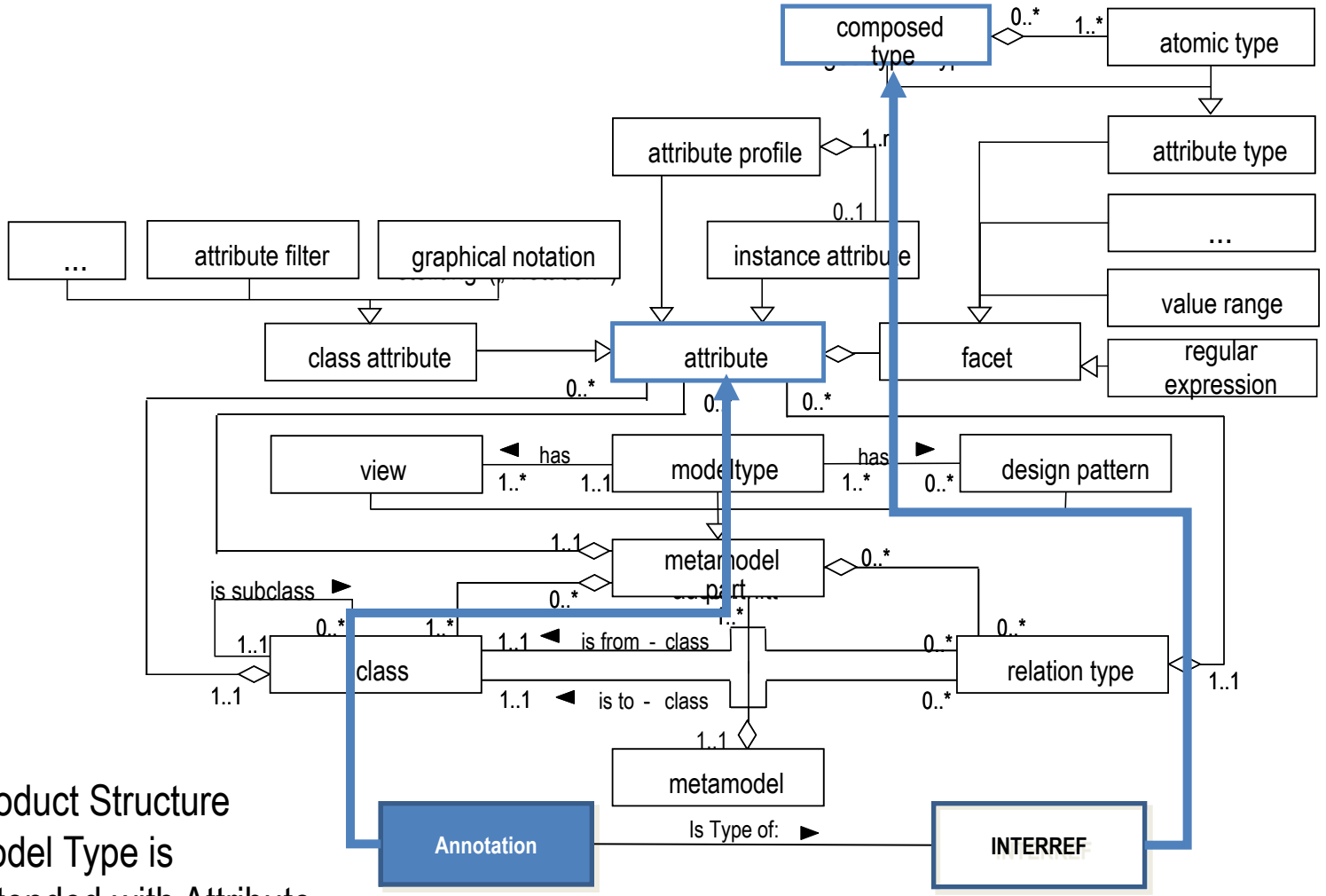
We classify the solutions into two categories based on their realization approach:

- Through the Extending Modeling Language
  - SL1 - Model to Model Instance with Interref
  - SL2 - Textual Annotation within Model

- Through Mechanisms & Algorithms
  - SL3 - Model to Model Instance through the Semantic Transit Model
  - SL4 - Model to Text File with AdoScript
  - SL5 - Model to Web Service with AdoScript
  - SL6 - Graphical Annotation within Model (Whiteboard)
  - SL7 - Model to Web Service through the Semantic Transit Model

SL1 - Model to Model Instance with Interref

Extending the Modelling Method with inheriting required concepts from ADOxx Meta²model:



Product Structure Model Type is extended with Attribute "Annotation"

Extension of: Kühn et al. (1999a), S. 79

Provided Functionality of Metamodelling Platform

Used meta-modelling functionality :

- **Meta²Model:** MODELTYPE, GRAPHREP, ATTREP, ATTRIBUTE TYPE, CLASS

- **ADOxx Meta2Model Component:**
  - Model Editor incl. Notebook
  - Intermodel Reference (INTERREF)
  - Database

ADOxx Realisation HANDS-ON

1. Defining **MODELTYPES**
2. Inheriting **CLASSES** from ADOxx Meta Model
3. Implementing **GRAPHREP**
4. Defining **ATTRIBUTES** and **ATTREP**

Used ADOxx Functionality

- Introduction
- Setup of Implementation Environment

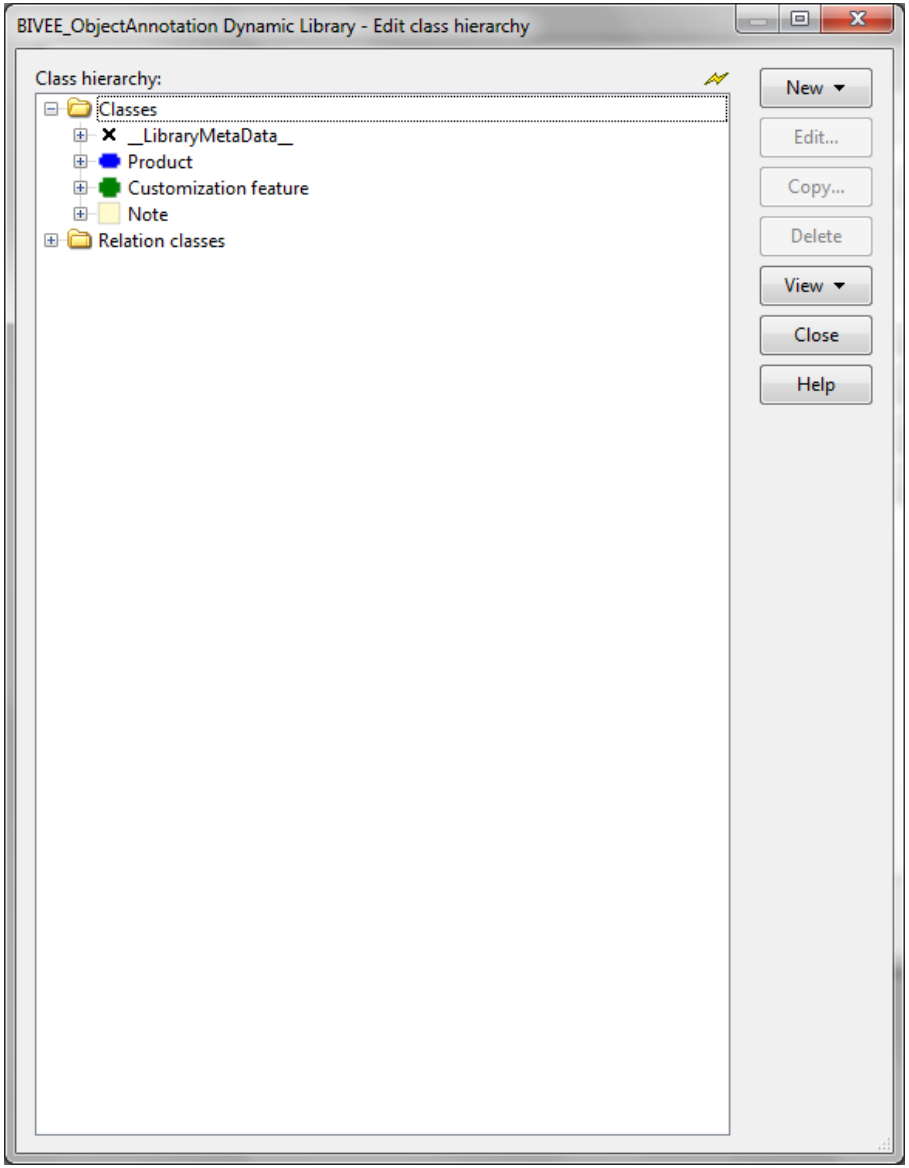
- Modelling Language Implementation
  - **Classes**
  - **Relations**
  - **Class Attributes and Attributes**
    - **GRAPHREP**
    - **ATTRREP**
    - CLASS Cardinality
    - CONVERSION
    - Model Pointer
  - **Attribute Facets**
  - **Model Types**

- 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

SL1 - Model to Model Instance with Interref

HANDS ON: SL1 - Model to Model Instance with InterRef



1. Create Class
   - Product
   - Customization Feature
   - Note

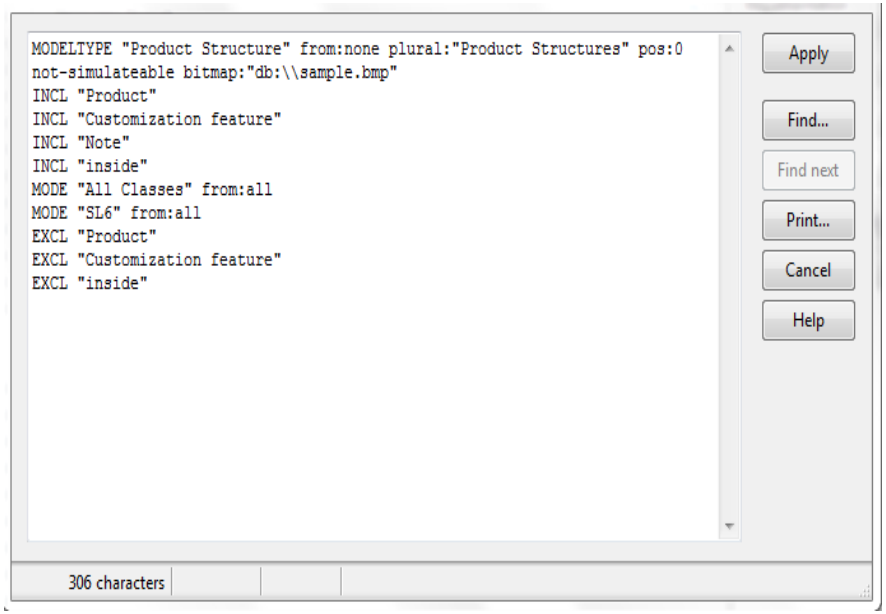
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

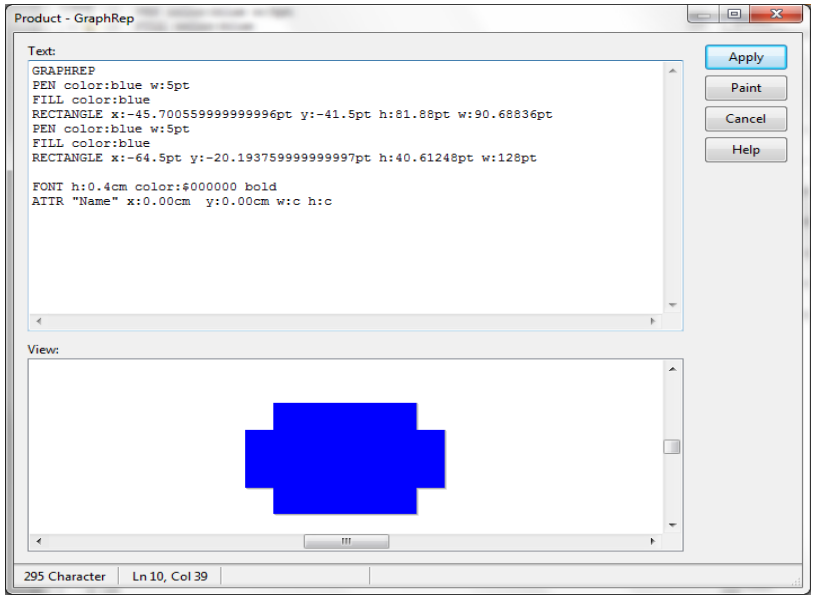
3. Define Atttributes:
   - <Customization Feature>:
     - "Annotation" (INTERREF)
       (see slide **SL1 - Model to Model Instance with InterRef: Code**)
     - "Type" (ENUMERATION)

4. Define AttrRep for class "Customization Feate":

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Select Concept"
ATTR "Type"
```

HANDS ON: SL1 - Model to Model Instance with InterRef





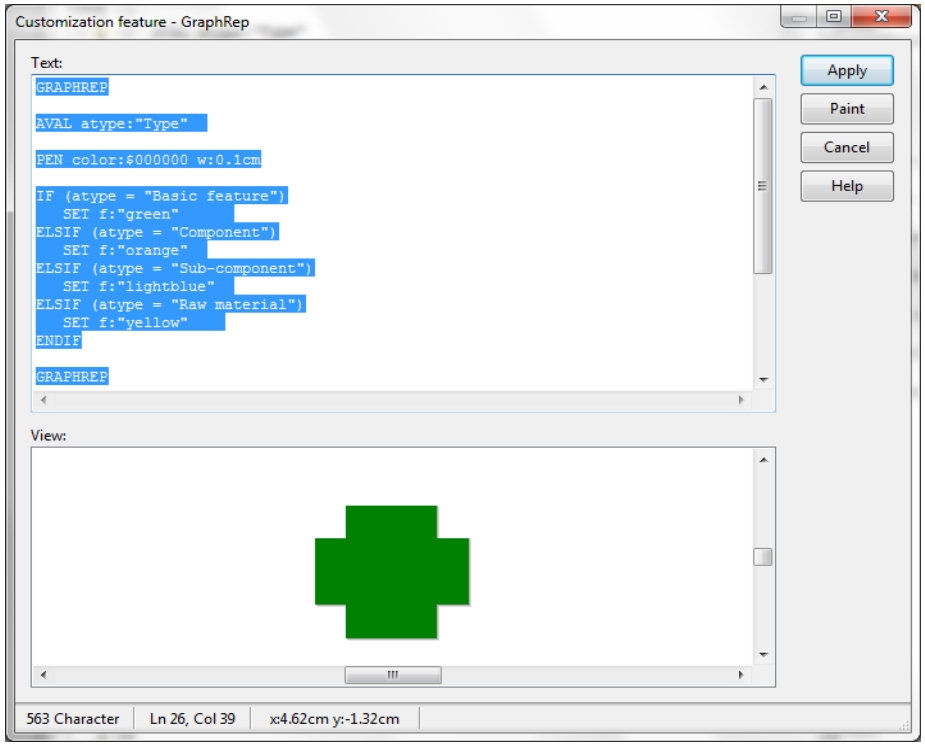
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

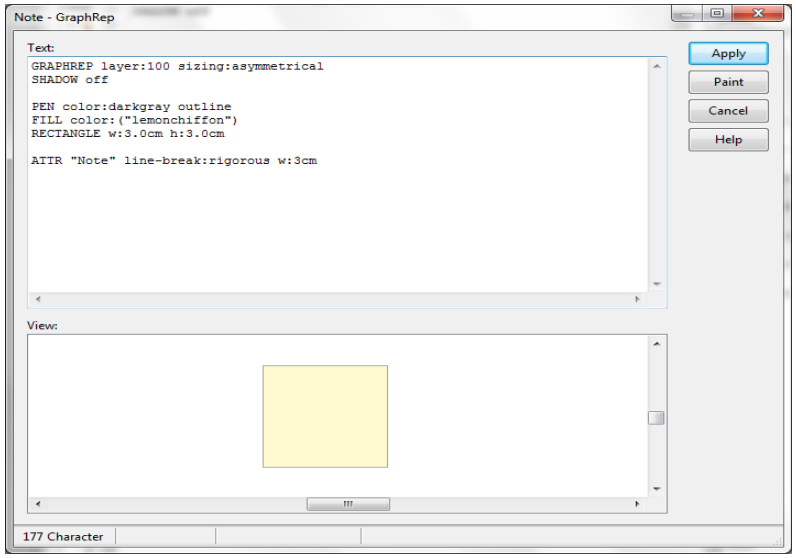
HANDS ON: SL1 - Model to Model Instance with InterRef



7. Define GraphRep for “Customization Feautre”:

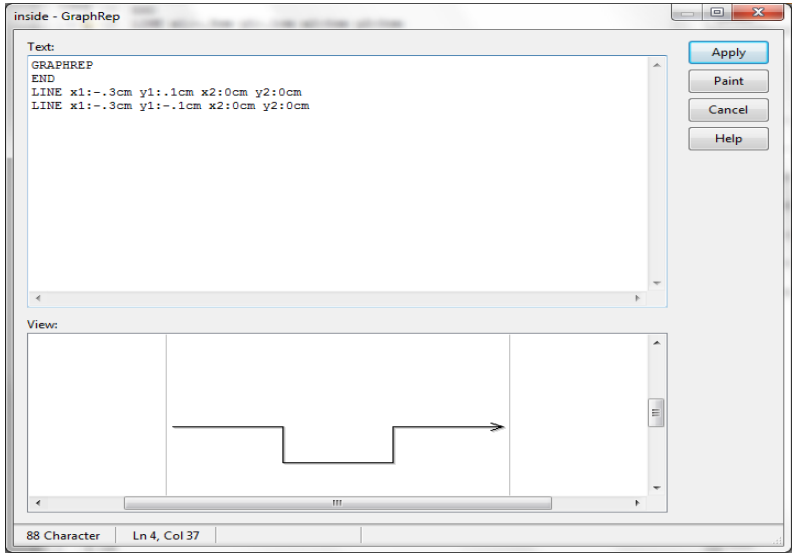
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL1 - Model to Model Instance with InterRef



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

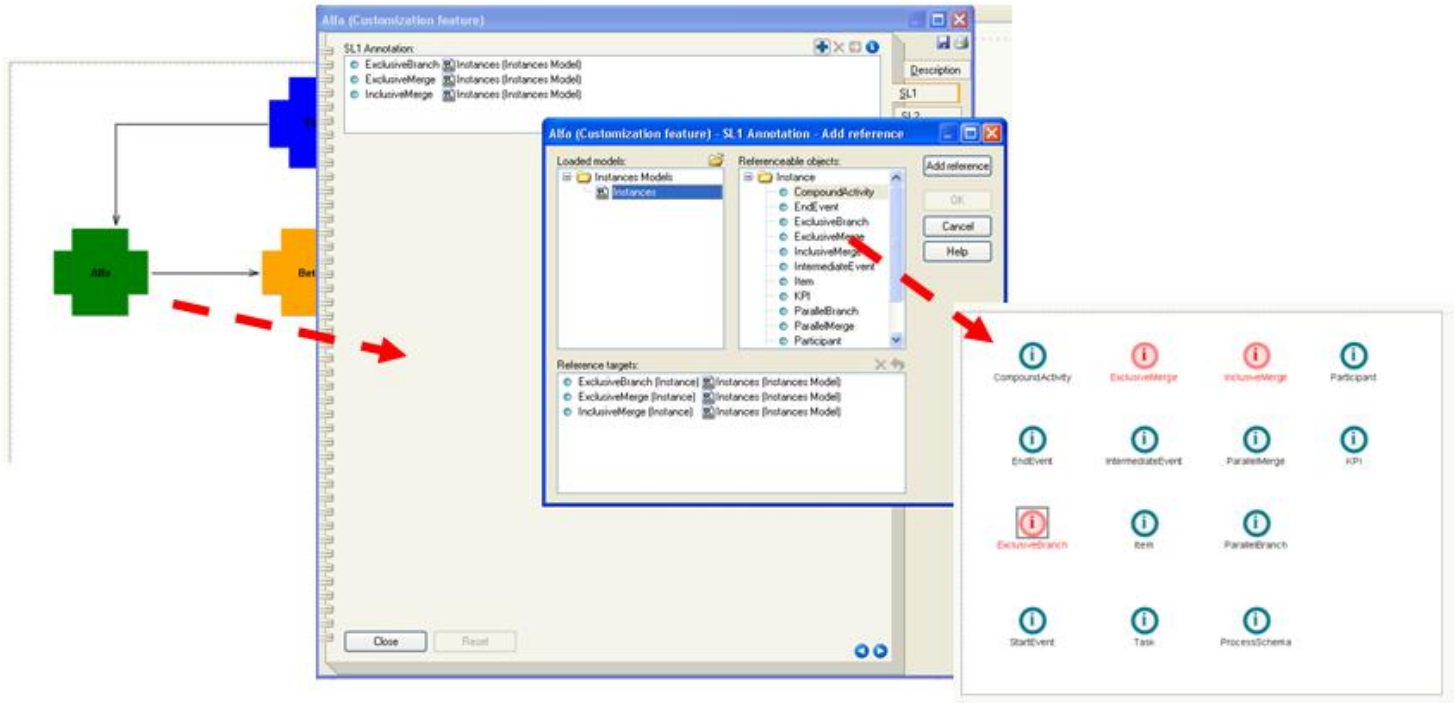
SL1 - Model to Model Instance with InterRef: Code

```
ATTRIBUTE <Annotation>
TYPE INTERREF
           FACET <MultiLineString>
           VALUE 0
           FACET <AttributeHelpText>
           VALUE "This SL approach is realized by providing an INTERREF connection to a one or
more concepts/instances"
           FACET <AttributeRegularExpression>
           VALUE ""
           FACET <AttributeInterRefDomain>
           VALUE "REFDOMAIN

