ADOxx Tutorial

MODELLING METHOD INTEGRATION: SEMANTIC LIFTING APPROACH

Scenario: Supply Chain Management

Agenda

www.omilab.org

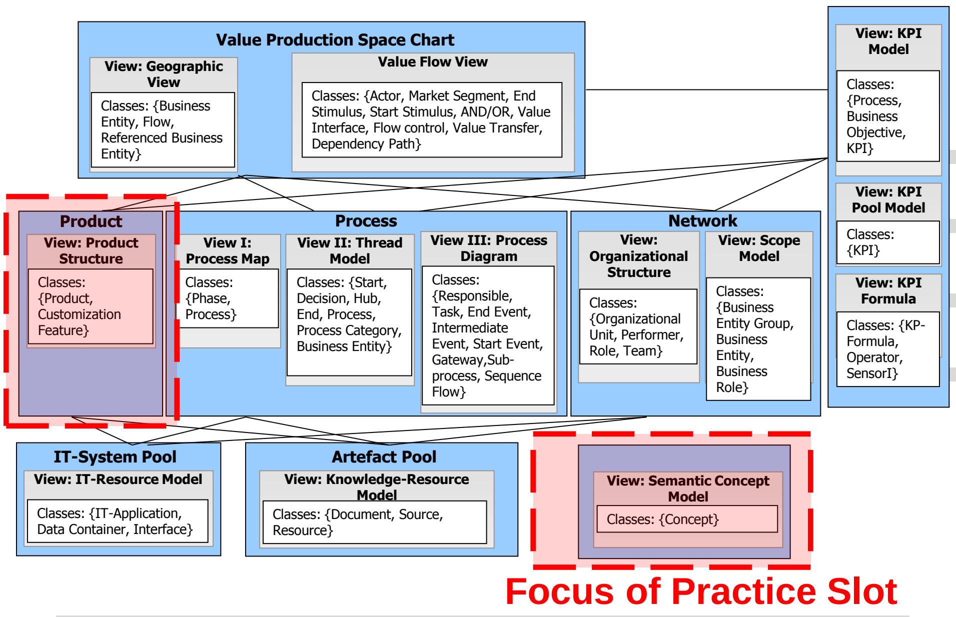
- **SCM Scenario**
- Hybrid Modelling Approach Introduction
- Semantic Lifting
- Hands On Sessions:
  - Prerequisites
  - Definition of Model Structure on ADOxx
  - Semantic Lifting
- Q&A

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


Value Production Space Chart
View: Geographic View
Classes: {Business Entity, Flow, Referenced Business Entity}
Value Flow View
Classes: {Actor, Market Segment, End Stimulus, Start Stimulus, AND/OR, Value Interface, Flow control, Value Transfer, Dependency Path}
View: KPI Model
Classes: {Process, Business Objective, KPI}
View: KPI Pool Model
Classes: {KPI}
View: KPI Formula
Classes: {KP-Formula, Operator, SensorI}
Product
View: Product Structure
Classes: {Product, Customization Feature}
Process
View I: Process Map
Classes: {Phase, Process}
View II: Thread Model
Classes: {Start, Decision, Hub, End, Process, Process Category, Business Entity}
View III: Process Diagram
Classes: {Responsible, Task, End Event, Intermediate Event, Start Event, Gateway,Sub-process, Sequence Flow}
Network
View: Organizational Structure
Classes: {Organizational Unit, Performer, Role, Team}
View: Scope Model
Classes: {Business Entity Group, Business Entity, Business Role}
IT-System Pool
View: IT-Resource Model
Classes: {IT-Application, Data Container, Interface}
Artefact Pool
View: Knowledge-Resource Model
Classes: {Document, Source, Resource}
View: Semantic Concept Model
Classes: {Concept}
Focus of Practice Slot

Supply Chain Management Scenario

Goals

1. Define Model Structure using ADOxx for describing Product Model
2. Apply Semantic Lifting approach to extend the Product Model with an external Semantic Meta Model
3. Implement functionality to apply Semantic Lifting:
   1. Using an ADOxx Interref
   2. Using Direct Textual Annotation within Model Object
   3. Using Semantic Transit Model (simple and using WebService Interface)
   4. Using Model Objects as a Whiteboard