OBJREF
     mt:"Instances Model"
     c:"Instance"
"
```

Result:

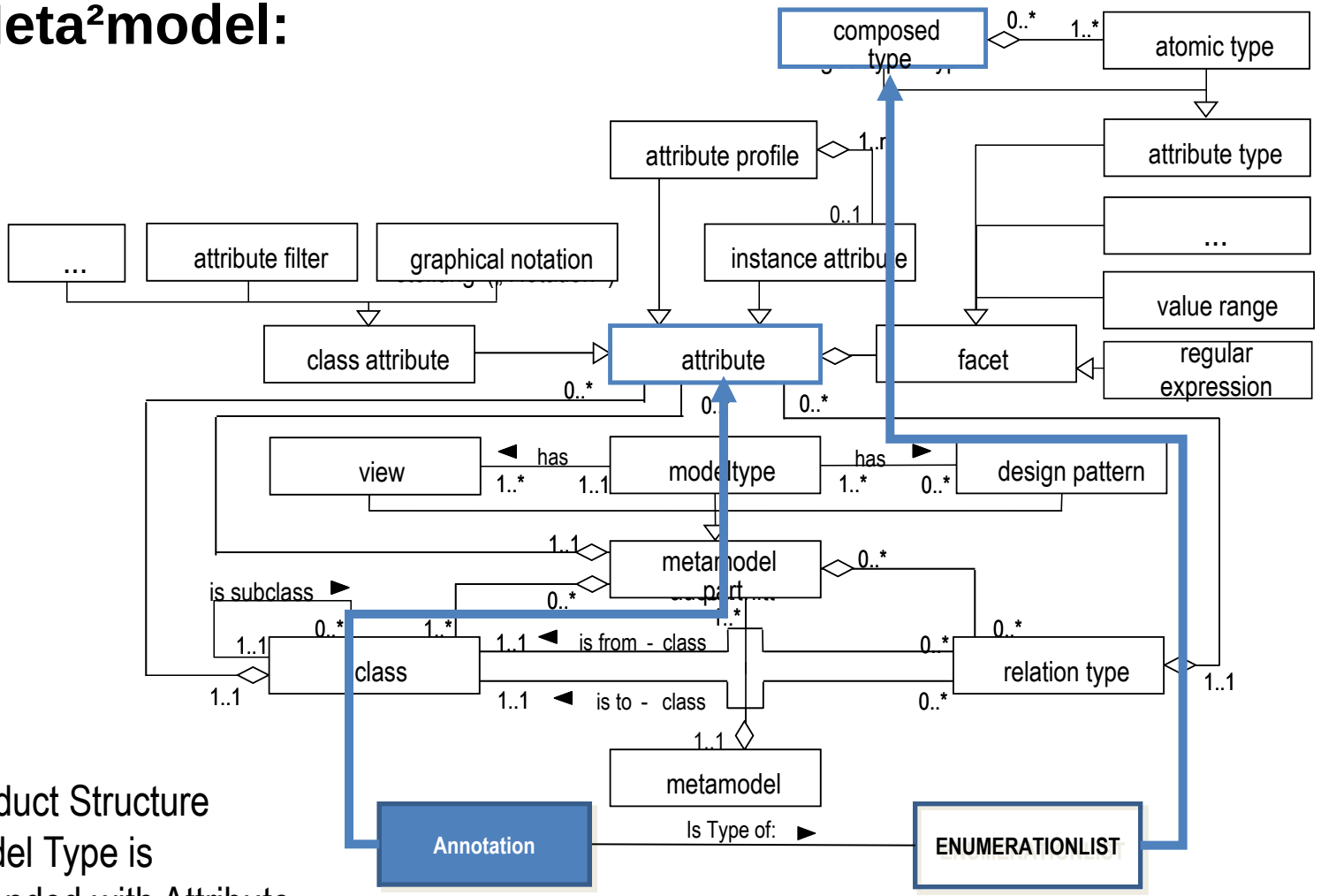
SL1 - Model to Model Instance with Interref: Scenario

This scenario provides a functionality to annotate desired modelling object with the annotation concept instance available in an accessible model by using the INTERREF relation.



SL2 - Textual Annotation within Model

Development of the Modelling Method with inheriting required concepts from ADOxx Meta²model:



Product Structure Model Type is extended with Attribute "Annotation"

Extension of: Kühn et al. (1999a), S. 79

Provided Functionality of Metamodelling Platform

Used meta-modelling functionality :

- **Meta²Model:** MODELTYPE, GRAPHREP, ATTREP, ATTRIBUTE TYPE, CLASS

- **ADOxx Meta2Model Component:**
  - Model Editor incl. Notebook
  - ENUMERATIONLIST
  - Database

HANDS ON: SL2 - Textual Annotation within Model

1. Defining **MODELTYPES**
2. Inheriting **CLASSES** from ADOxx Meta Model
3. Implementing **GRAPHREP**
4. Inherit **RELATIONCLASSES** from ADOxx Meta Model
5. Defining **ATTRIBUTES** and **ATTREP**

Used ADOxx Functionality

- Introduction
- Setup of Implementation Environment

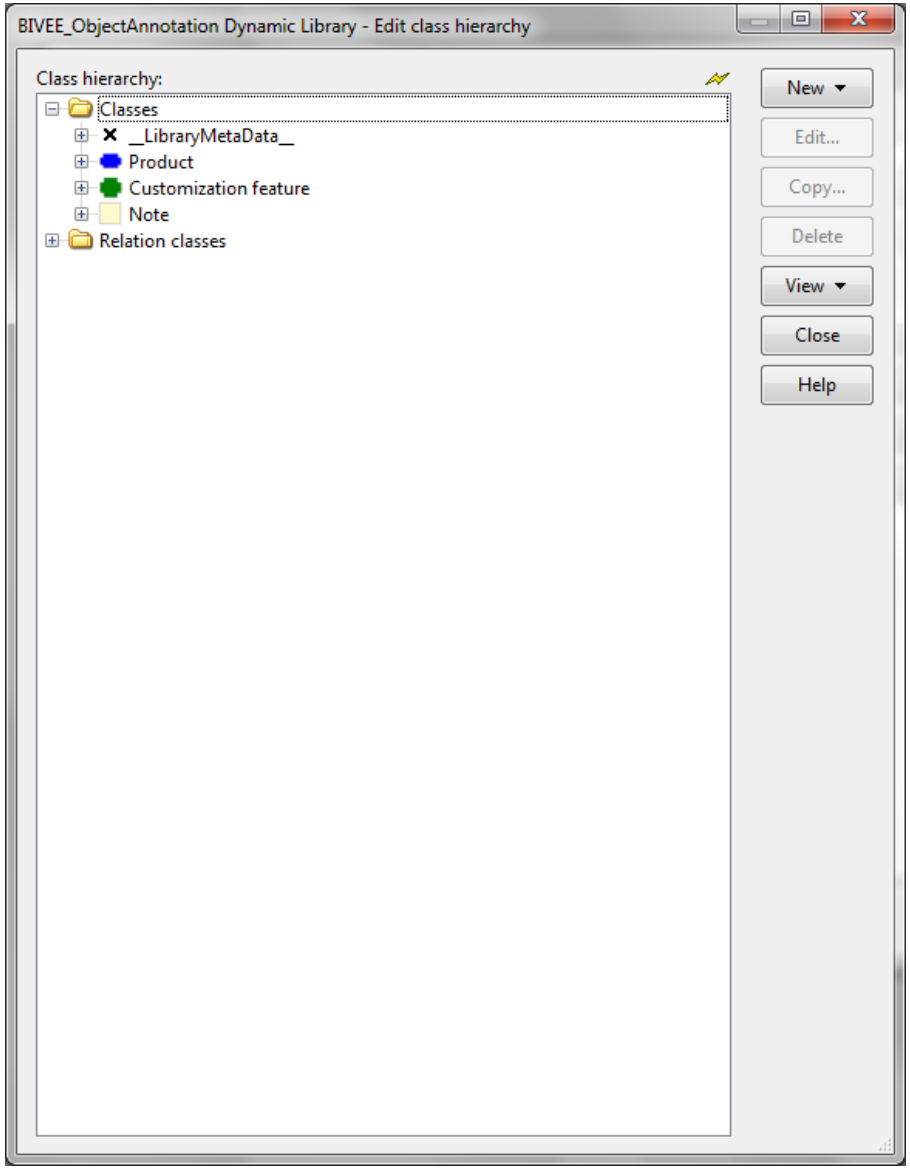
- Modelling Language Implementation
  - **Classes**
  - **Relations**
  - **Class Attributes and Attributes**
    - **GRAPHREP**
    - **ATTRREP**
    - CLASS Cardinality
    - CONVERSION
    - Model Pointer
  - **Attribute Facets**
  - **Model Types**

- 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

SL2 - Textual Annotation within Model

HANDS ON: SL2 - Textual Annotation within Model



1. Create Class
   - Product
   - Customization Feature
   - Note

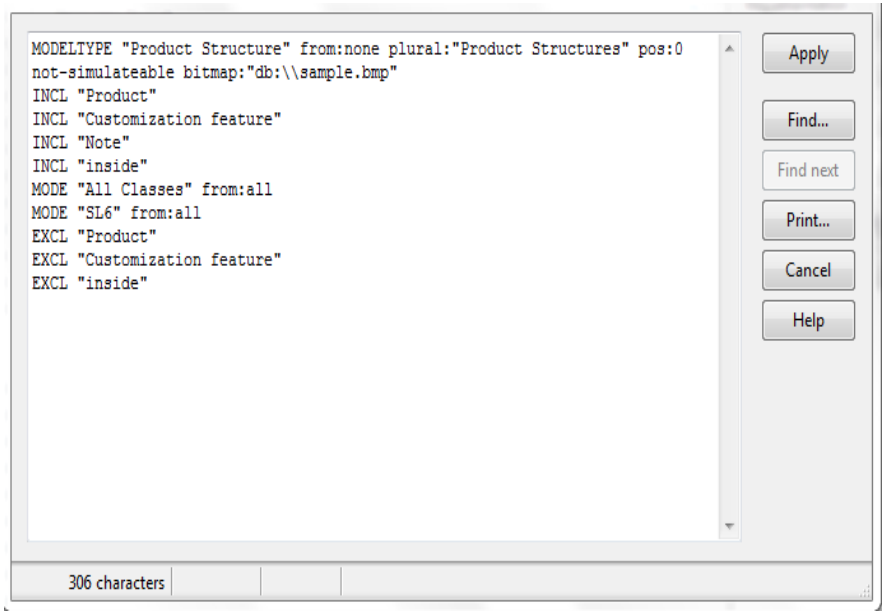
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

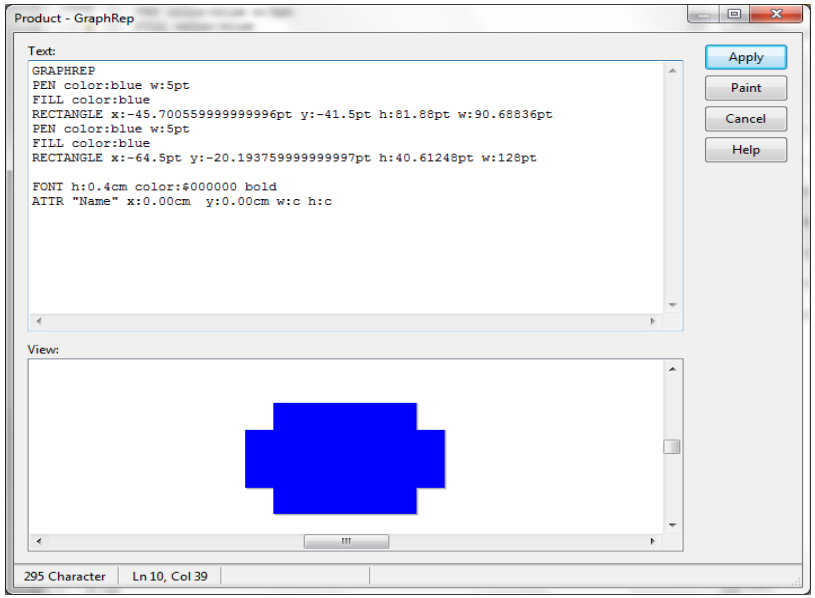
3. Define Attribute
   - <Customization Feature>:
     - "Annotation" of Type ENUMERATIONLIST (see slide **SL2 - Textual Annotation within Model: Code**)
     - „Type" (ENUMERATION)

4. Define AttrRep for class "Customization Feature":

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Annotation"
ATTR "Type"
```

HANDS ON: SL2 - Textual Annotation within Model





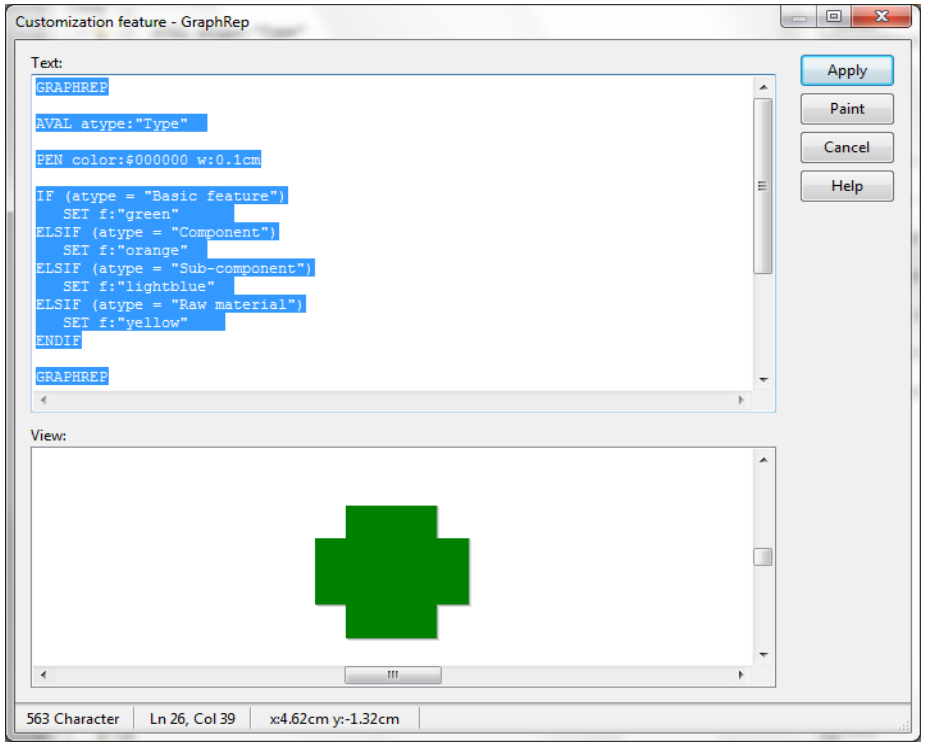
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

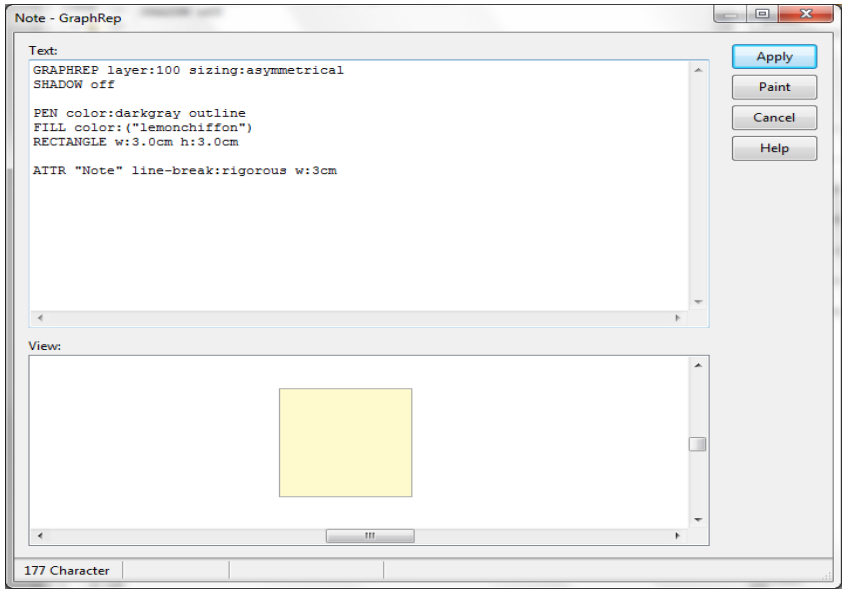
HANDS ON: SL2 - Textual Annotation within Model



7. Define GraphRep for “Customization Feature”:

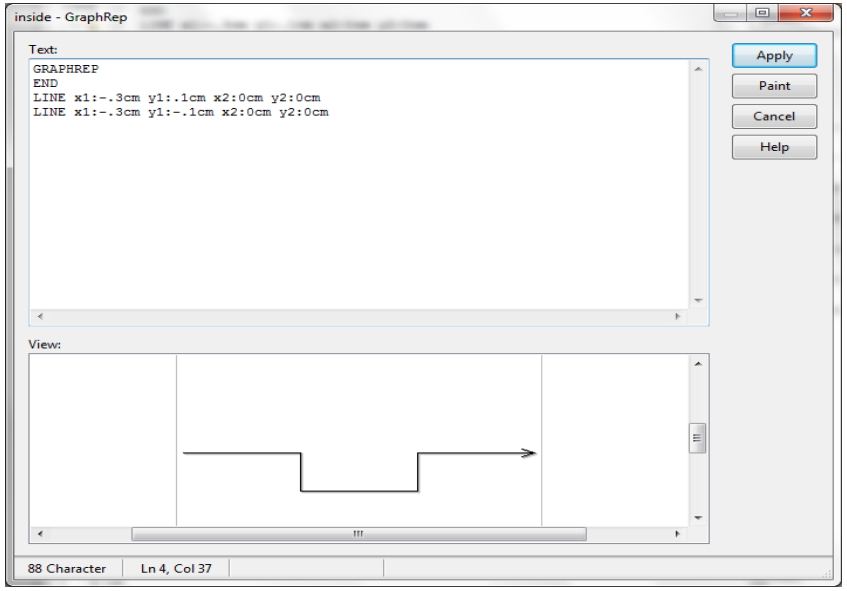
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL2 - Textual Annotation within Model



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

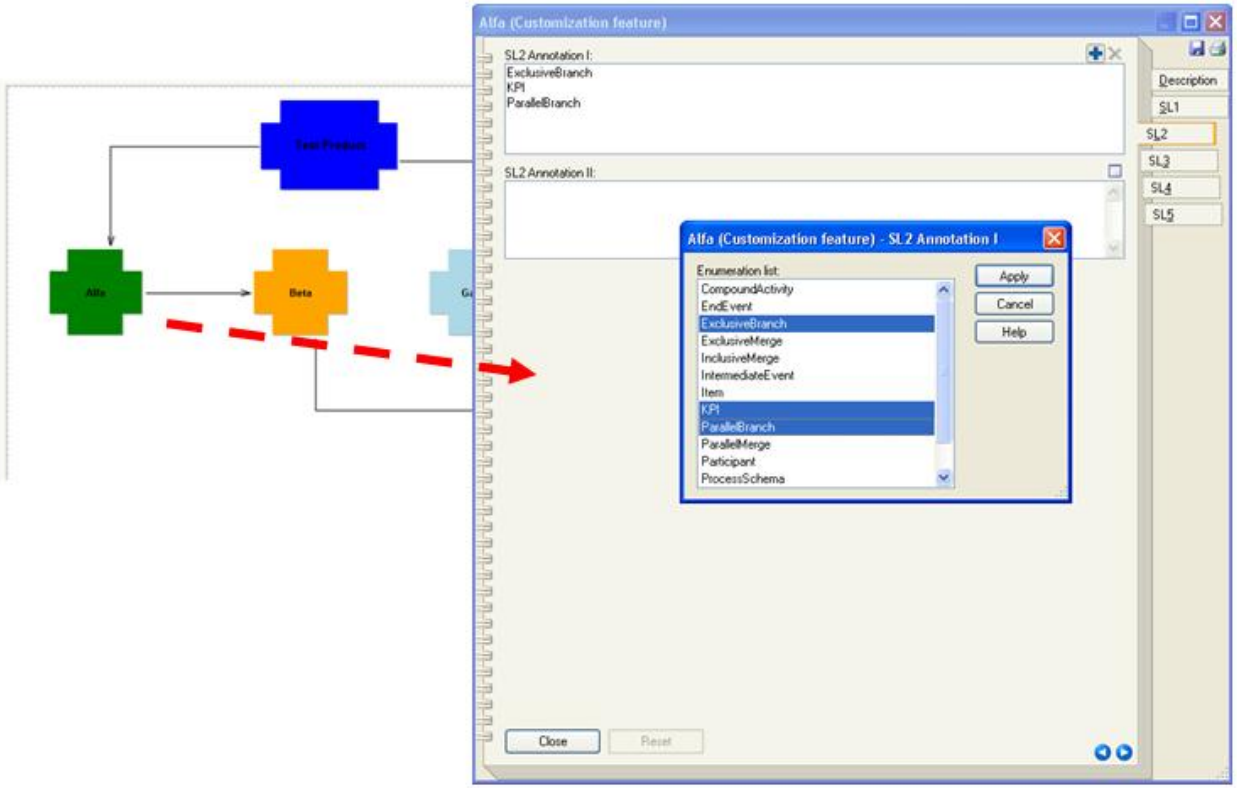
SL2 - Textual Annotation within Model: Code

```
            ATTRIBUTE <SL2 Annotation I>
            TYPE ENUMERATIONLIST

                        FACET <EnumerationDomain>
                        VALUE
"CompoundActivity@EndEvent@ExclusiveBranch@ExclusiveMerge@InclusiveMerge@IntermediateEvent@I
tem@KPI@ParallelBranch@ParallelMerge@Participant@ProcessSchema@StartEvent@Task"
                        FACET <MultiLineString>
                        VALUE 0
                        FACET <AttributeHelpText>
                        VALUE ""
                        FACET <AttributeRegularExpression>
                        VALUE ""

            ATTRIBUTE <SL2 Annotation I>
            VALUE ""
```

Result:

SL2 - Textual Annotation within Model: Scenario

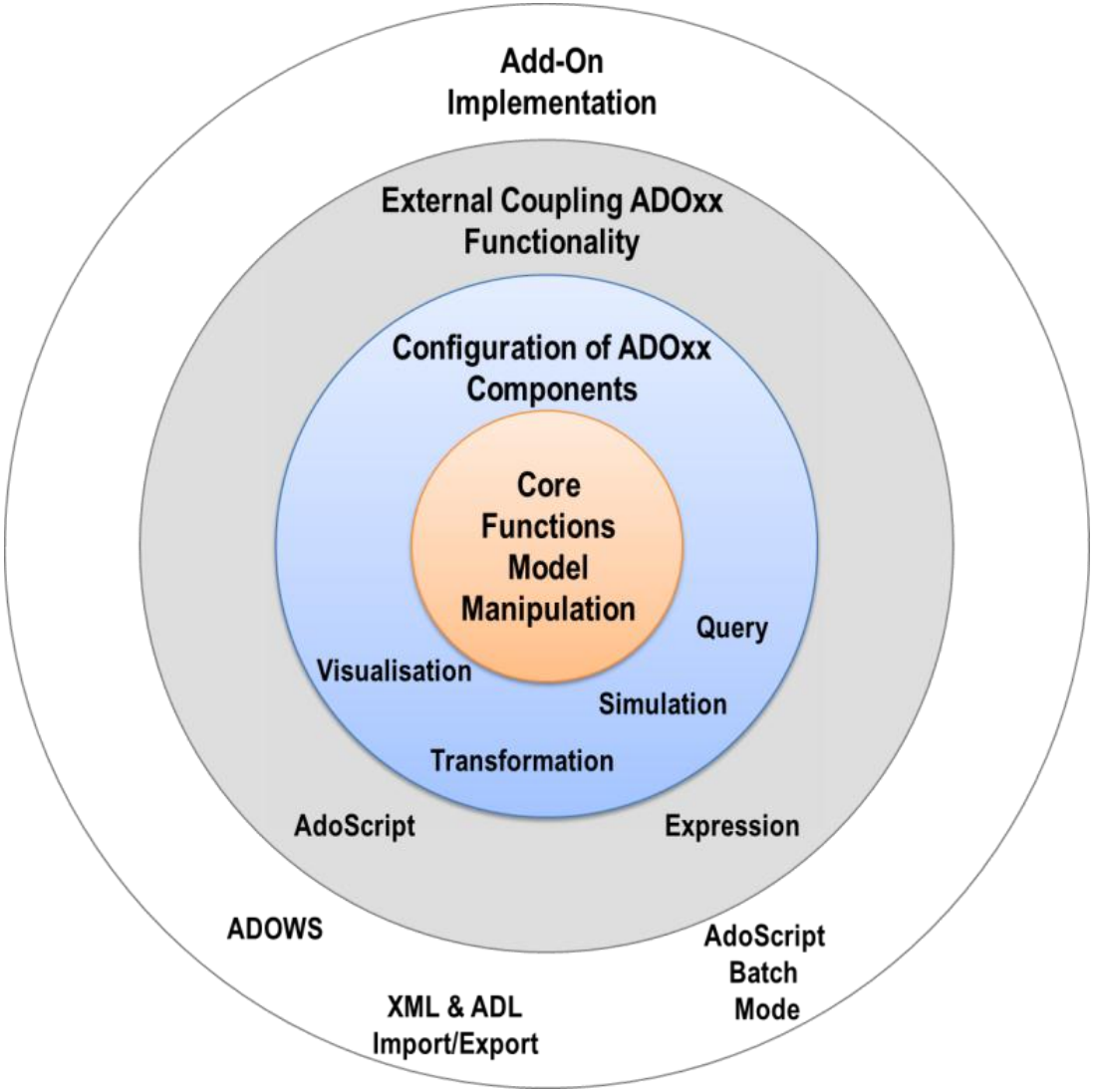
This scenario provides a functionality to annotate desired modelling object with either predefined set of concepts (hard-coded/selected by an expert) or to add a new previously non existing annotation by using a free text input field