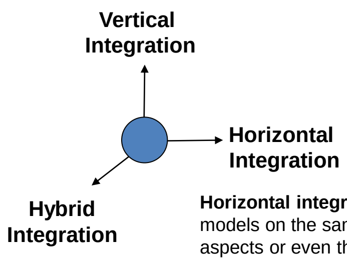
Agenda

www.omilab.org

- SCM Scenario
- **<u>Hybrid Modelling Approach Introduction</u>**
- Semantic Lifting
- Hands On Sessions:
  - Prerequisites
  - Definition of Model Structure on ADOxx
  - Semantic Lifting
- Q&A

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.

Agenda

www.omilab.org

- SCM Scenario
- Hybrid Modelling Approach Introduction
- **Semantic Lifting**
- Hands On Sessions:
  - Prerequisites
  - Definition of Model Structure on ADOxx
  - Semantic Lifting
- Q&A

How can we realize Semantic Lifting (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.

How can we realize Semantic Lifting (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.

Realization of the Semantic Lifting in the Hands On Session

Scenarios

- SL1 - Model to Model Instance with Interref
- SL2 - Textual Annotation within Model
- 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

Agenda

www.omilab.org

- SCM Scenario
- Hybrid Modelling Approach Introduction
- Semantic Lifting
- **Hands On Sessions:**
  - **Prerequisites**
  - Definition of Model Structure on ADOxx
  - Semantic Lifting
- Q&A

Prerequisites for Semantic Lifting

- **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

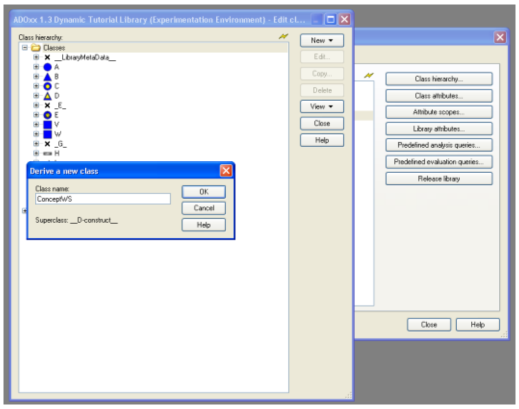
Agenda

OMiLAB®
www.omilab.org

- SCM Scenario
- Hybrid Modelling Approach Introduction
- Semantic Lifting
- **Hands On Sessions:**
  - Prerequisites
  - **Definition of Model Structure on ADOxx**
  - Semantic Lifting
- Q&A

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types


Derive a new class

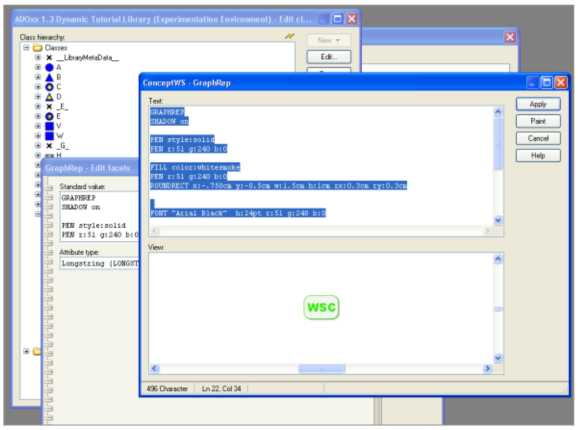
Class name: ConceptWS

Superclass: __D-construct__

1. Open Library Management
2. Define a new Classes named "Product", "Customization feature", "Concept", "ConceptWS", "Instance", "Root concept" & "Note" in the "Dynamic Tutorial Library"

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types


ConceptWS - GraphRep

Text:

```
GRAPHREP
SHADOW on

PEN style:solid
PEN r:51 g:240 b:0

FILL color:whitesmoke
PEN r:51 g:240 b:0
ROUNDRECT x:-.750cm y:-0.5cm w:1.5cm h:1cm rx:0.3cm ry:0.3cm


FONT "Arial Black"  h:24pt r:51 g:240 b:0
```