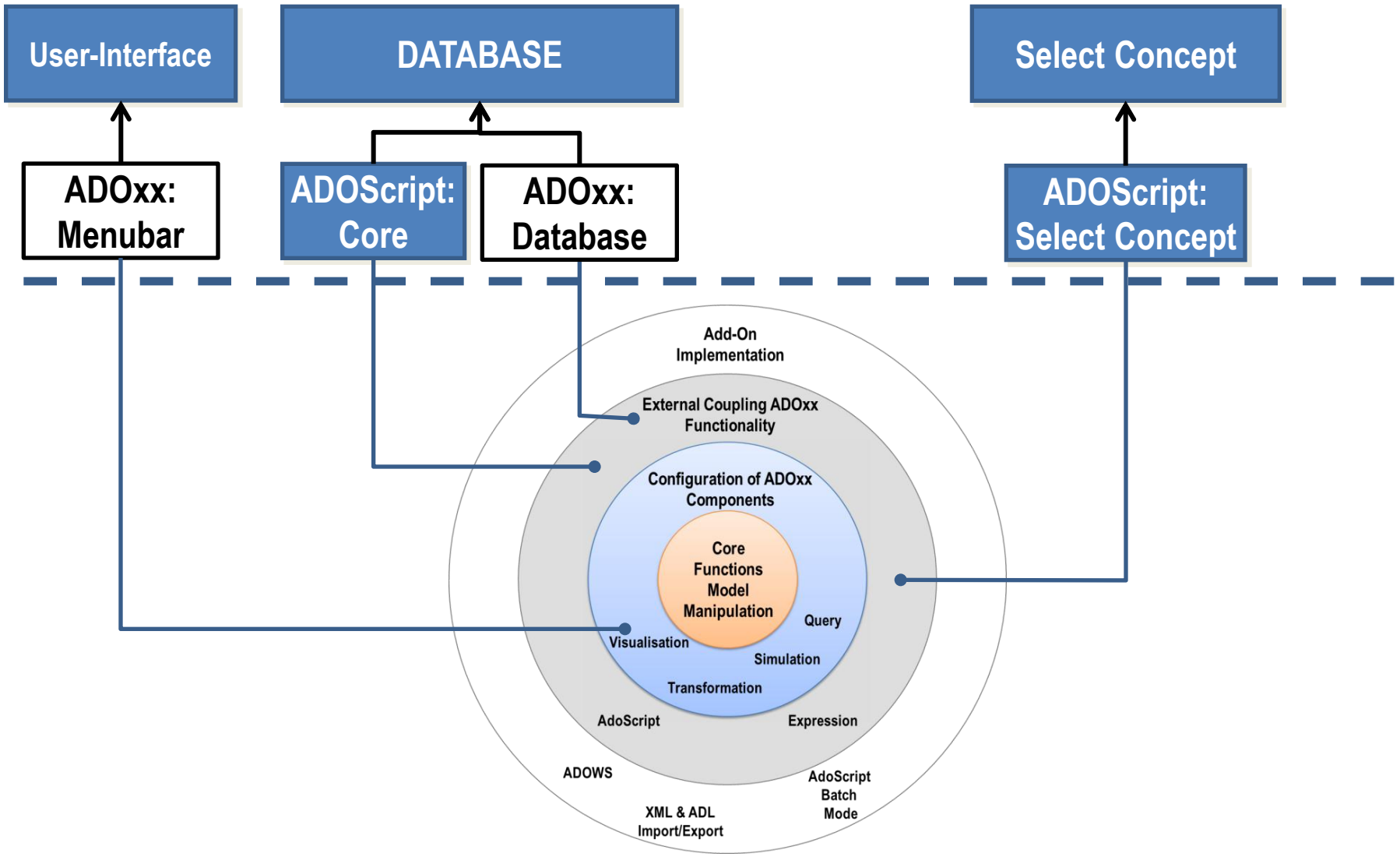
Mechanisms and Algorithms

SL3 – Model to Model Instance through the Semantic Transit Model

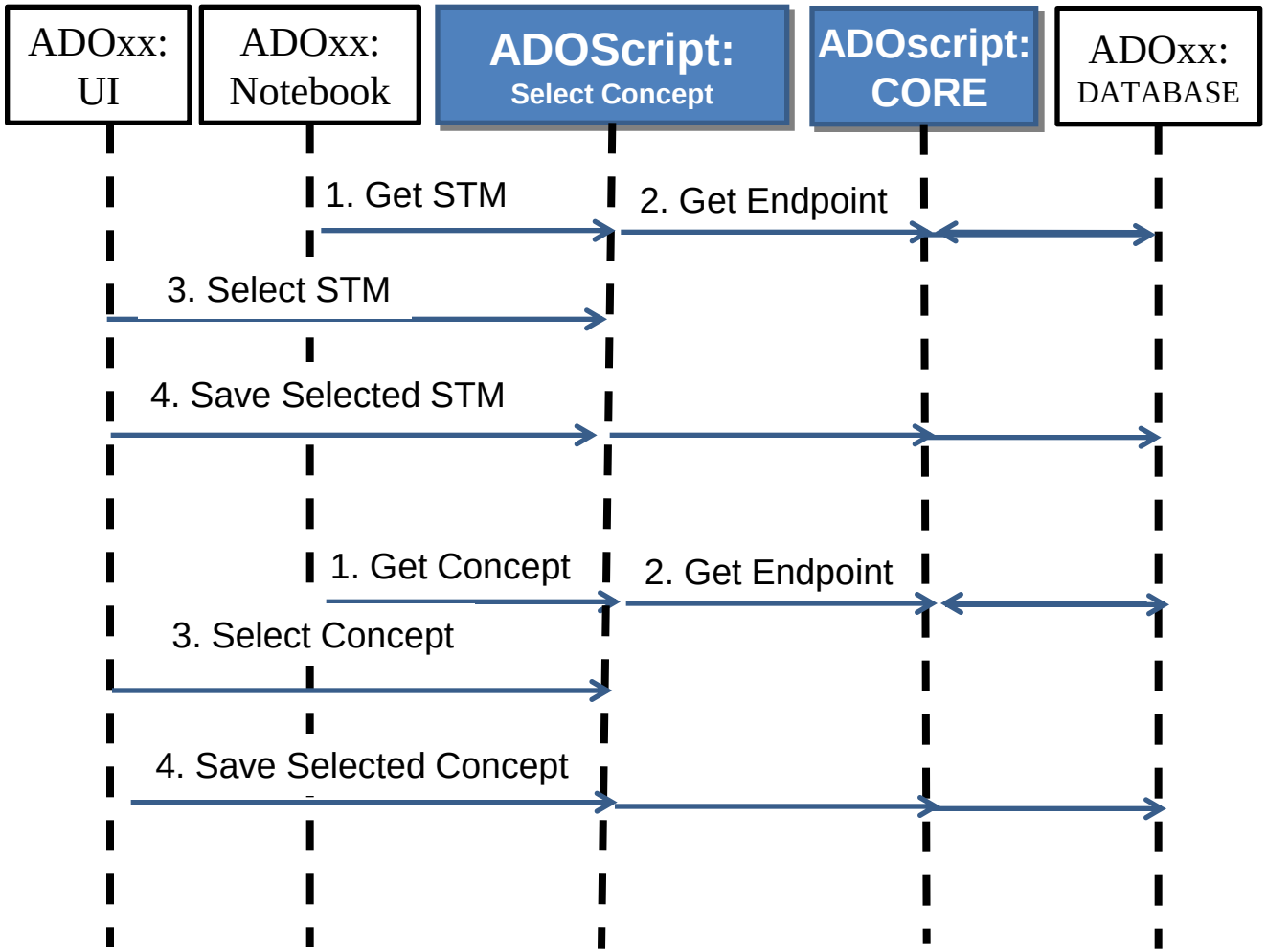
ADOxx Functionality on Meta Level



Mapping ADOxx Functionality



Description of the Algorithm



STM…..Semantic Transit Model

Provided Functionality of Metamodelling Platform

Used meta-modelling functionality :

- **Meta²Model:** MODELTYPE, GRAPHREP, ATTREP, ATTRIBUTE TYPE, CLASS

- **ADOxx Meta2Model Component:**
  - Model Editor incl. Notebook
  - ADOscript interpreter and ADOscript syntax
  - Database

HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model

1. Defining **MODELTYPES**
2. Inheriting **CLASSES** from ADOxx Meta Model
3. Implementing **GRAPHREP**
4. Inherit **RELATIONCLASSES** from ADOxx Meta Model
5. Defining **ATTRIBUTES** and **ATTREP**
6. Implement **AdoScript** to retrieve and select concepts from semantic transit model

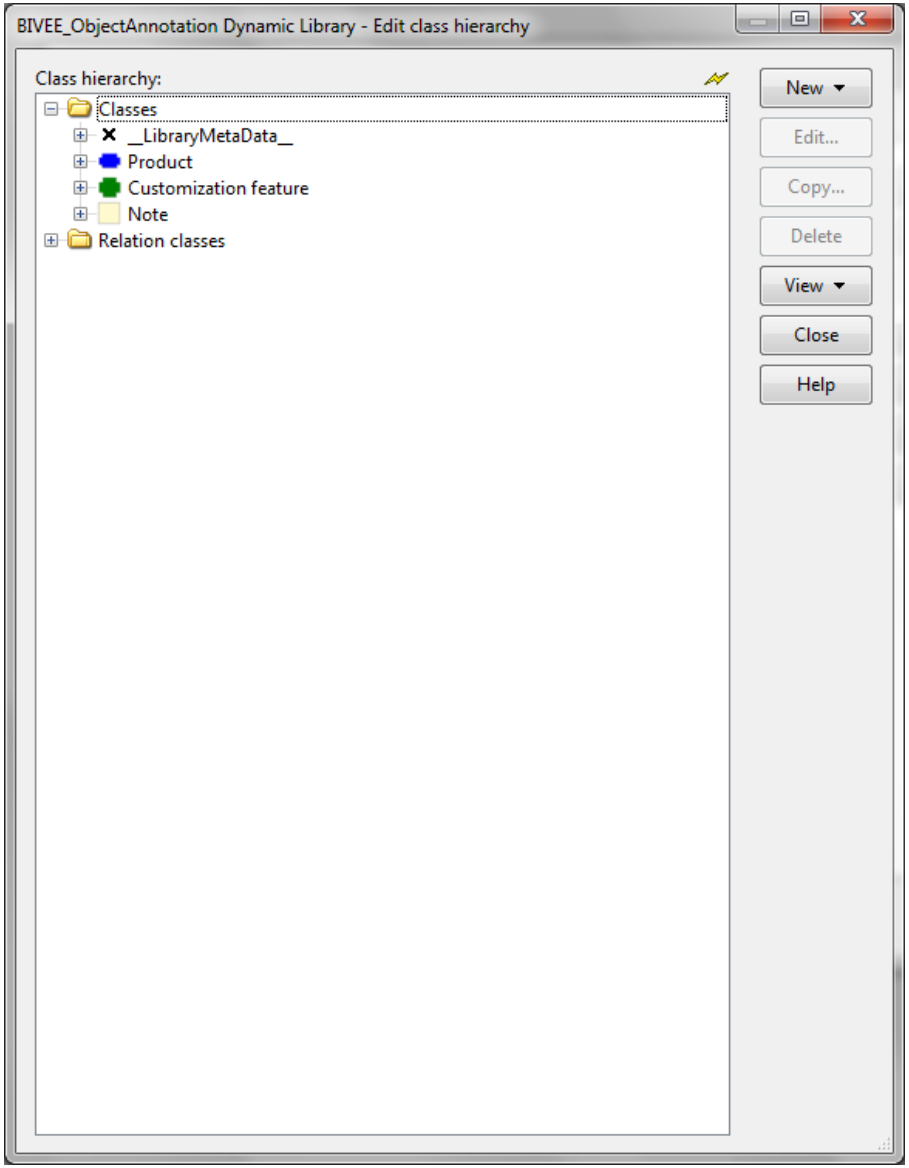
Used ADOxx Functionality

- 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

SL3 – Model to Model Instance through the Semantic Transit Model

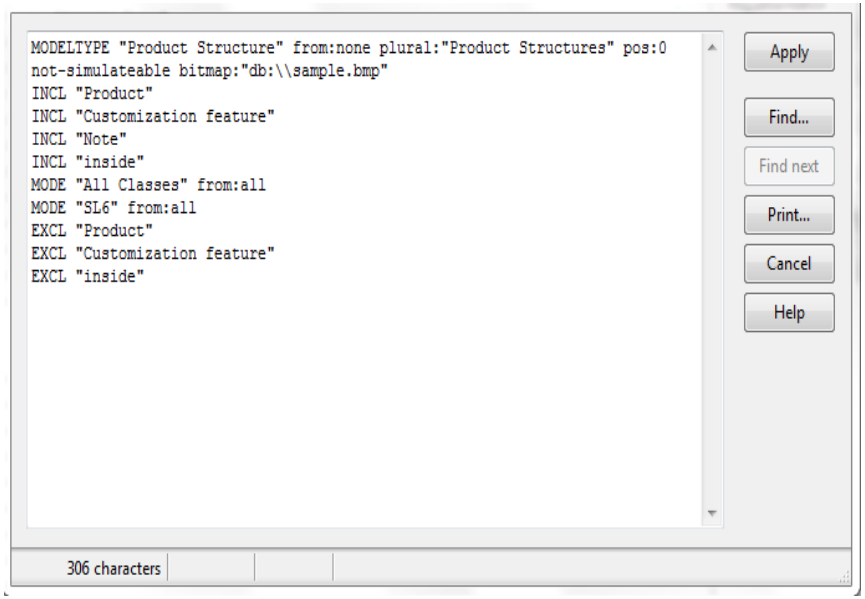
HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model



1. Create Class
   - Product
   - Customization Feature
   - Note
   - Concept
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature
3. Define Atttributes:
   - <Customization Feature>:
     - "Select Concept" (ProgramCall)
       (see slide **SL3 - Model to Model Instance through the Semantic Transit Model: Code 1+2**)
     - „Semantic Transit Model" (INTERREF)
     - „Detailed Annotation" (INTERREF)
     - „Type" (ENUMERATION)
   - <Concept>:
     - "Concept Type" (INTERREF)
4. Define AttrRep for class "Customization Feate":

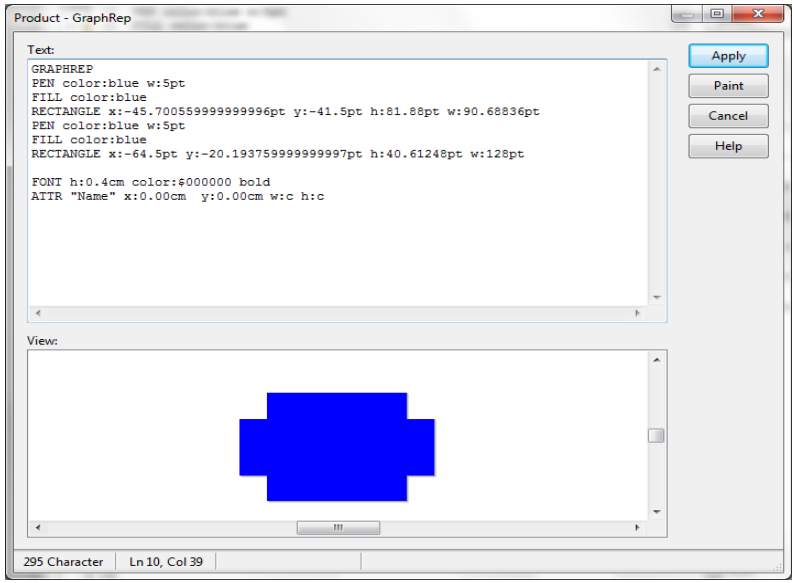
```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Select Concept"
ATTR "Type"
```

HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model



5. Define Modeltype:

MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside“

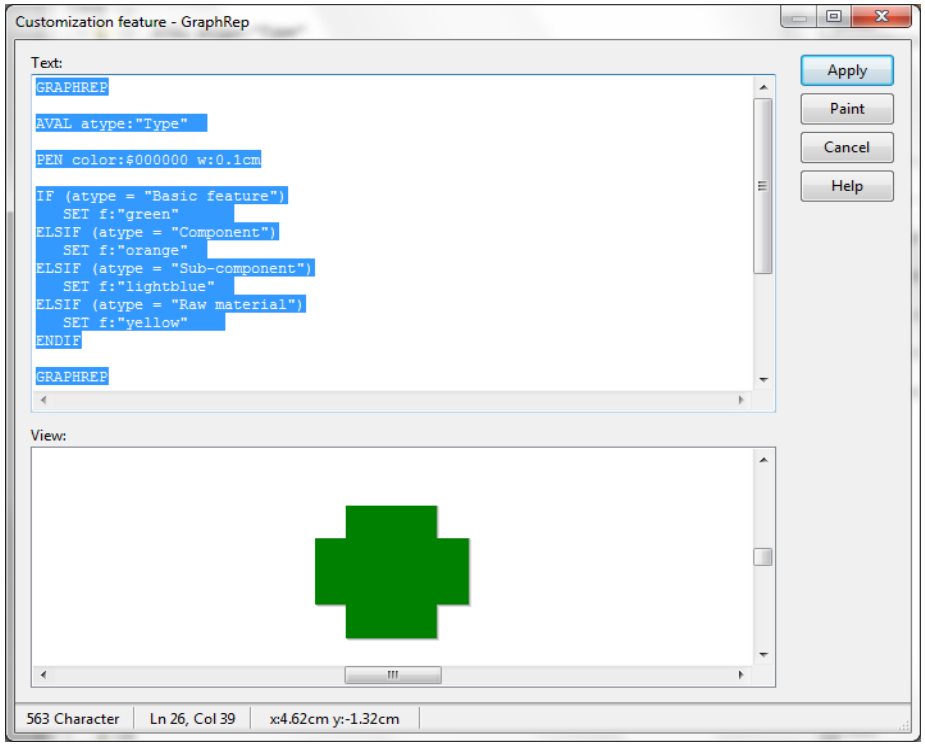


6. Define GraphRep for “Product”

GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm y:0.00cm w:c h:c

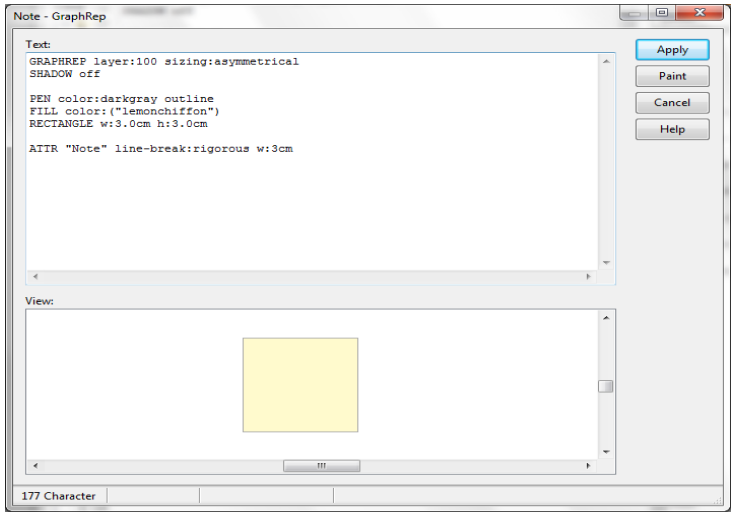
HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model



7. Define GraphRep for “Customization Feature”:

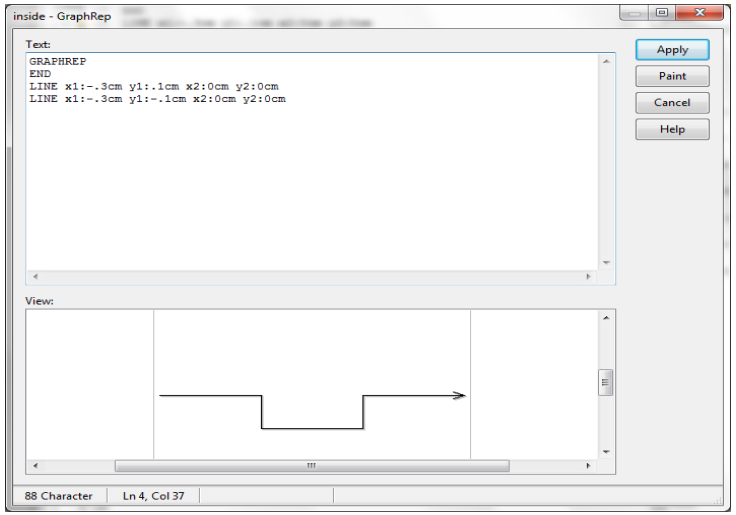
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

HANDS ON: SL3 - Model to Model Instance through the Semantic Transit Model: Code (1/2)

```
ITEM "Select Concept"
SET my_objid: (STR objid)
SET myobjid: (VAL my_objid)
CC "Modeling" GET_ACT_MODEL
SET myModelID: (modelid)

CC "Core" GET_CLASS_ID objid: (myobjid)
CC "Core" GET_ATTR_ID classid: (classid) attrname: ("Semantic Transit Model")
CC "Core" GET_INTERREF objid: (myobjid) attrid: (attrid) index: 0
CC "Core" GET_CLASS_ID objid: (tobjid)
CC "Core" GET_ATTR_ID classid: (classid) attrname: ("Concept type")
CC "Core" GET_INTERREF objid: (tobjid) attrid: (attrid) index: 0

SET
aql_str:(("{\"")+(tobjname)+("\":\"")+(tclassname)+("\":\"")+(tmodelname)+("\":\"")+(tmodeltype)+("\"}
<-\"belongs to\""))
CC "AQL" EVAL_AQL_EXPRESSION expr: (aql_str) modelid: (tmodelid)

SET conceptIDs:(objids)

CC "AdoScript" TLB_CREATE title:"Select Detailed Annotation" oktext:"OK" canceltext:"Cancel"
boxtext:"Please choose one of the following related concepts" no-help:1 button-w:60 max-w:500 max-
h:367 min-w:200 min-h:150 sorted: 1 checklistbox:1
FOR conceptID in:(conceptIDs)
{
CC "Core" GET_OBJ_NAME objid: (VAL conceptID)
SET conceptName: (objname)
```

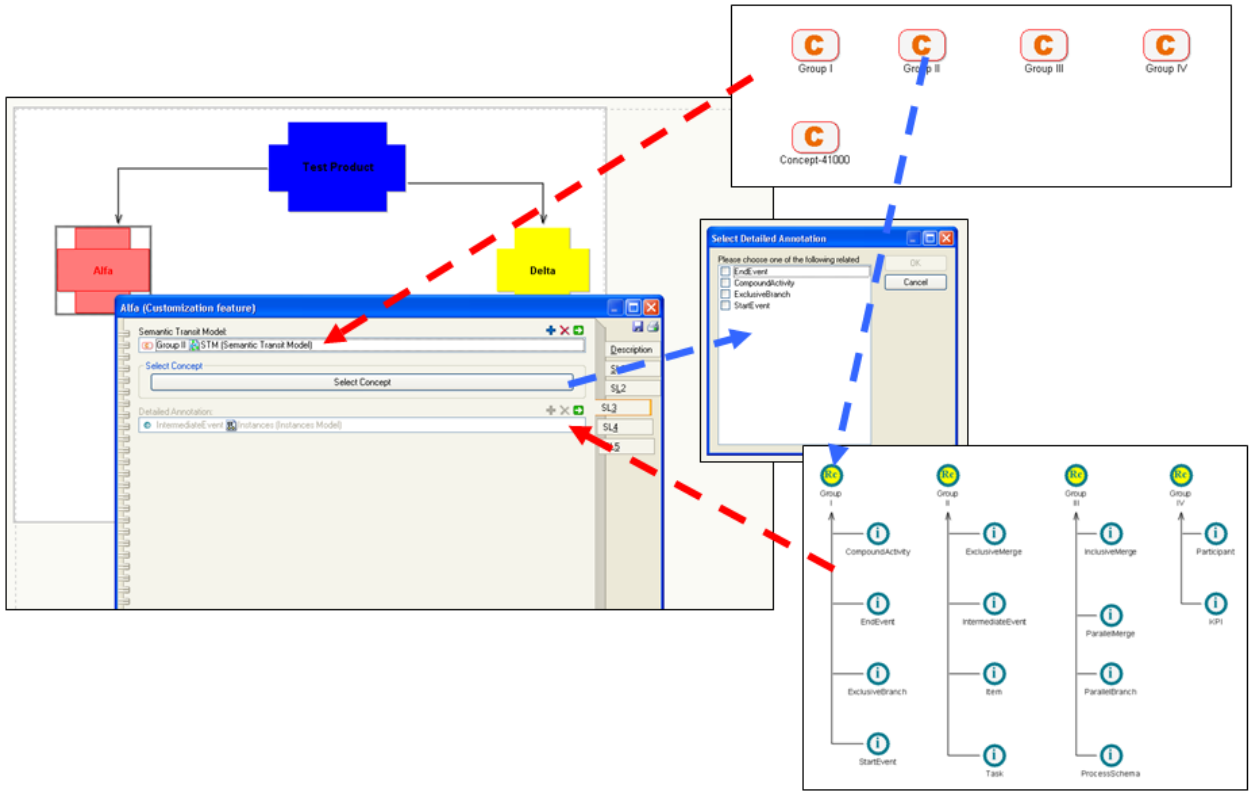
SL3 - Model to Model Instance through the Semantic Transit Model: Code (2/2)

```
CC "AdoScript" TLB_INSERT id:(VAL conceptID) text:(conceptName)
}
CC "AdoScript" TLB_SHOW
IF (ecode = 0)
{
            CC "Core" GET_CLASS_ID classname: ("Customization feature")
            SET myclassid: (classid)
            CC "Core" GET_ATTR_ID classid: (myclassid) attrname: "Detailed Annotation"
            SET myattrid: (attrid)
            CC "Core" GET_INTERREF_COUNT objid: (myobjid) attrid: (myattrid)
            IF ((count)>0)
            {
              CC  "Core" REMOVE_INTERREF objid: (myobjid) attrid: (myattrid) index: 0
            }
            FOR mySelectedID in: (selectedids)
            {
             CC "Core" ADD_INTERREF objid: (myobjid) attrid: (myattrid) tobjid: (VAL mySelectedID)
             EXIT
            } }
ELSE
{
    CC "AdoScript" INFOBOX ("You cancelled the dialog without selecting detailed annotation!")
}
```

Result:

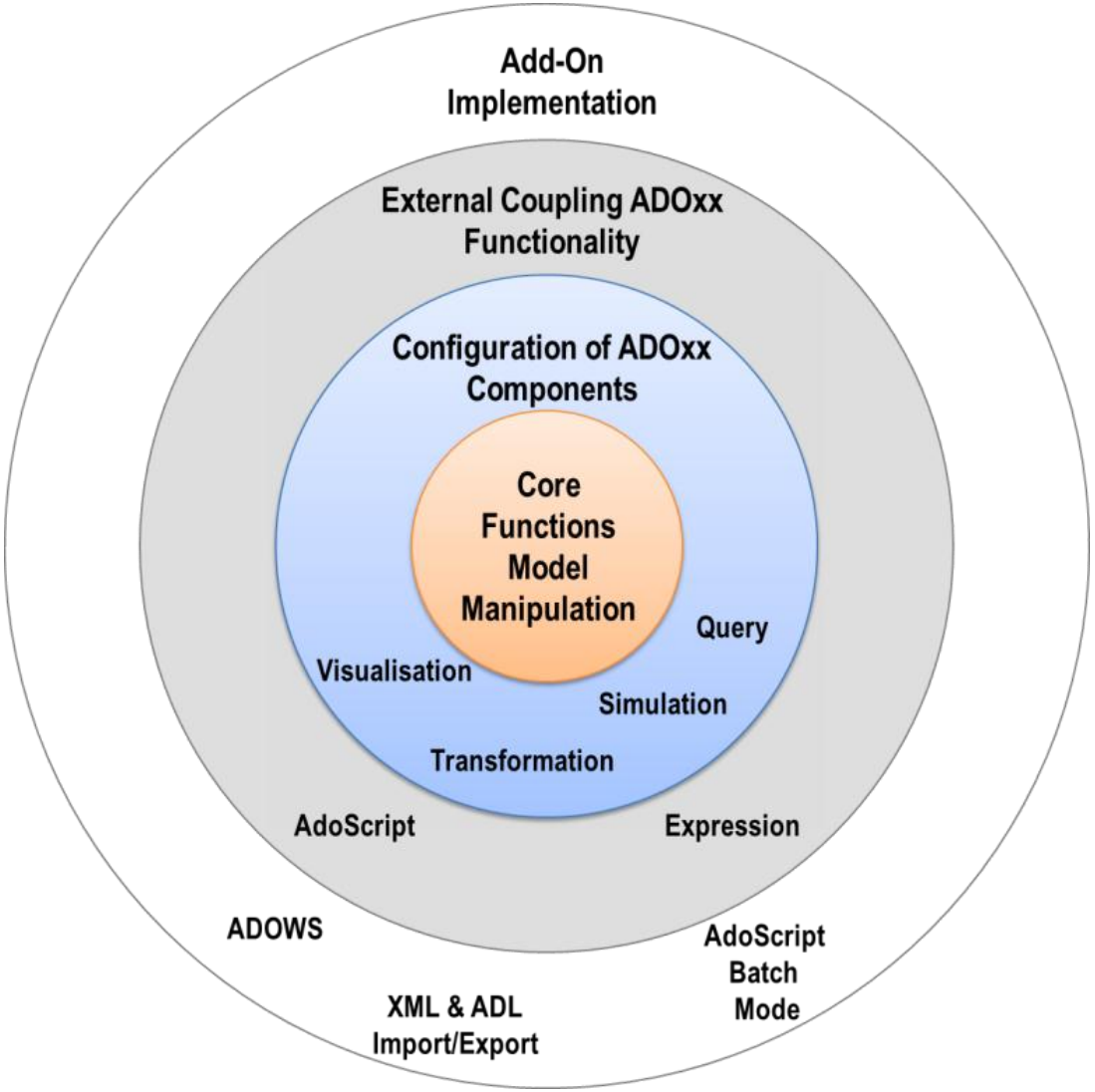
SL3 - Model to Model Instance through the Semantic Transit Model: Scenario

This scenario provides a functionality to annotate desired modelling object with a set of concepts (available as a model/modelling objects) that can be connected through a Semantic Transit Models and used as annotation concepts.

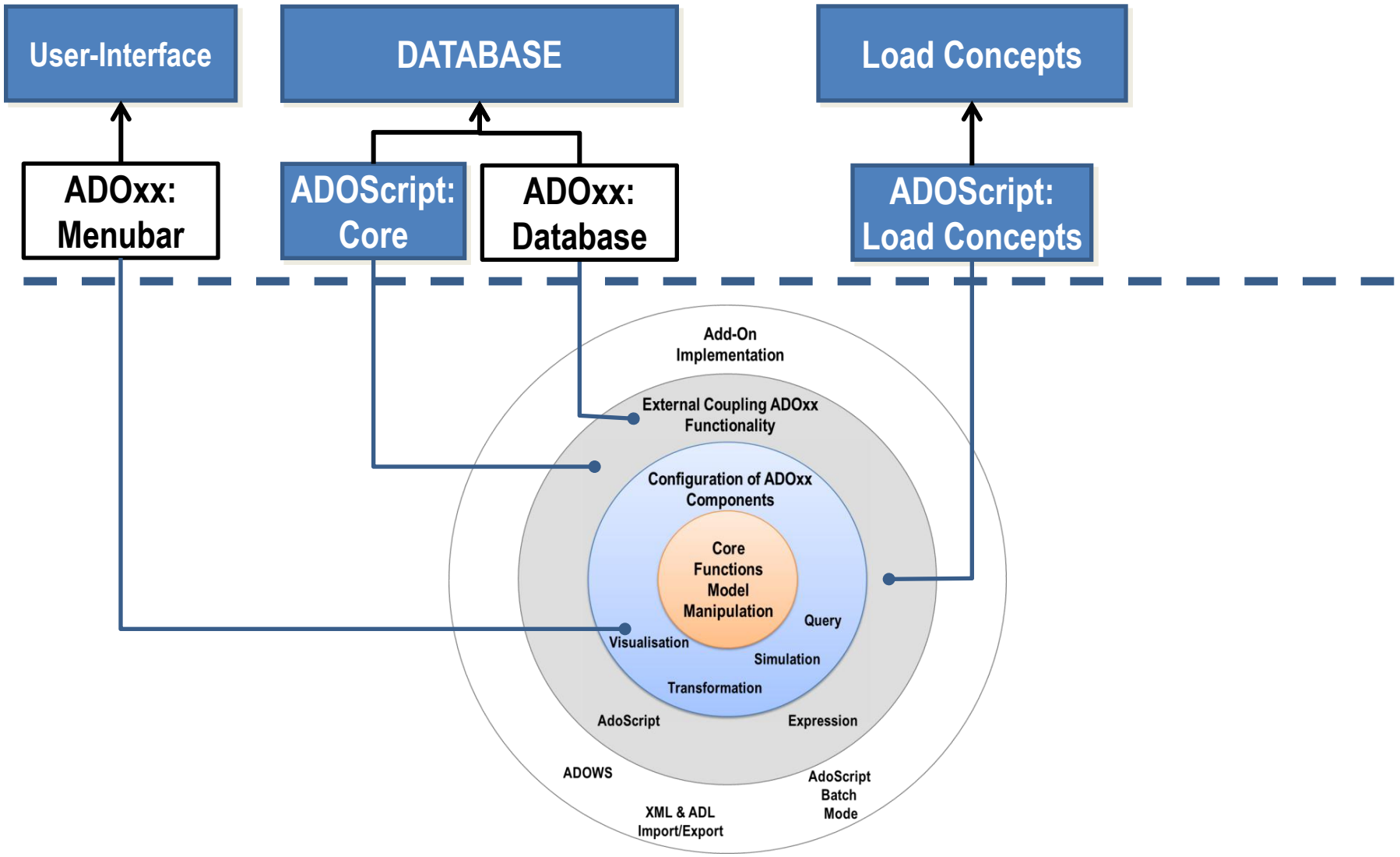


SL4 - Model to Text File with AdoScript

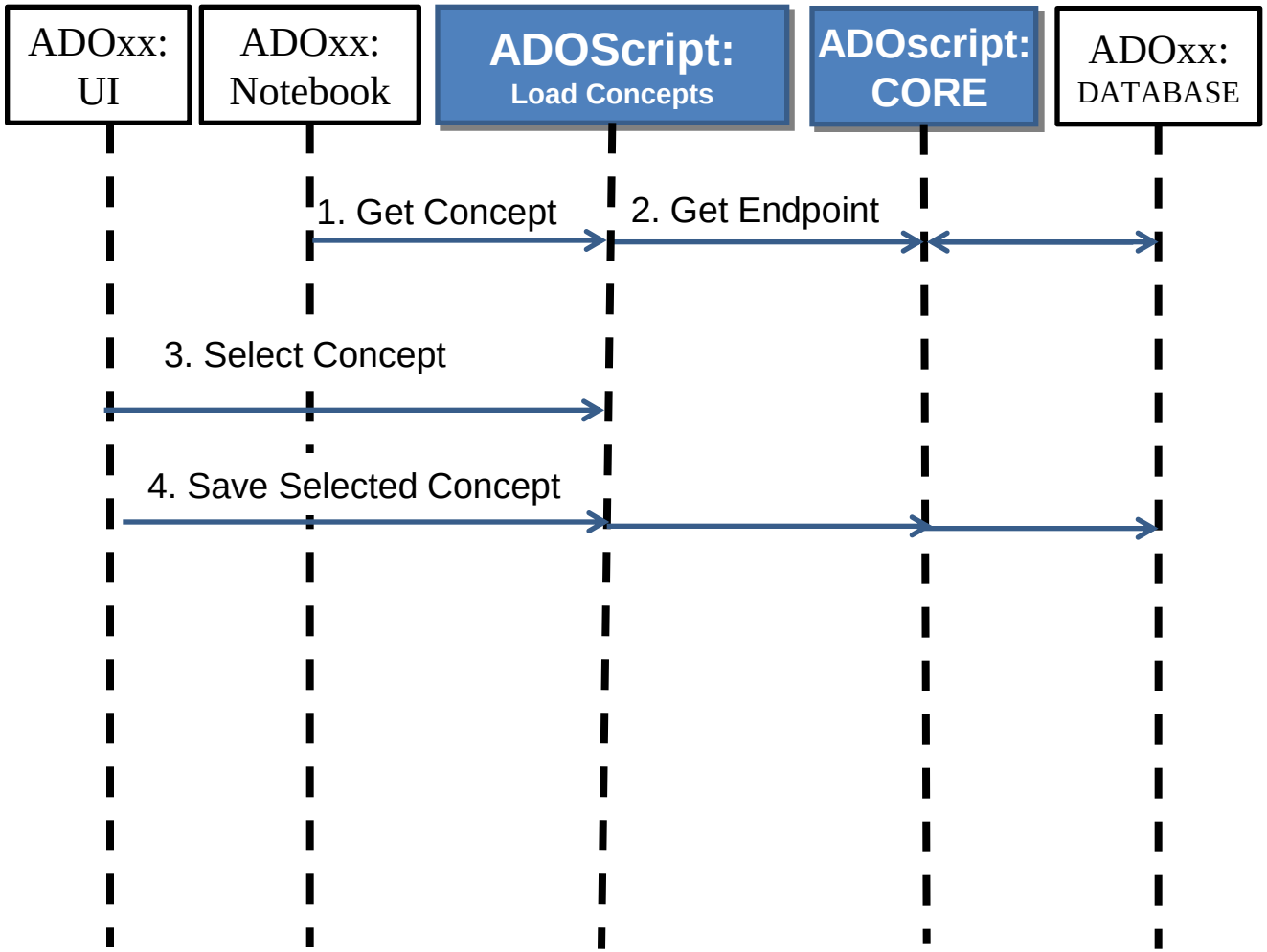
ADOxx Functionality on Meta Level



Mapping ADOxx Functionality



Description of the Algorithm



Used ADOxx Functionality

- Introduction
- Setup of Implementation Environment

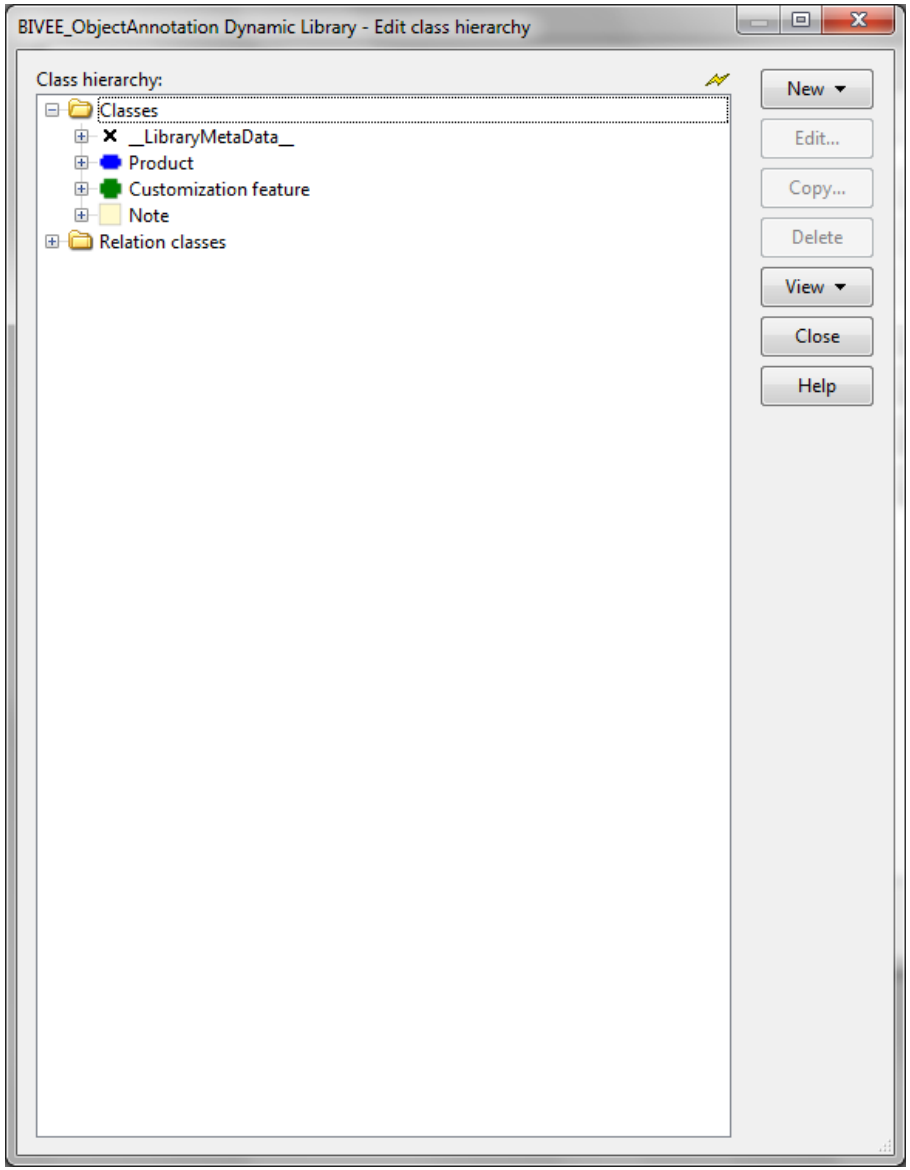
- Modelling Language Implementation
  - **Classes**
  - **Relations**
  - **Class Attributes and Attributes**
    - **GRAPHREP**
    - **ATTRREP**
    - CLASS Cardinality
    - CONVERSION
    - Model Pointer
  - **Attribute Facets**
  - **Model Types**

- 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

SL4 - Model to Text File with AdoScript

HANDS ON: SL4 - Model to Text File with AdoScript



1. Create Class
   - Product
   - Customization Feature
   - Note

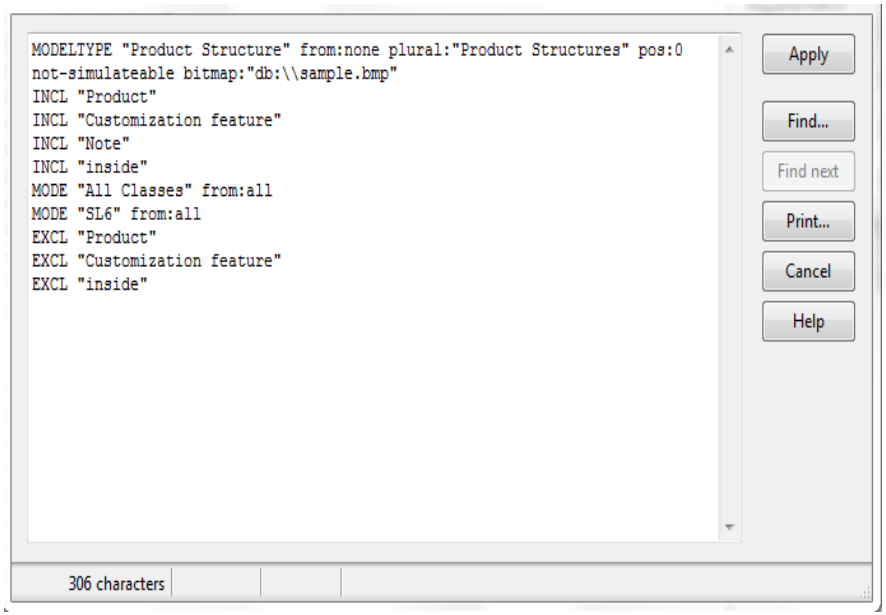
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

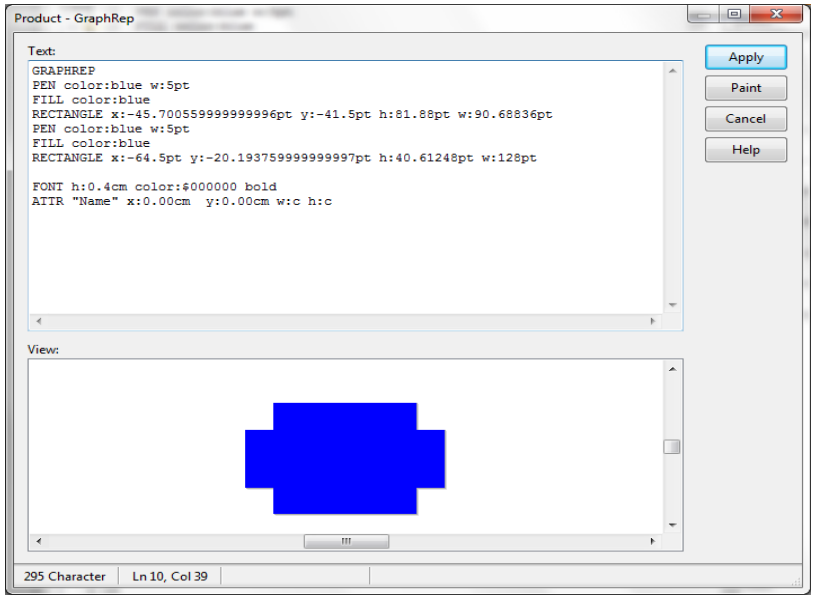
3. Define Attribute for class
   - <Customization Feature>:
     - "Load Concepts" (Programcall)
       (see slide **SL4 - Model to Text File with AdoScript: Code**)
     - "Text File" (STRING)
     - "Detailed Annotation 2" (STRING)
     - „Type" (ENUMERATION)

4. Define AttrRep for class "Customization Feate":N

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Load Concepts"
ATTR "Type"
```

HANDS ON: SL4 - Model to Text File with AdoScript





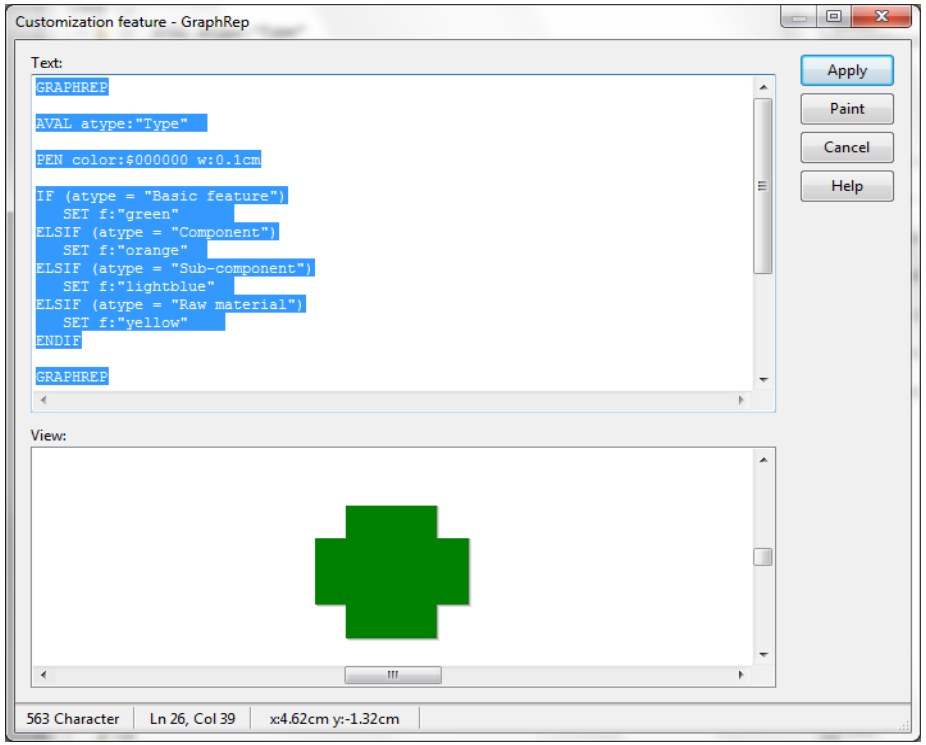
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

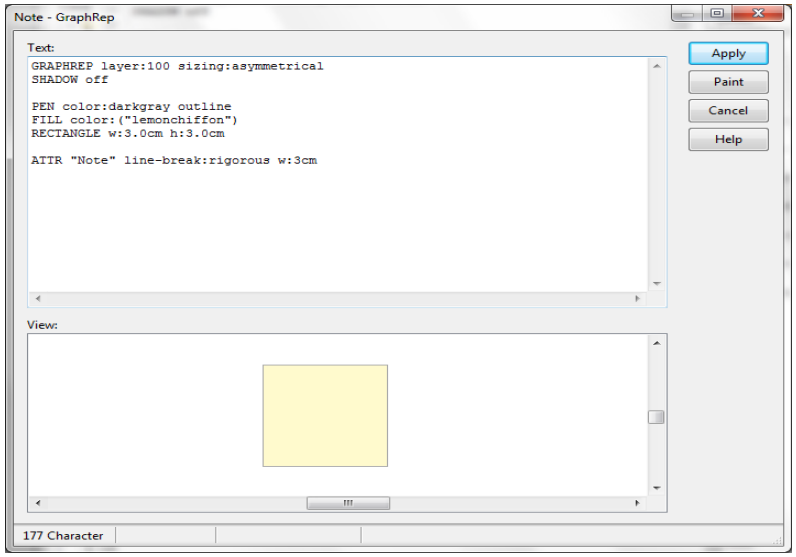
HANDS ON: SL4 - Model to Text File with AdoScript



7. Define GraphRep for “Customization Feature”:

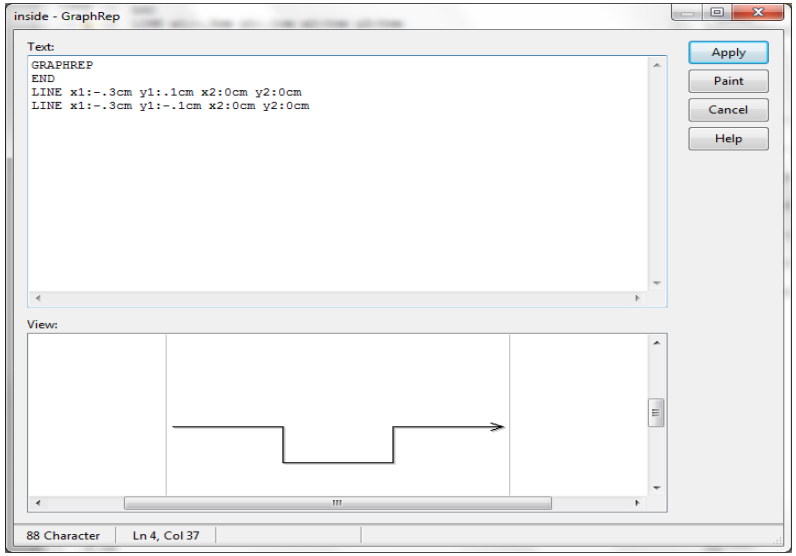
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL4 - Model to Text File with AdoScript



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

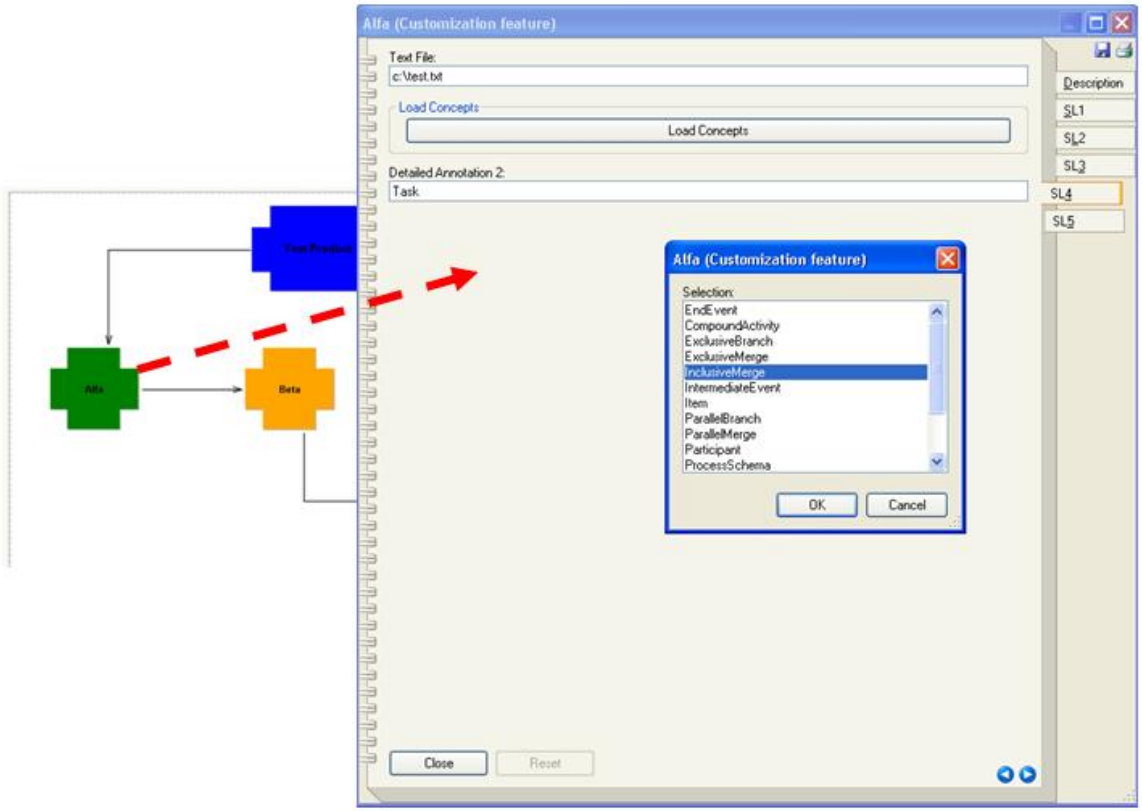
SL4 - Model to Text File with AdoScript: Code

```
ITEM "Load Concepts"
SET my_objid: (STR objid)
SET myobjid: (VAL my_objid)
CC "Modeling" GET_ACT_MODEL
SET myModelID: (modelid)
CC "Core" GET_ATTR_VAL objid:(myobjid) attrname:("Text File")
SET txt:(val)
CC "AdoScript" FREAD file: (txt)
SET text_new: (text)
CC "AdoScript" LISTBOX entries: (text_new) toksep: "@"
IF (endbutton = ("ok"))
{

            CC "Core" GET_CLASS_ID objid:(myobjid)
            SET myclassid: (classid)
            CC "Core" GET_ATTR_ID classid: (myclassid) attrname: "Detailed Annotation 2"
            SET myattrid: (attrid)
            FOR mySelectedID in: (selection)
            {
             CC "Core" SET_ATTR_VAL objid:(myobjid) attrid:(myattrid) val:(mySelectedID)
             EXIT
            }

}
ELSE
{
    CC "AdoScript" INFOBOX ("You cancelled the dialog without selecting detailed annotation!")
}
```

Result:

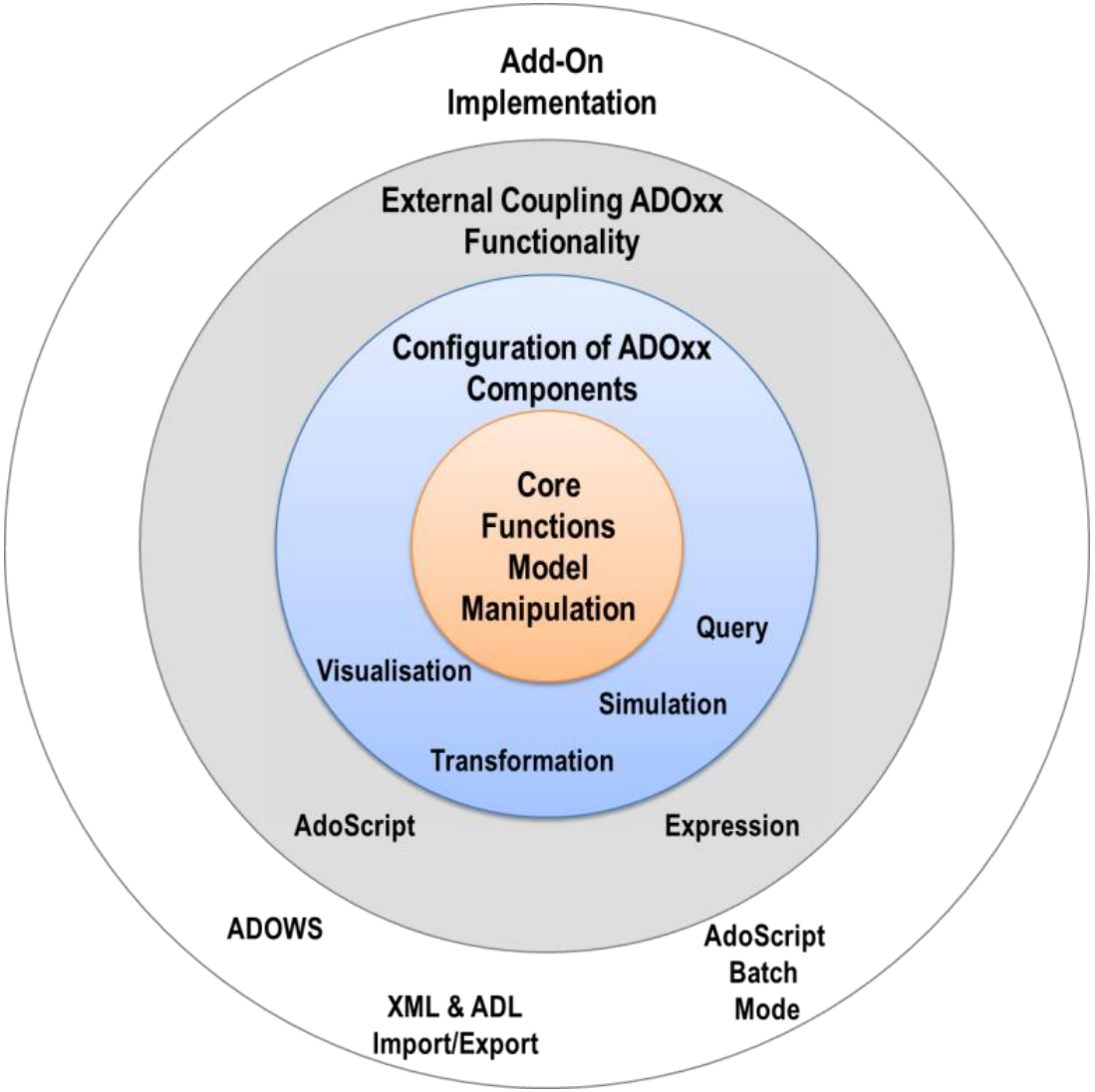
SL4 - Model to Text File with AdoScript: Scenario

This scenario provides a functionality to annotate desired modelling object with a set of concepts available in a flat file format (e.g. .txt)

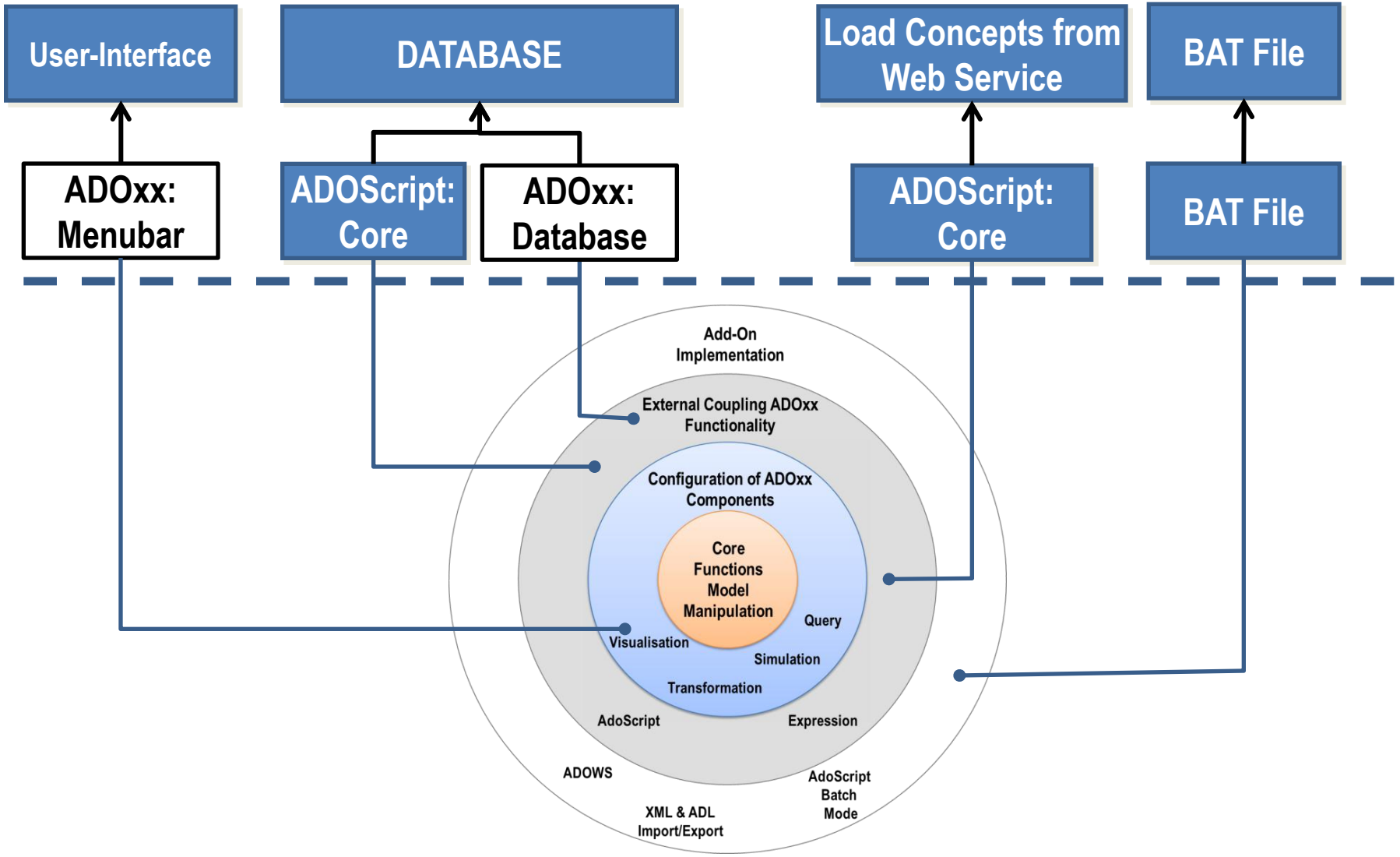


SL5 - Model to Web Service with AdoScript

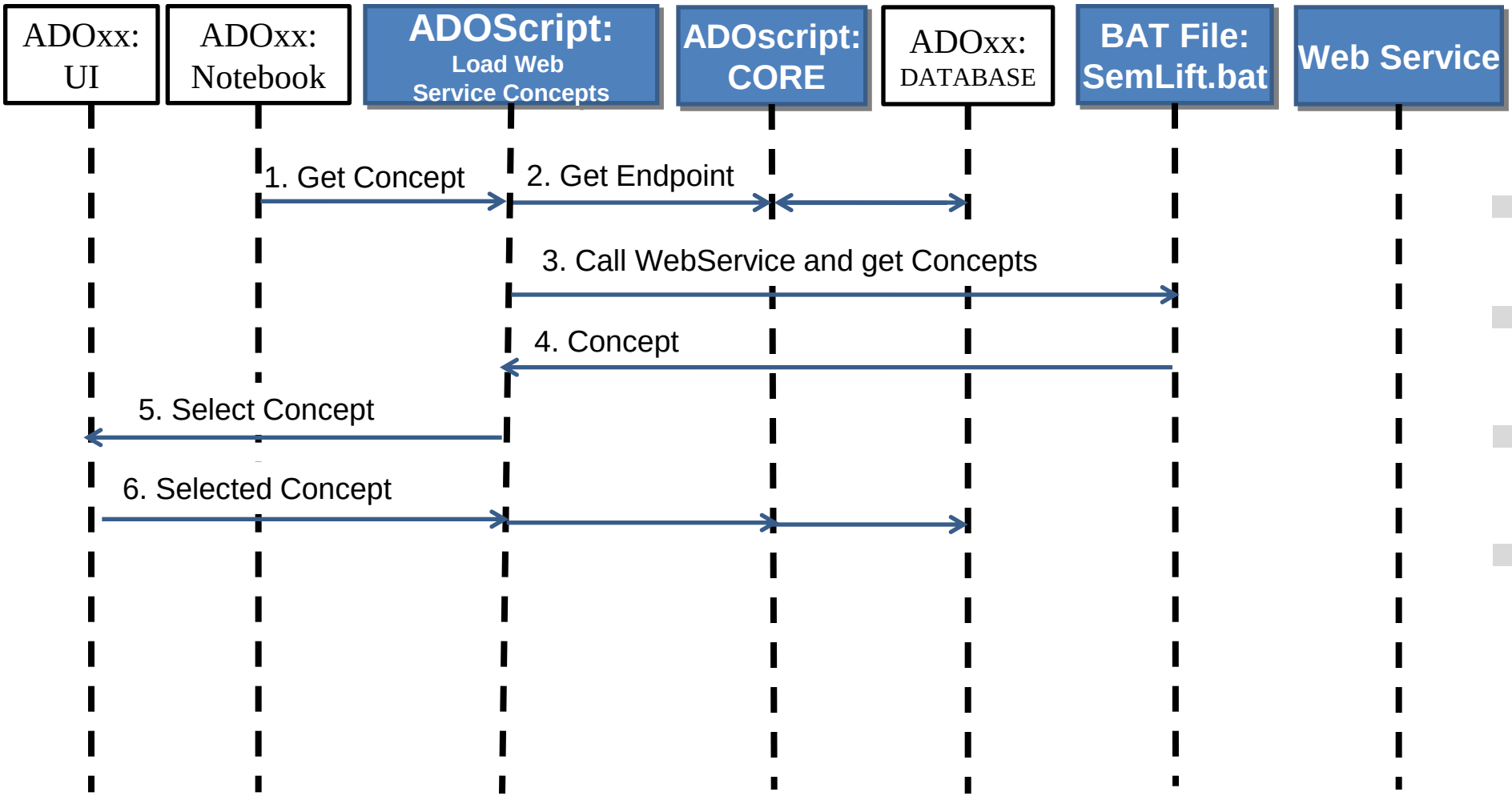
ADOxx Functionality on Meta Level



Mapping ADOxx Functionality



Description of the Algorithm



Used ADOxx Functionality

- Introduction
- Setup of Implementation Environment

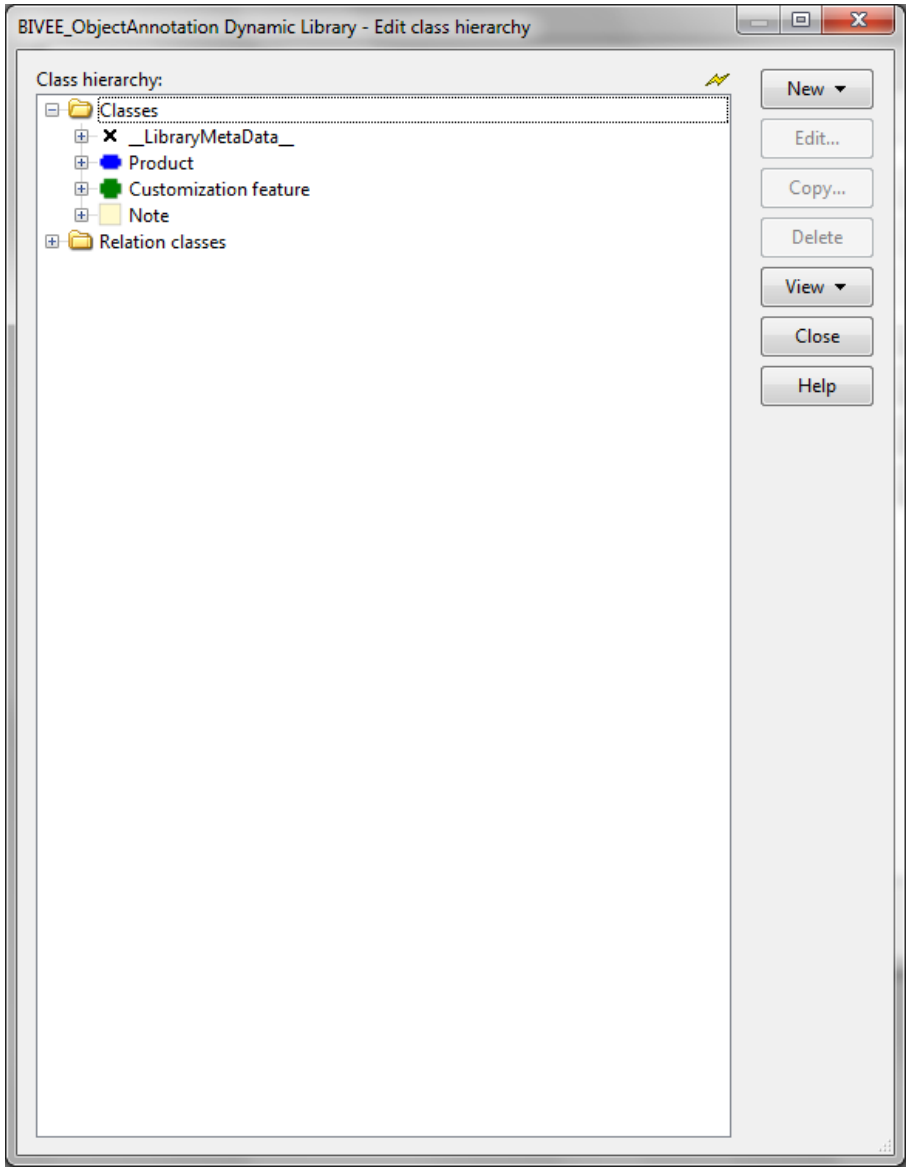
- Modelling Language Implementation
  - **Classes** ✓
  - **Relations**
  - **Class Attributes and Attributes** ✓
    - **GRAPHREP**
    - **ATTRREP** ✓
    - CLASS Cardinality
    - CONVERSION
    - Model Pointer
  - **Attribute Facets** ✓
  - **Model Types** ✓

- 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

SL5 - Model to Web Service with AdoScript

HANDS ON: SL5 - Model to Web Service with AdoScript



1. Create Class
   - Product
   - Customization Feature
   - Note

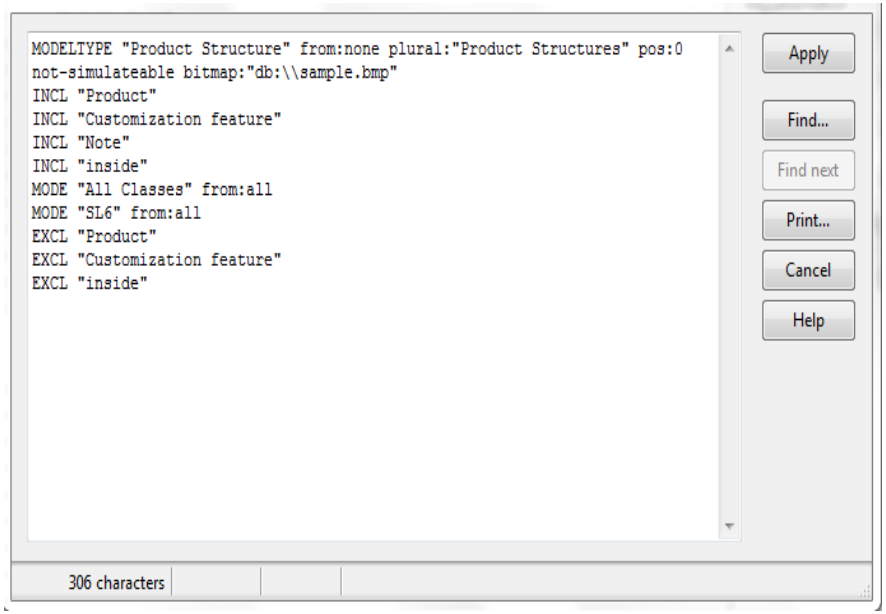
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

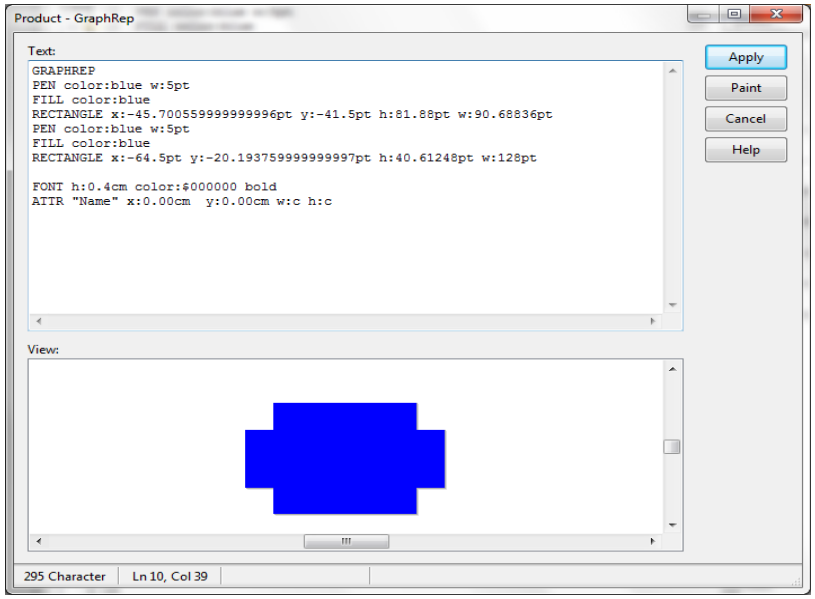
3. Define Attribute
   - <Customization Feature>:
     - "Load WebService Concepts" (ProgramCall) (see slide **SL5 - Model to Web Service with AdoScript: Code**)
     - WS Annotation (STRING)
     - WebService (STRING)
     - „Type" (ENUMERATION)

4. Define AttrRep for class "Customization Feature":

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Load WebService Concepts"
ATTR "Type"
```

HANDS ON: SL5 - Model to Web Service with AdoScript





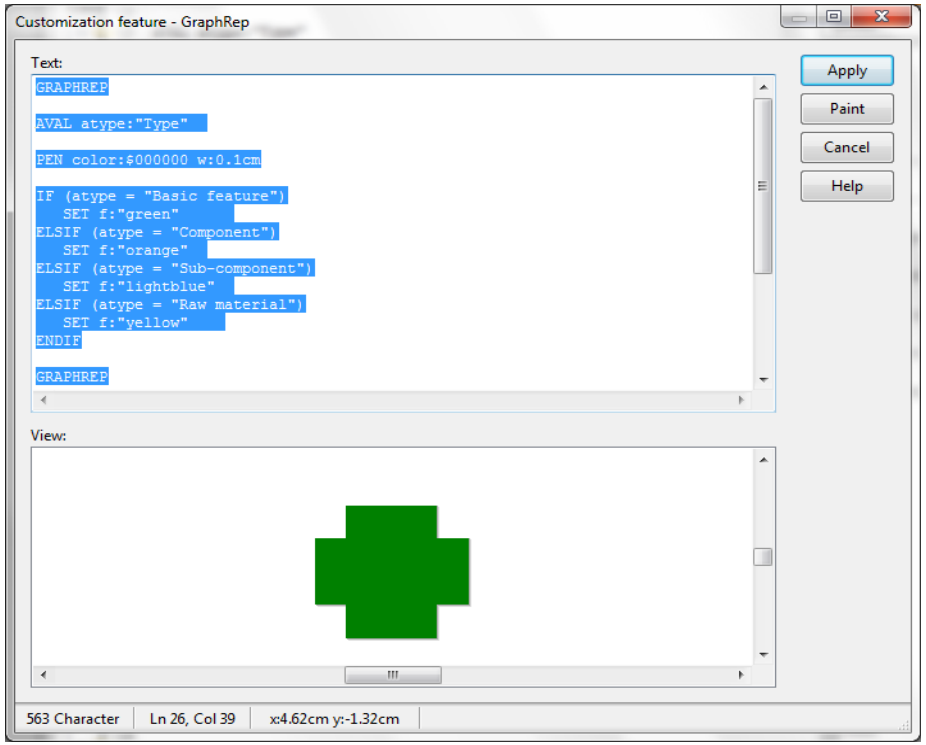
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

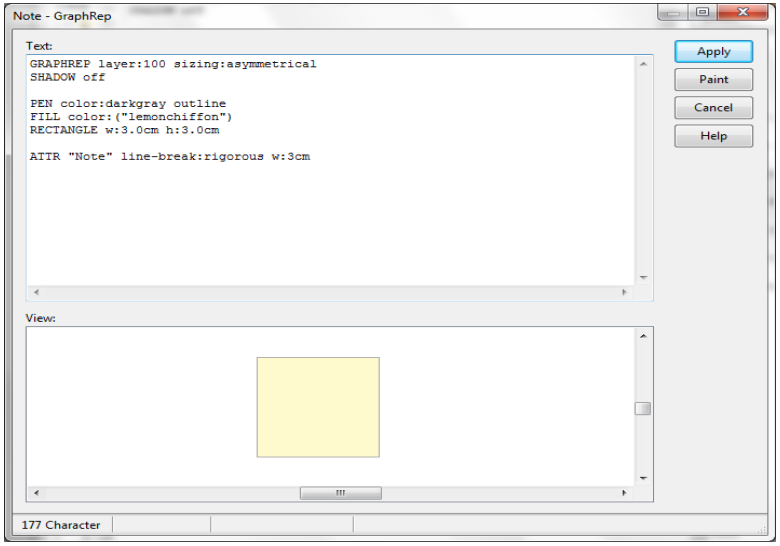
HANDS ON: SL5 - Model to Web Service with AdoScript



7. Define GraphRep for “Customization Feature”:

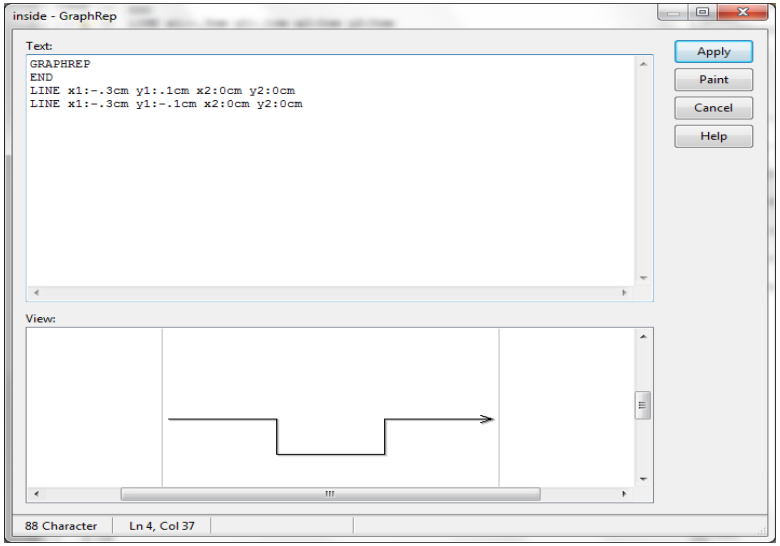
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL5 - Model to Web Service with AdoScript



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

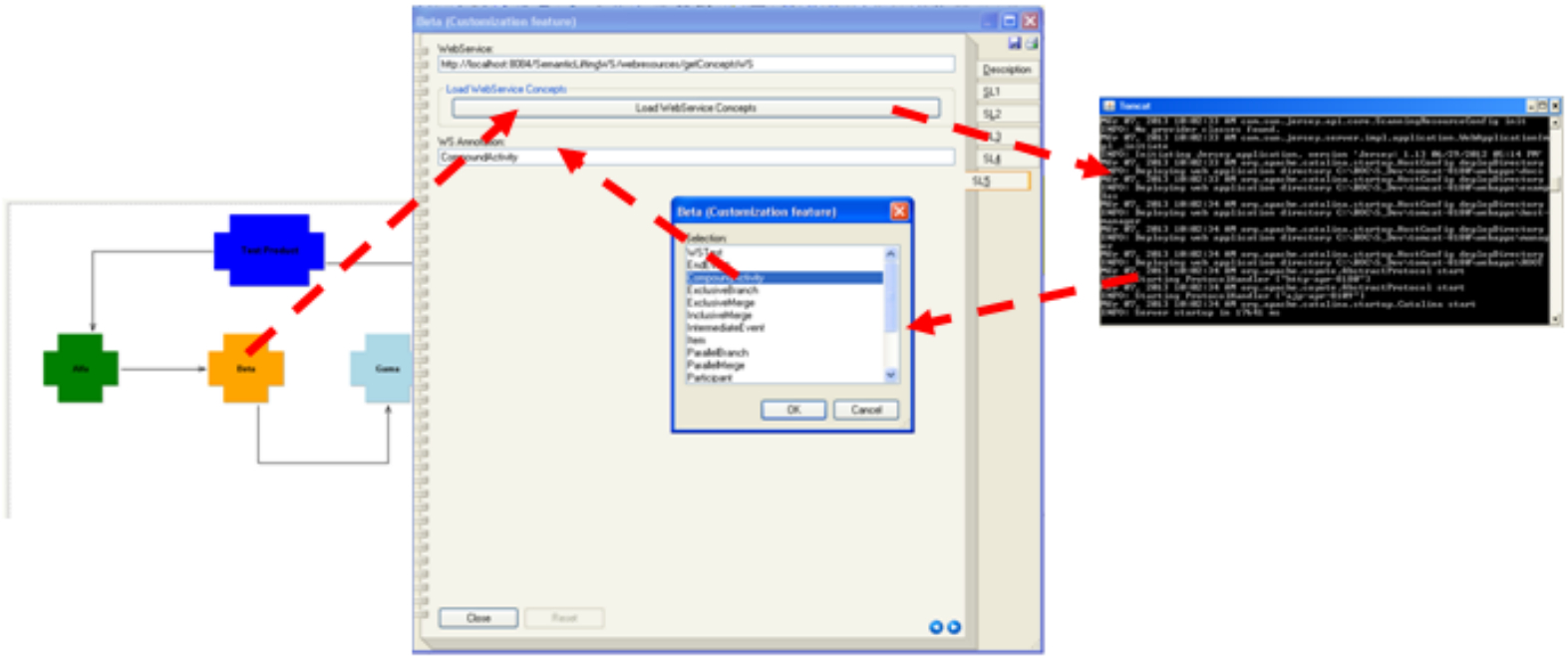
SL5 - Model to Web Service with AdoScript: Code

```
ITEM "Load WebService Concepts"
SET my_objid: (STR objid)
SET myobjid: (VAL my_objid)
CC "Modeling" GET_ACT_MODEL
SET myModelID: (modelid)
CC "Core" GET_ATTR_VAL objid:(myobjid) attrname:("WebService")
SET txtWS:(val)
SYSTEM ("C:\\SemLift.bat \""+txtWS+"\"")
SET txt:("C:\\getConceptsWS")
CC "AdoScript" FREAD file: (txt)
SET text_new: (text)
CC "AdoScript" LISTBOX entries: (text_new) toksep: "@"
IF (endbutton = ("ok"))
{
            CC "Core" GET_CLASS_ID objid:(myobjid)
            SET myclassid: (classid)
            CC "Core" GET_ATTR_ID classid: (myclassid) attrname: "WS Annotation"
            SET myattrid: (attrid)
            FOR mySelectedID in: (selection)
            {
             CC "Core" SET_ATTR_VAL objid:(myobjid) attrid:(myattrid) val:(mySelectedID)
             CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
             EXIT
            }
            CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
}
ELSE
{
    CC "AdoScript" INFOBOX ("You cancelled the dialog without selecting detailed annotation!")
    CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
}
CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
```

Result:

SL5 - Model to Web Service with AdoScript: Scenario

This scenario provides a functionality to annotate desired modelling object with a set of concepts available from a WebService API (REST), thus enabling dynamic availability of the required concepts



SL6 - Graphical Annotation within Model

Prerequisites for SL-6 and SL7

- **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
- **Apache Tomcat 7.x and Java JDK 1.7**
  download at http://tomcat.apache.org and http://www.oracle.com/technetwork/java/index.html

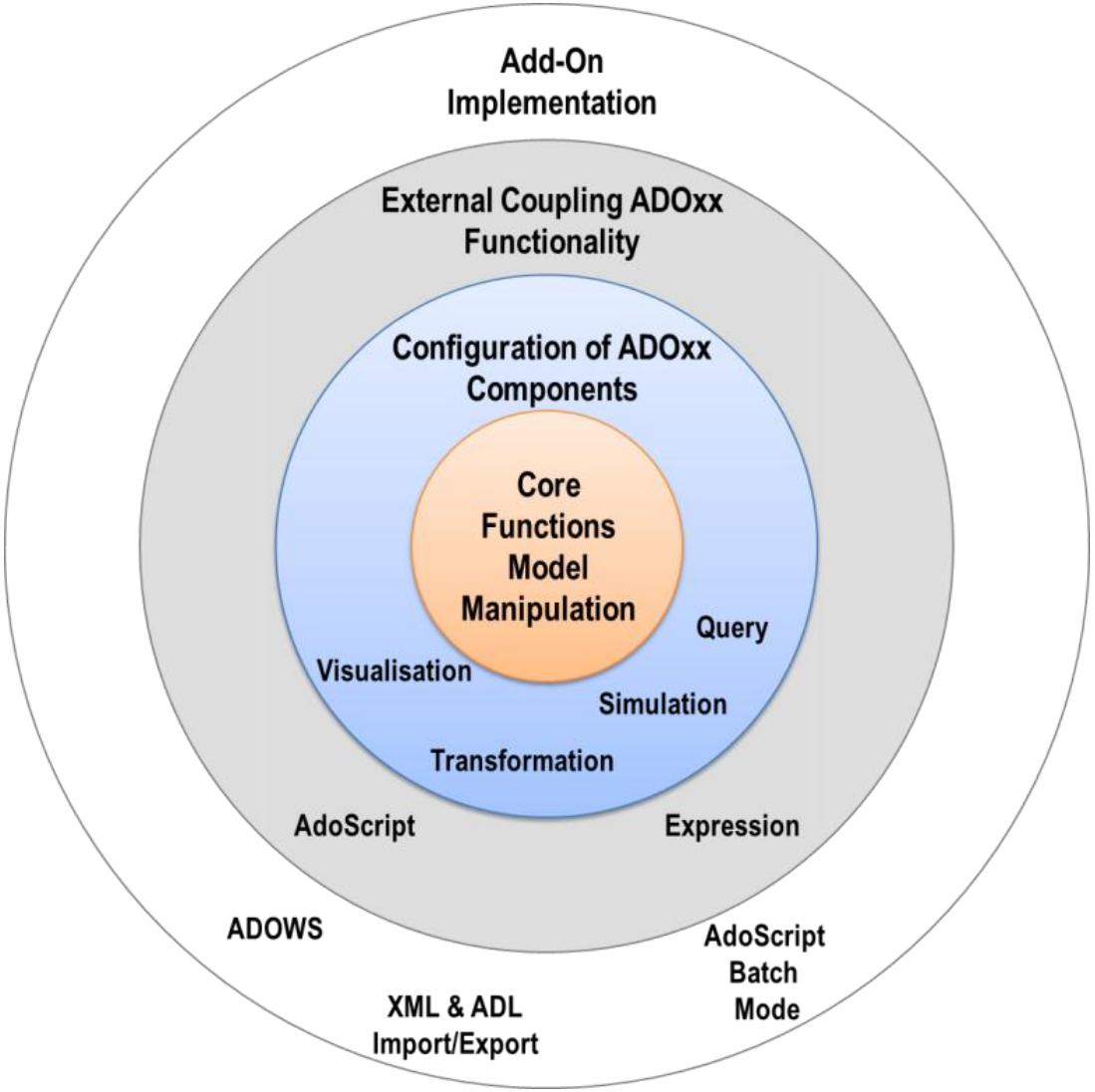
Prerequisites for Semantic Lifting

Setup

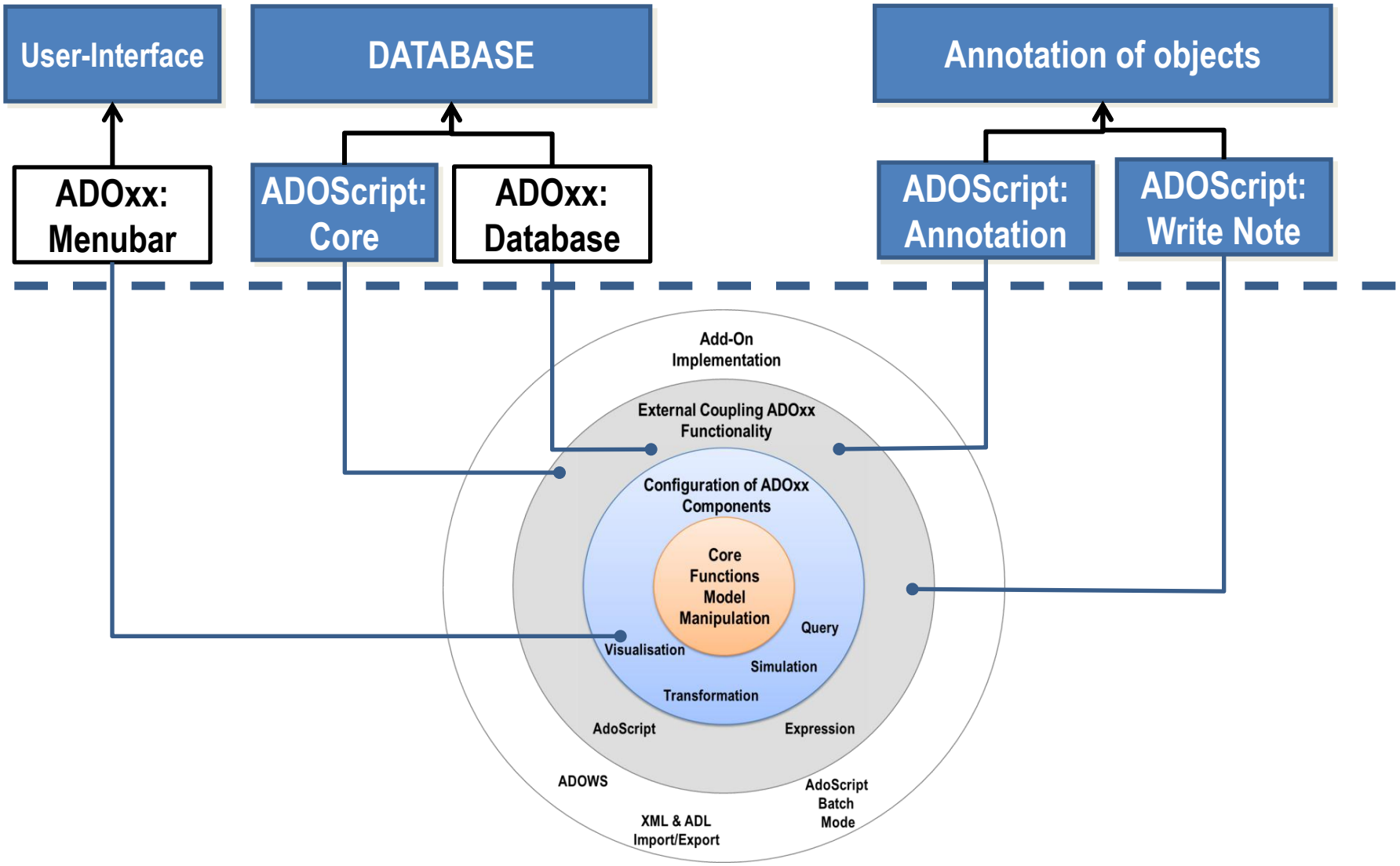
- After installing the required prerequisites:
  - Deploy SemanticLiftingWS1 to WS4.war to Tomcat
  - Store the SemLift.bat file at C:\ drive*.
  - Store the test.txt file at C:\ drive*.
  - If required change your installation path for the GnuWin32 Wget.
  - Check if you are using a proxy to access the internet and adapt the SemLift.bat accordingly

* If you do not have access rights to write/read from C drive then place it somewhere else and adapt the scripts accordingly

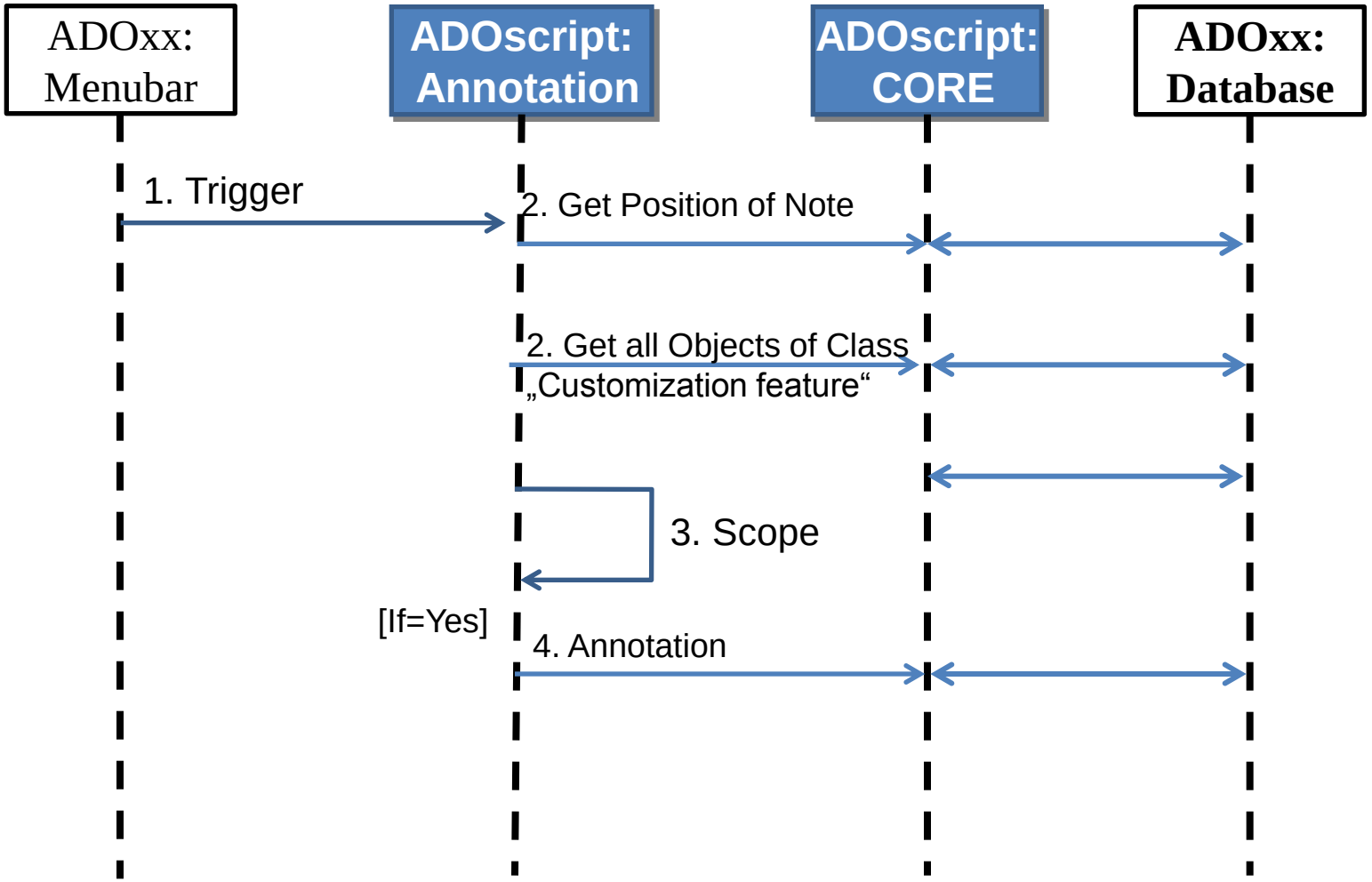
ADOxx Functionality on Meta Level



Mapping ADOxx Functionality



ADOxx Realisation Approach



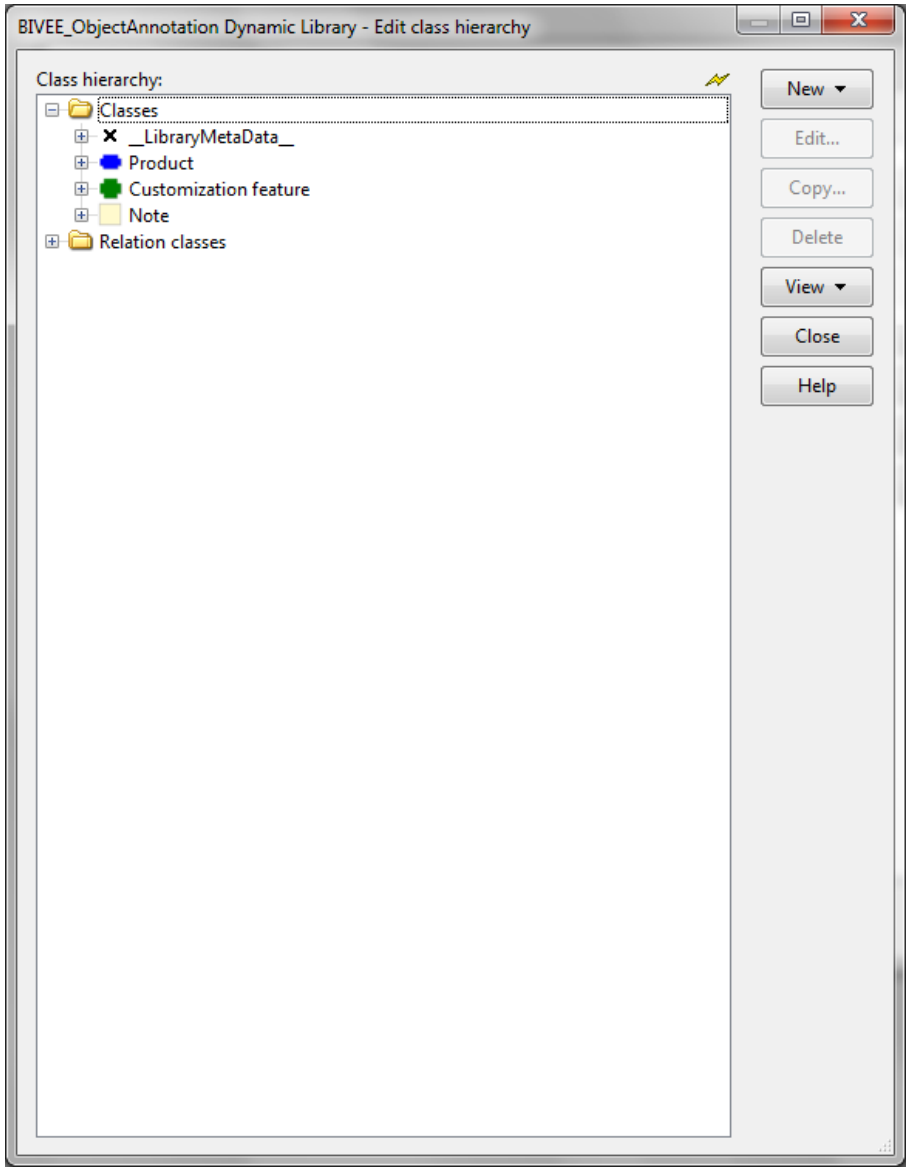
Used ADOxx Functionality

- 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

SL6 - Graphical Annotation within Model

HANDS ON: SL6 - Graphical Annotation within Model (Whiteboard)



1. Create Class
   - Product
   - Customization Feature
   - Note

2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

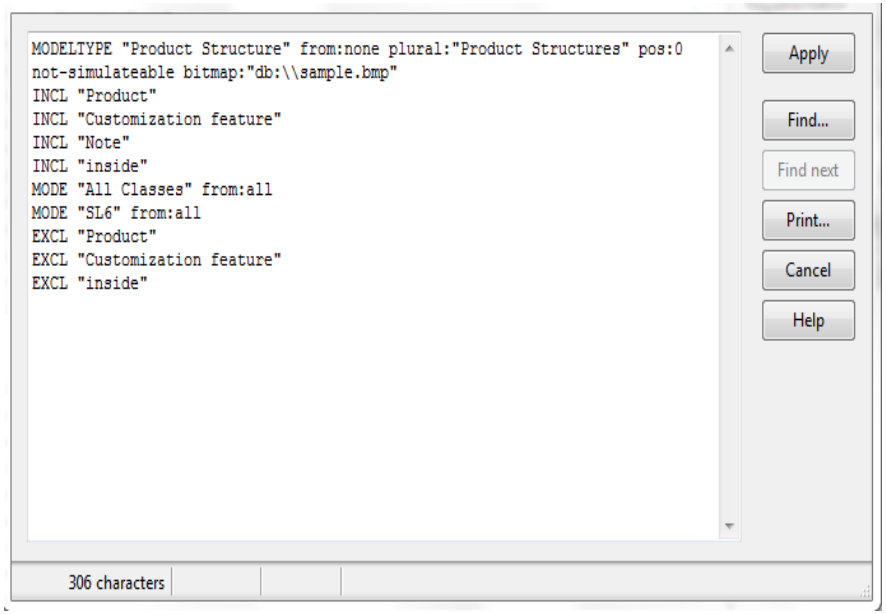
3. Define Atttrbute for class
   - <Customization Feature>:
     - "Annotation" (LONGSTRING)
     - „Type" (ENUMERATION)
   - <Note>:
     - Annotated (ENUMERATION)
     - Note (LONGSTRING)

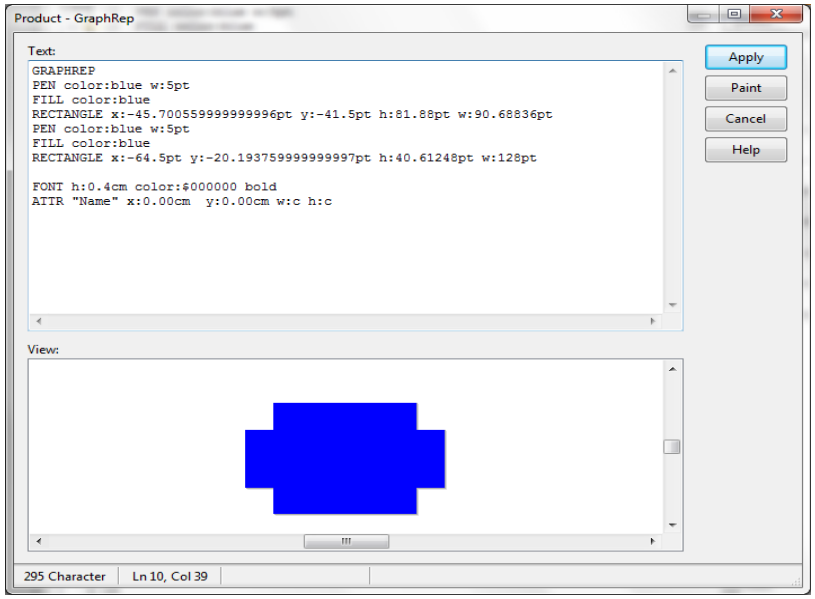
4. Define AttrRep for class "Customization Feature":

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Annotation"
ATTR "Type"
```

5. Implement Algorithm in "Add onn"
   (see slide **SL6 - Graphical Annotation within Model (Whiteboard): Code 1-3**)

HANDS ON: SL6 - Graphical Annotation within Model (Whiteboard)





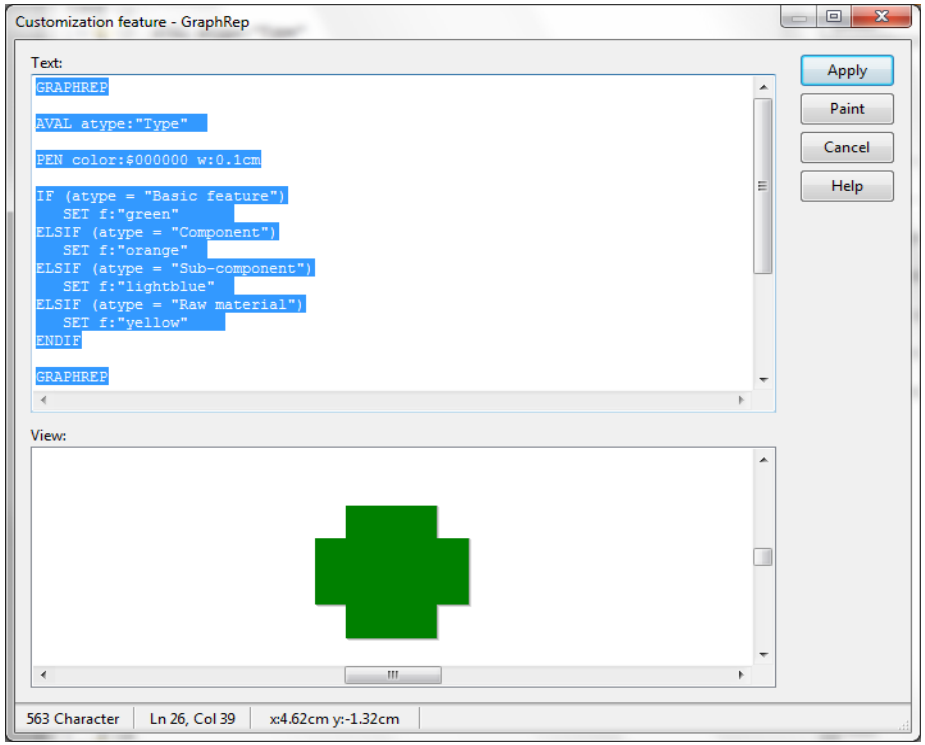
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

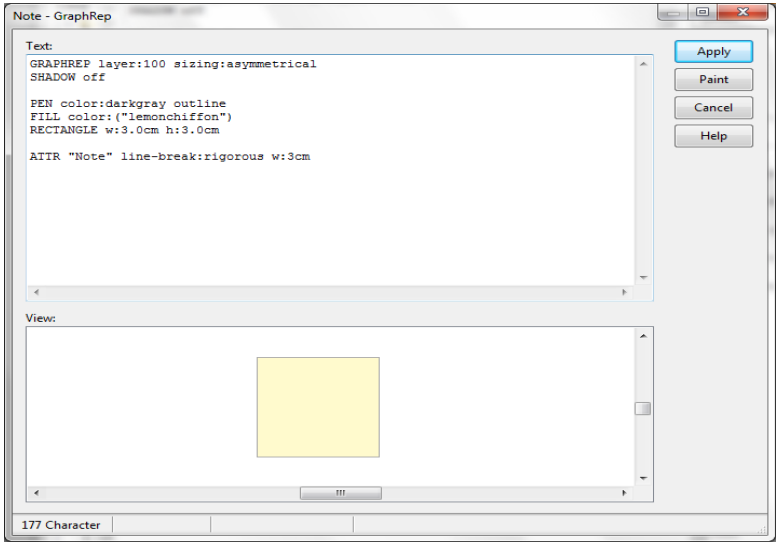
HANDS ON: SL6 - Graphical Annotation within Model (Whiteboard)



7. Define GraphRep for “Customization Feature”:

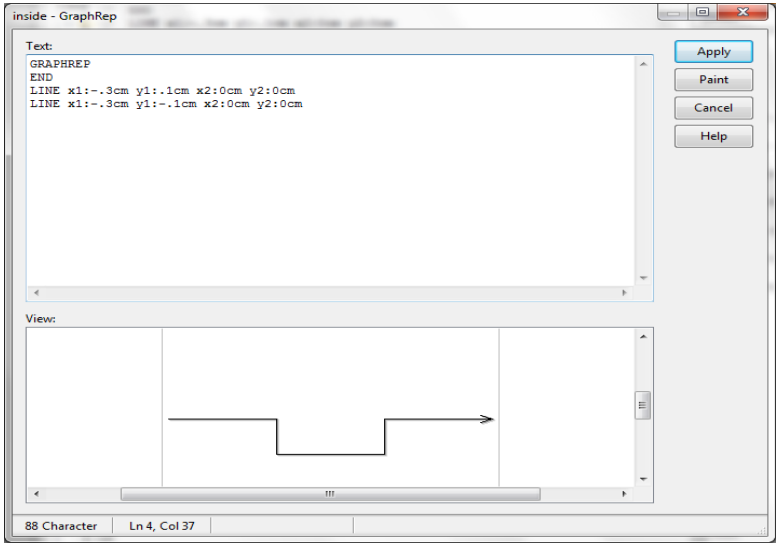
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON: SL6 - Graphical Annotation within Model (Whiteboard)



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

SL6 - Graphical Annotation within Model (Whiteboard): Code (1/3)

```
#-----------------------------------------------
ITEM "Set SL6 Annotation"
     acquisition:"Extras" modeling:"Extras" analysis:"Extras"
     simulation:"Extras" evaluation:"Extras" importexport:"Extras"
#-----------------------------------------------
CC "Modeling" GET_ACT_MODEL
CC "Core" GET_ALL_OBJS modelid: (modelid)
SET str_objids: (objids)
SET id_noteid: (-1)
FOR id_objectid in: (str_objids)
{
  CC "Core" GET_CLASS_NAME classid: (VAL(id_objectid))
IF (classname = "Note")
  {
    SET id_noteid: (VAL(id_objectid))
    CC "Core" GET_ATTR_VAL objid: (id_noteid) attrname: "Annotated"
IF ( (val) = "no")
    {
              CC "Core" GET_CLASS_ID objid: (id_noteid)
              CC "Core" GET_ATTR_ID classid: (classid) attrname: "Position"
              CC "Core" GET_ATTR_VAL objid: (id_noteid) attrid: (attrid)
              SET str_notePosition: (val)
              SET n_xnote:0
              GET_PARAM_VAL main_string:(str_notePosition) idx_param:1 param_val:n_xnote
              SET n_ynote:0
              GET_PARAM_VAL main_string:(str_notePosition) idx_param:2 param_val:n_ynote
              SET n_wnote:0
              GET_PARAM_VAL main_string:(str_notePosition) idx_param:3 param_val:n_wnote
              SET n_hnote:0
              GET_PARAM_VAL main_string:(str_notePosition) idx_param:4 param_val:n_hnote
```

SL6 - Graphical Annotation within Model (Whiteboard): Code (2/3)

```
              SET n_cxnote: (n_xnote + n_wnote/2)
              SET n_cynote: (n_ynote + n_hnote/2)
              SET id_newobjid: (-1)
              FOR id_objectid2 in: (str_objids)
              {
               CC "Core" GET_CLASS_ID objid: (VAL(id_objectid2))
               CC "Core" GET_CLASS_NAME classid: (classid)
               IF (classname = "Customization feature")
               {
                CC "Core" GET_CLASS_ID objid: (VAL(id_objectid2))
                CC "Core" GET_ATTR_ID classid: (classid) attrname: "Position"
                CC "Core" GET_ATTR_VAL objid: (VAL(id_objectid2)) attrid: (attrid)
                SET str_objPosition: (val)
                SET n_xobj:0
                GET_PARAM_VAL main_string:(str_objPosition) idx_param:1 param_val:n_xobj
                SET n_yobj:0
                GET_PARAM_VAL main_string:(str_objPosition) idx_param:2 param_val:n_yobj
                SET n_wobj:0
                GET_PARAM_VAL main_string:(str_objPosition) idx_param:3 param_val:n_wobj
                SET n_hobj:0
                GET_PARAM_VAL main_string:(str_objPosition) idx_param:4 param_val:n_hobj
                IF ((n_cxnote > n_xobj - n_wobj/2) AND (n_cxnote < (n_xobj + n_wobj/2)) AND (n_cynote >
n_yobj - n_hobj/2) AND (n_cynote < (n_yobj + n_hobj/2)))
                 {
             SET id_newobjid: (VAL(id_objectid2))
              CC "Core" GET_CLASS_ID objid: (id_newobjid)
              CC "Core" GET_CLASS_NAME classid: (classid)
              CC "Core" GET_ATTR_VAL objid: (id_noteid) attrname:"Note"
              SET str_noteValue: (val)
```

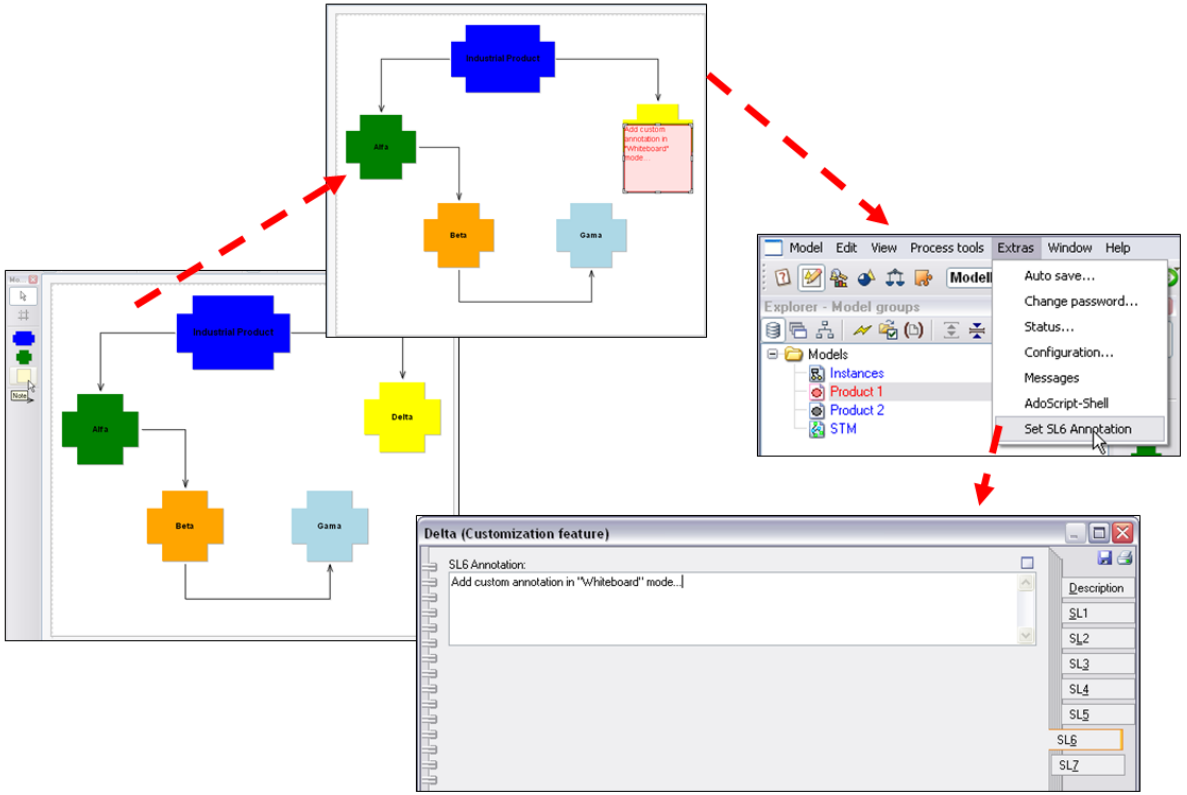
SL6 - Graphical Annotation within Model (Whiteboard): Code (3/3)

```
              CC "Core" GET_ATTR_ID classid: (classid) attrname: "SL6 Annotation"
              CC "Core" GET_ATTR_VAL objid: (id_newobjid) attrname: ("SL6 Annotation") as-string
              SET str_oldValue: (val)
              CC "Core" SET_ATTR_VAL objid: (id_newobjid) attrid: (attrid) val: ((str_oldValue) +
(str_noteValue) + "\n********\n")
              CC "Core" GET_OBJ_NAME objid: (id_newobjid)
              CC "AdoScript" INFOBOX ("Attribute \"SL6 Annotation\" changed for object \"" + objname +
"\" !")
              CC "Core" SET_ATTR_VAL objid: (id_noteid) attrname: ("Annotated") val: ("yes")
                }
               }
             }
    }
   }
}

IF (id_noteid = -1)
{
  CC "AdoScript" INFOBOX "No Note available in this model"    #if no "Note" available, display an
error message
}

PROCEDURE GET_PARAM_VAL main_string: string idx_param: integer
                                  param_val: reference
{
             SETL str_param: (token(main_string, idx_param, " " ))
     SETL str_paramValue: (token(str_param, 1, ":"))
             SETL param_val: (VAL (token(str_paramValue, 0, "c")))
}
```

Result:

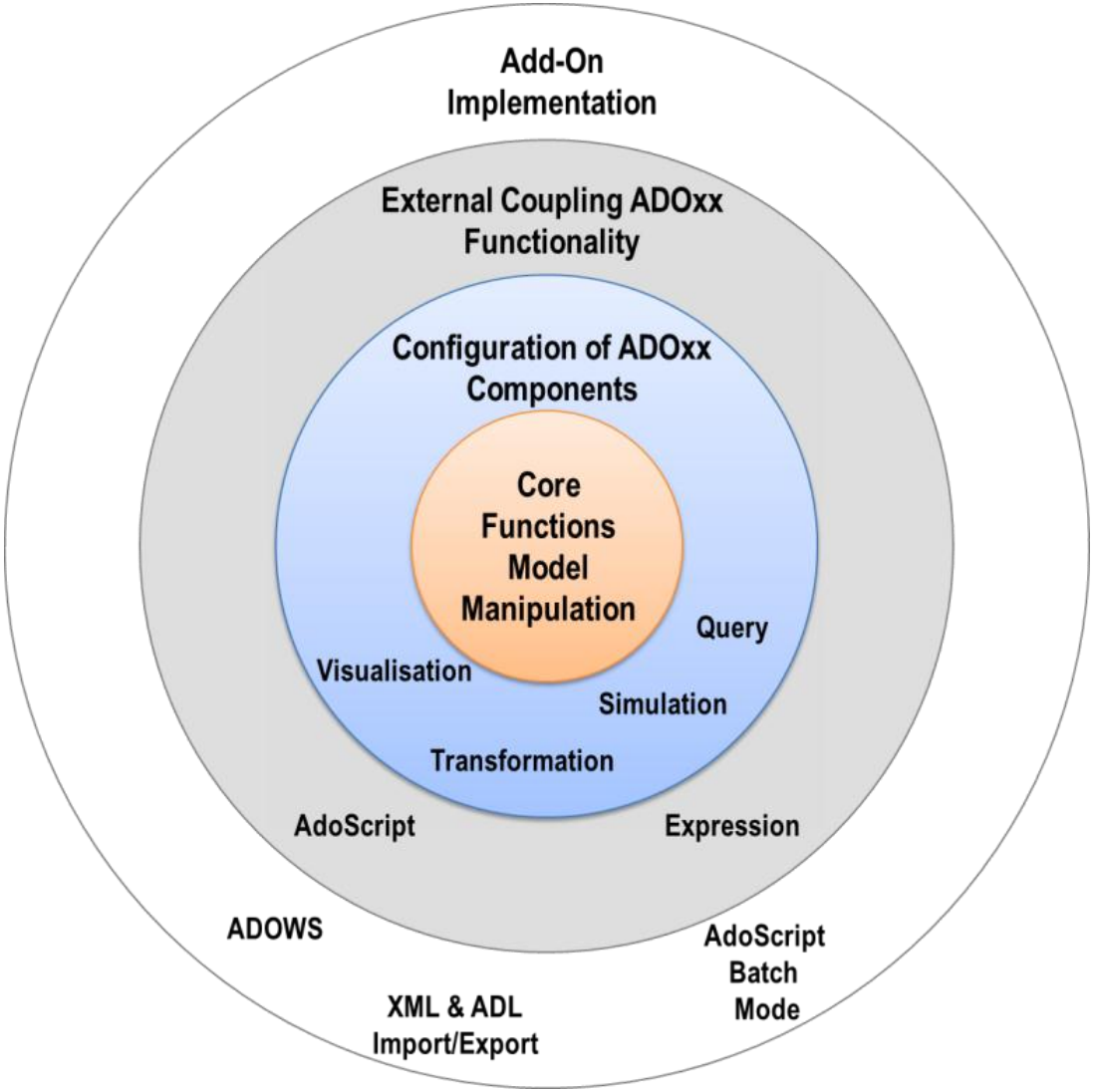
SL6 - Graphical Annotation within Model (Whiteboard): Scenario

This scenario provides a “whiteboard” functionality to annotate a desired modelling object with previously non existing concepts by using an object that functions as a post-it on a whiteboard.

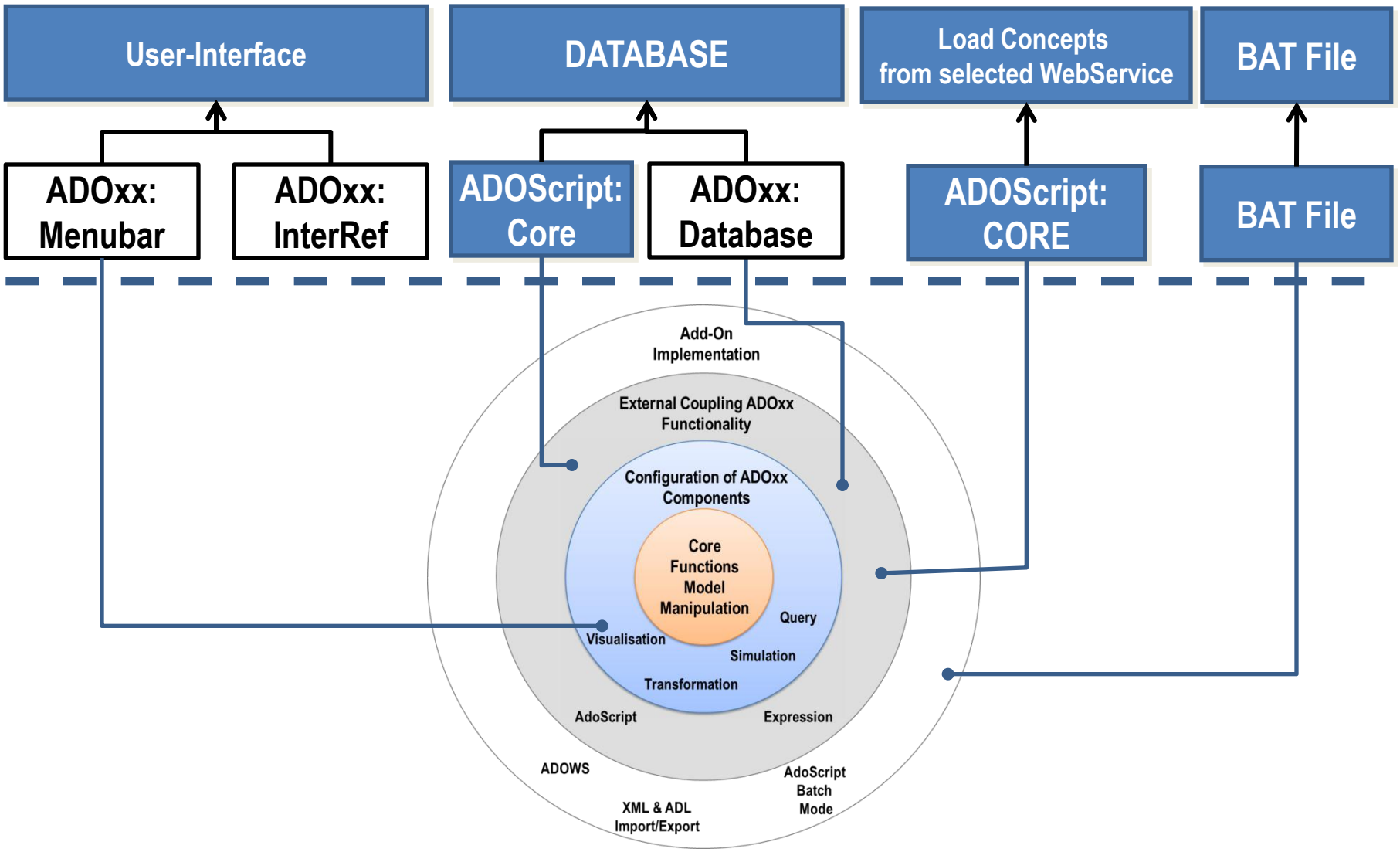


SL7 - Model to Web Service through the Semantic Transit Model

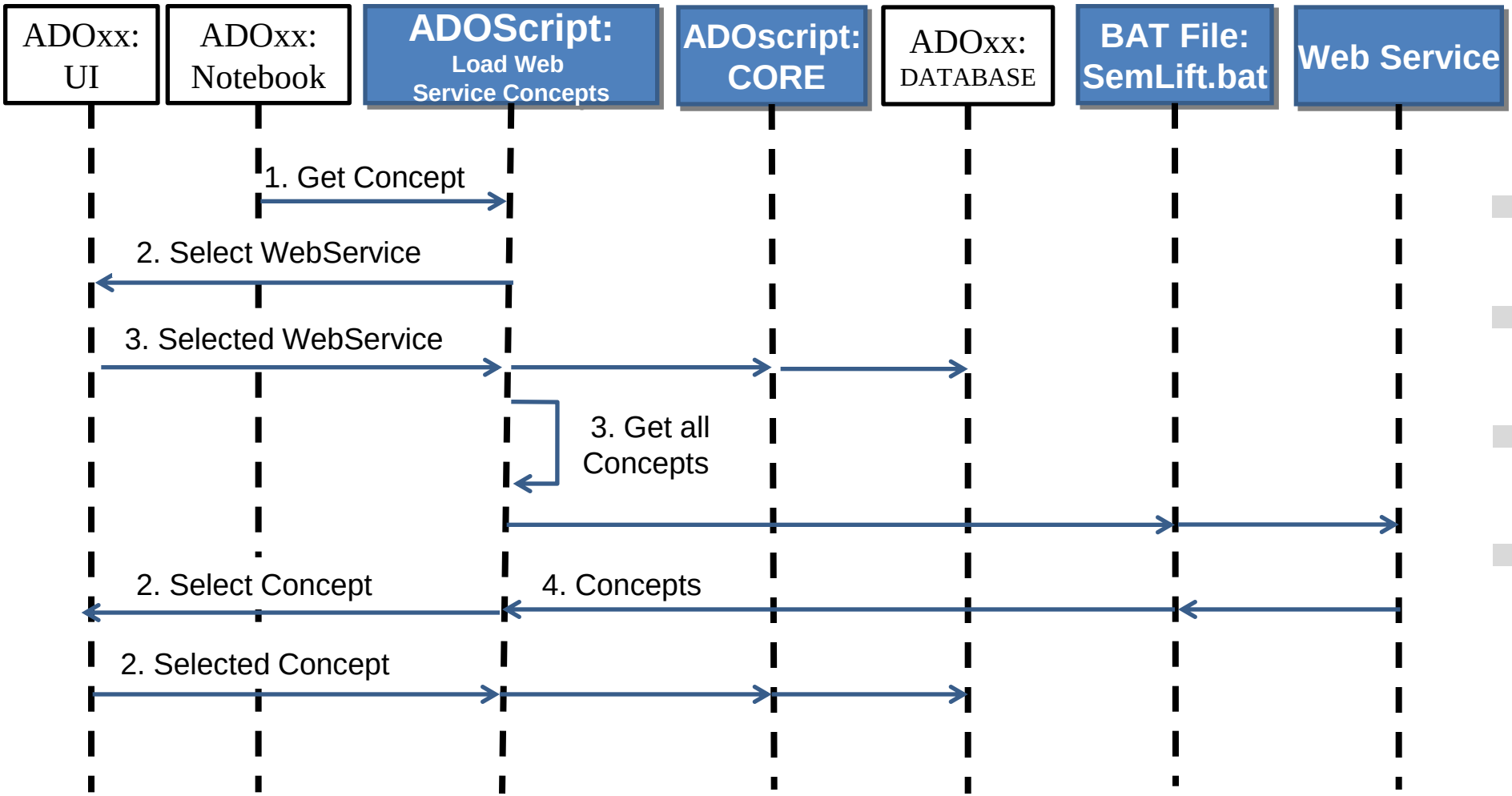
ADOxx Functionality on Meta Level



Mapping ADOxx Functionality



Description of the Algorithm



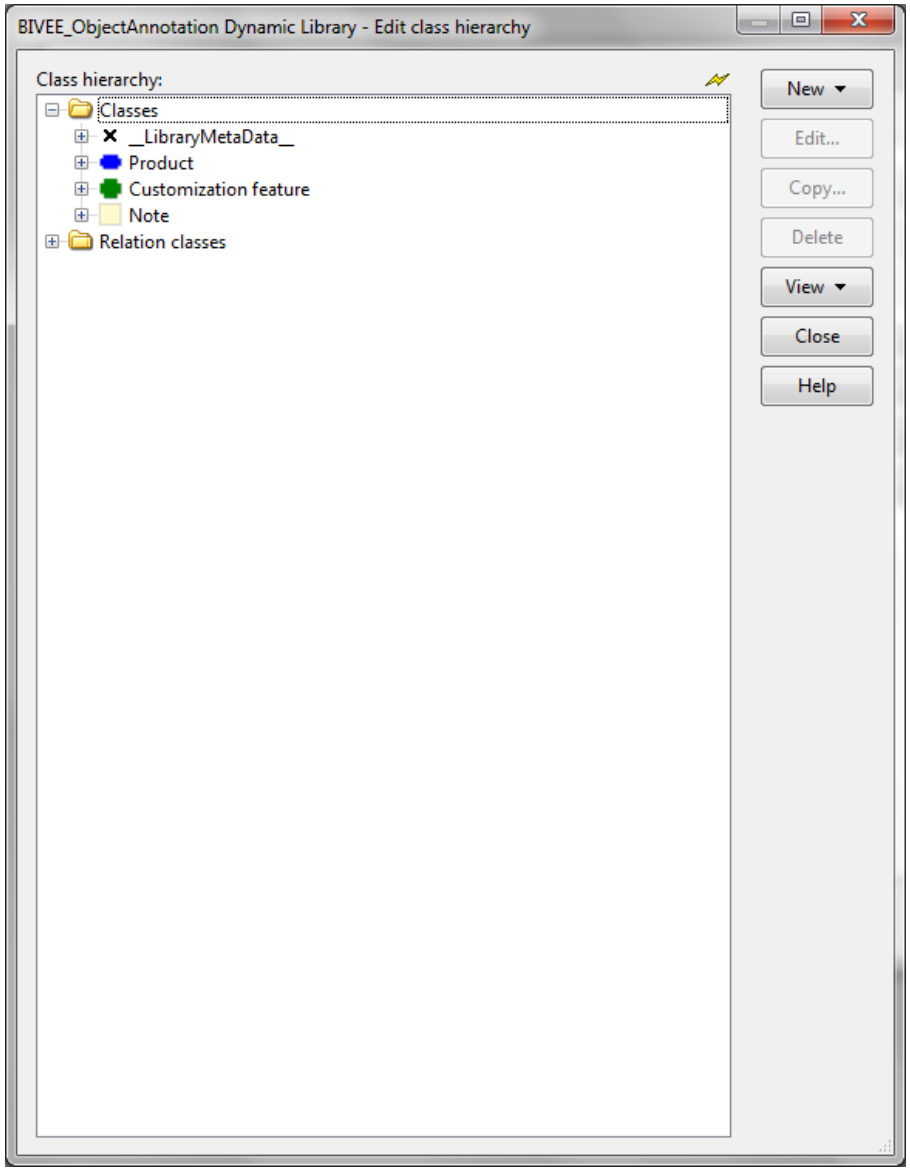
Used ADOxx Functionality

- 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

SL7 - Model to Web Service through the Semantic Transit Model

HANDS ON :SL7 - Model to Web Service through the Semantic Transit Model



1. Create Class
   - Product
   - Customization Feature
   - Note

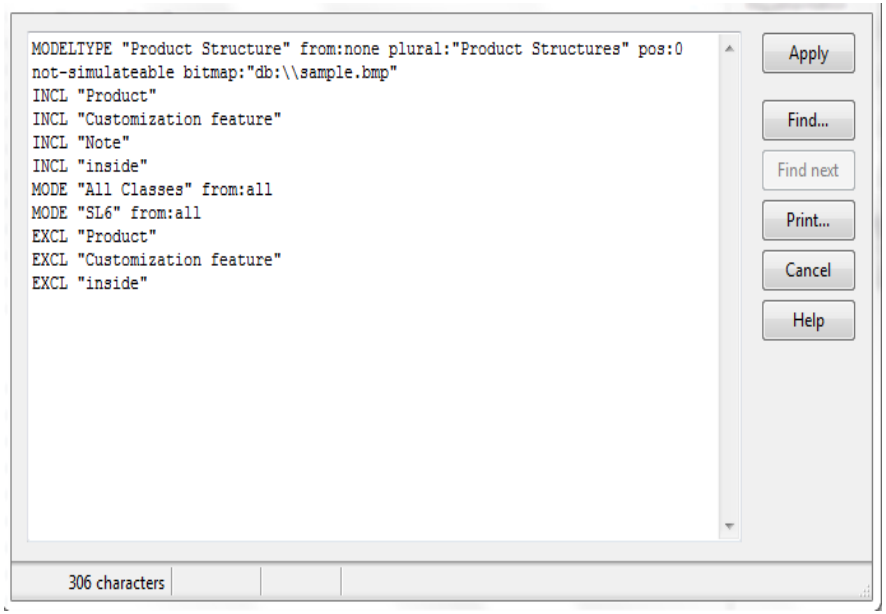
2. Create RelationClass:
   - inside:
     __D-construct__→ Customization Feature

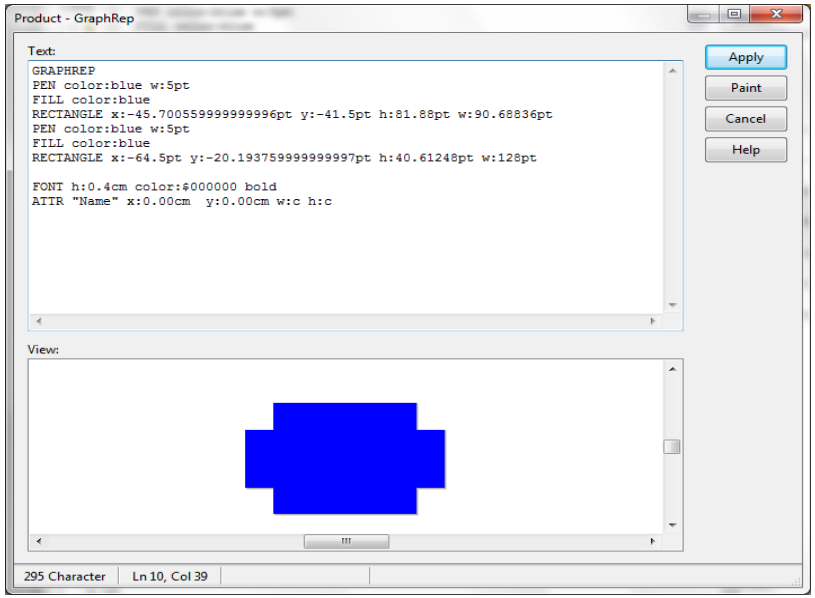
3. Define Attribute for class
   - <Customization Feature>:
     - "Get WS Concepts" (ProgramCall)
       (see slide **SL7 - Model to Web Service through the Semantic Transit Model: Code**)
     - "Semantic Transit Model - WS" (INTERREF)
     - "WS Annotation II" (STRING)
     - „Type" (ENUMERATION)

4. Define AttrRep for class "Customization Feature":

```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Get WS Concepts"
ATTR "Semantic Transit Model - WS"
ATTR "WS Annotation II"
ATTR "Type"
```

HANDS ON :SL7 - Model to Web Service through the Semantic Transit Model





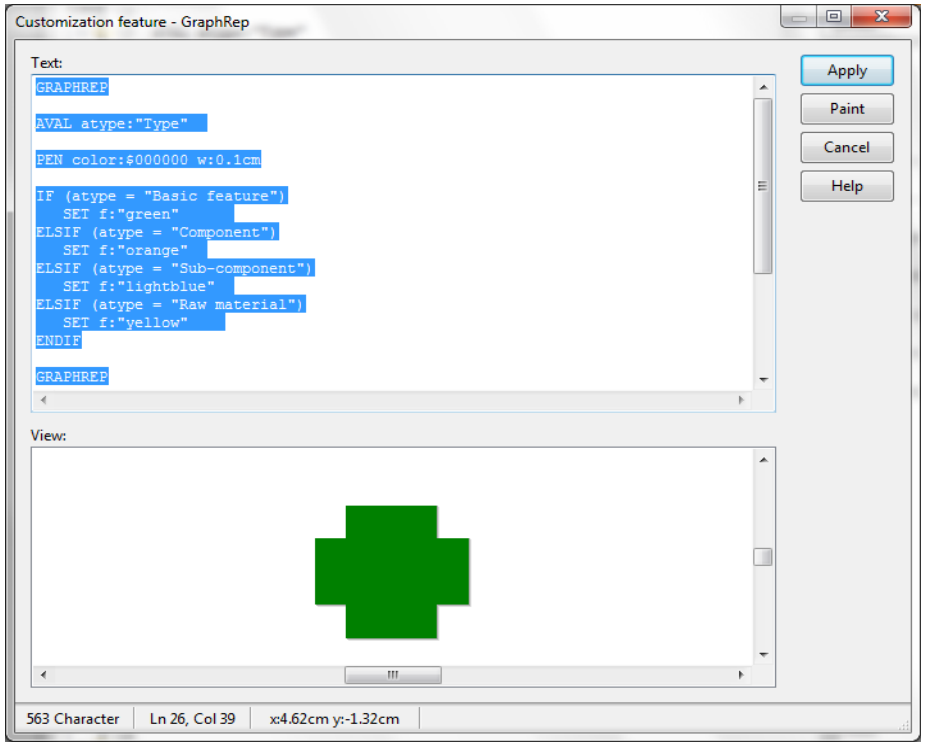
5. Define Modeltype:

```
MODELTYPE "Product Structure" from:none
plural:"Product Structures" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"
INCL "Product"
INCL "Customization feature"
INCL "Note"
INCL "inside"
MODE "All Classes" from:all
MODE "SL6" from:all
EXCL "Product"
EXCL "Customization feature"
EXCL "inside"
```

6. Define GraphRep for “Product”

```
GRAPHREP
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-45.700559999999996pt y:-41.5pt
h:81.88pt w:90.68836pt
PEN color:blue w:5pt
FILL color:blue
RECTANGLE x:-64.5pt y:-20.193759999999997pt
h:40.61248pt w:128pt

FONT h:0.4cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

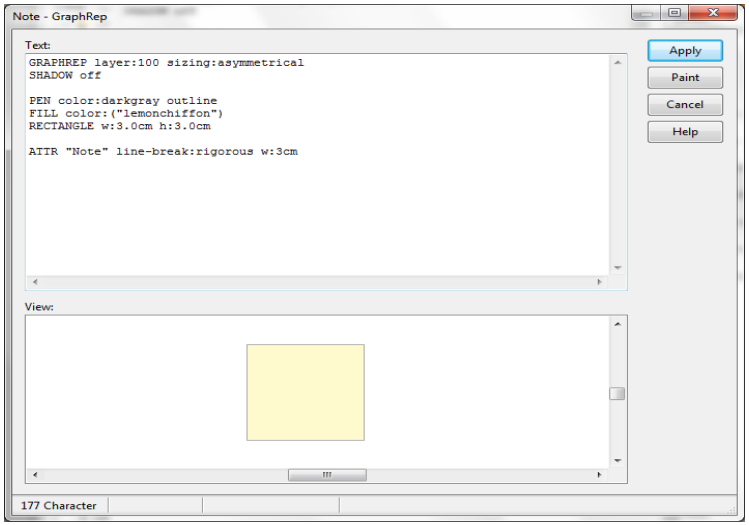
HANDS ON :SL7 - Model to Web Service through the Semantic Transit Model



7. Define GraphRep for “Customization Feature”:

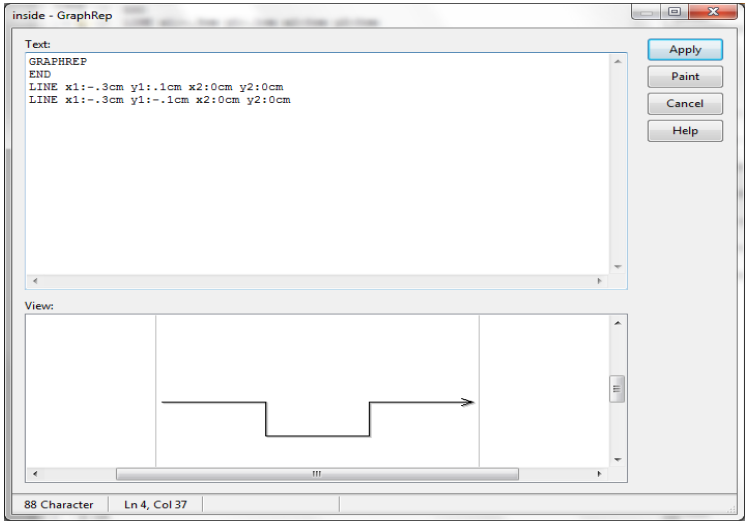
```
GRAPHREP
AVAL atype:"Type"
PEN color:$000000 w:0.1cm
IF (atype = "Basic feature")
  SET f:"green"
ELSIF (atype = "Component")
  SET f:"orange"
ELSIF (atype = "Sub-component")
  SET f:"lightblue"
ELSIF (atype = "Raw material")
  SET f:"yellow"
ENDIF

GRAPHREP
PEN color:(f)
FILL color:(f)
RECTANGLE x:-27.19999999999999pt y:-41.5pt h:81.88pt
w:52.688pt
PEN color:(f)
FILL color:(f)
RECTANGLE x:-45pt y:-21.193759999999997pt h:40.61248pt
w:89pt

FONT h:0.35cm color:$000000 bold
ATTR "Name" x:0.00cm  y:0.00cm w:c h:c
```

HANDS ON :SL7 - Model to Web Service through the Semantic Transit Model



8. Define GraphRep “Note”

GRAPHREP layer:100
sizing:asymmetrical
SHADOW off
PEN color:darkgray outline
FILL color:("lemonchiffon")
RECTANGLE w:3.0cm h:3.0cm
ATTR "Note" line-break:rigorous w:3cm



9. Define GraphRep “inside”

GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm

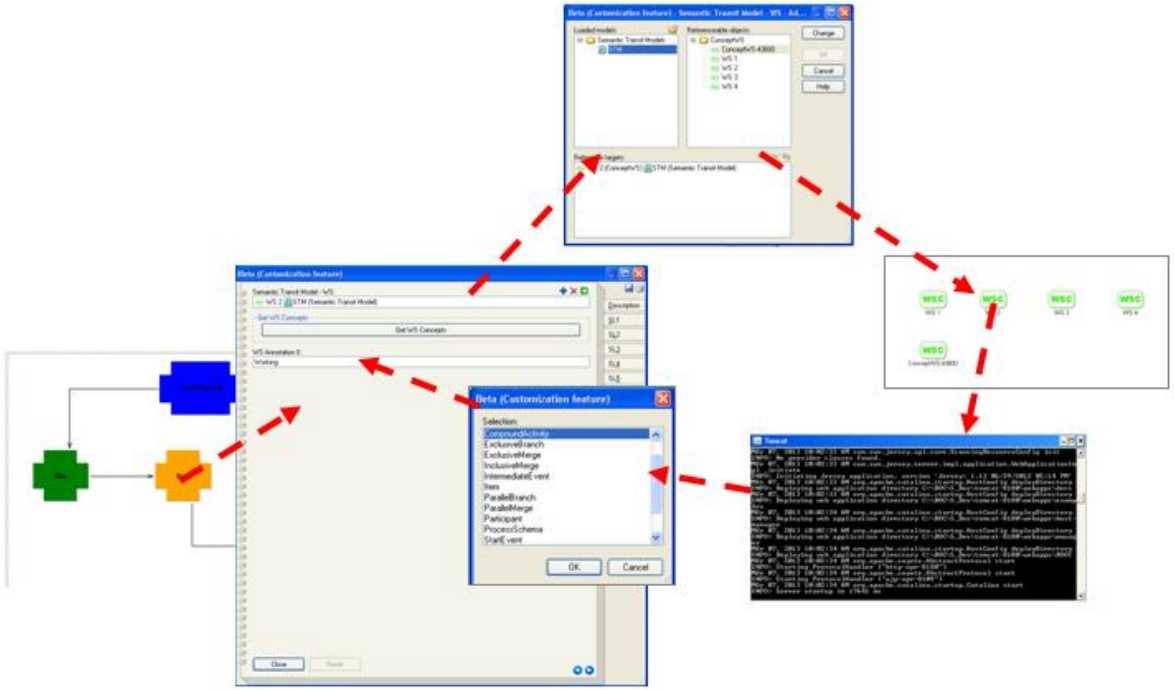
SL7 - Model to Web Service through the Semantic Transit Model: Code

```
ITEM "Get WS Concepts"
SET my_objid: (STR objid)
SET myobjid: (VAL my_objid)
CC "Modeling" GET_ACT_MODEL
SET myModelID: (modelid)
CC "Core" GET_CLASS_ID objid: (myobjid)
CC "Core" GET_ATTR_ID classid: (classid) attrname: ("Semantic Transit Model - WS")
CC "Core" GET_INTERREF objid: (myobjid) attrid: (attrid) index: 0
CC "Core" GET_ATTR_VAL objid:(tobjid) attrname:("WS Endpoint")
SET txtWS:(val)
SYSTEM ("C:\\SemLift.bat \""+txtWS+"\"")
SET txt:("C:\\getConceptsWS")
CC "AdoScript" FREAD file: (txt)
SET text_new: (text)
CC "AdoScript" LISTBOX entries: (text_new) toksep: "@"
IF (endbutton = ("ok")) {
            CC "Core" GET_CLASS_ID objid:(myobjid)
            SET myclassid: (classid)
            CC "Core" GET_ATTR_ID classid: (myclassid) attrname: "WS Annotation II"
            SET myattrid: (attrid)
            FOR mySelectedID in: (selection)
            {
             CC "Core" SET_ATTR_VAL objid:(myobjid) attrid:(myattrid) val:(mySelectedID)
             CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
             EXIT
            }
            CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
}
ELSE  {
    CC "AdoScript" INFOBOX ("You cancelled the dialog without selecting detailed annotation!")
     CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
}
CC "AdoScript" FILE_DELETE file:("C:\\getConceptsWS")
```

Result:

SL7 - Model to Web Service through the Semantic Transit Model

This scenario provides a functionality to annotate desired modelling object with a set of concepts available as an external resource which are accessed over a Web Service API. These concepts are then accessible through the Semantic Transit Models and can be used as annotation concepts.



Further Questions?

tutorial@adoxx.org

102