GraphRep - Edit facets

Standard value:

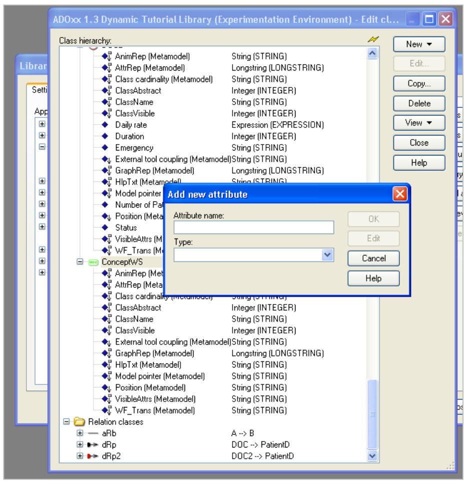
```
GRAPHREP
SHADOW on

PEN style:solid
PEN r:51 g:240 b:0
```

Attribute type: Longstring (LONGST

1. Select GraphRep Attributes of the created classes, and
2. write GraphRep code to visualise the classes

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

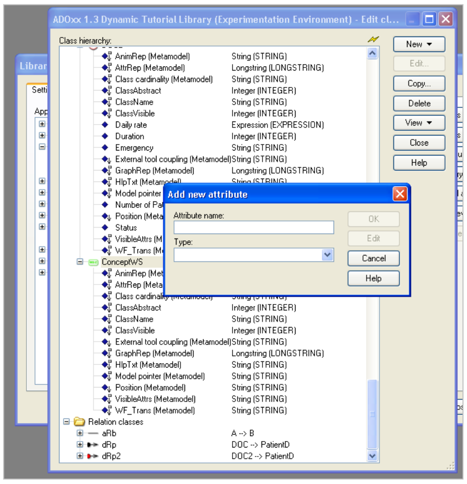


```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                  String (STRING)
    AttrRep (Metamodel)                  Longstring (LONGSTRING)
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    Daily rate                           Expression (EXPRESSION)
    Duration                             Integer (INTEGER)
    Emergency                            String (STRING)
    External tool coupling (Metamodel)   String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    External tool coupling (Metamodel)   String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer (Metamodel)            String (STRING)
    Position (Metamodel)                 String (STRING)
    VisibleAttrs (Metamodel)             String (STRING)
    WF_Trans (Metamodel)                 String (STRING)
Relation classes
  aRb                                    A --> B
  dRp                                    DOC --> PatientD
  dRp2                                   DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “Product”
2. Select New Attribute
3. Create following Attributes:
   1. Name (STRING)
   2. Description (STRING)
   3. Cost (DOUBLE)
   4. Price (DOUBLE)
   5. Status (ENUMERATION)
   6. Notify (STRING)
   7. Send Mobile Notification (PROGRAMCALL)

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types



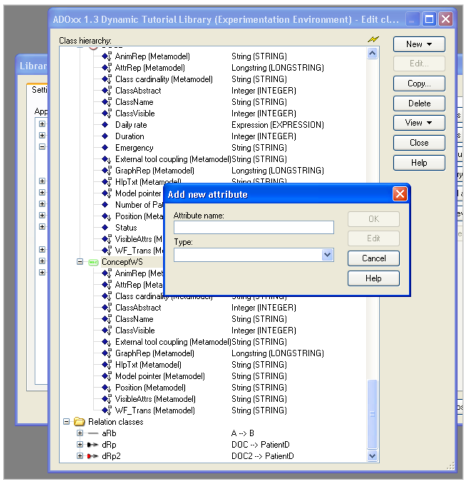
```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                  String (STRING)
    AttrRep (Metamodel)                  Longstring (LONGSTRING)
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    Daily rate                           Expression (EXPRESSION)
    Duration                             Integer (INTEGER)
    Emergency                            String (STRING)
    External tool coupling (Metamodel)   String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    External tool coupling (Metamodel)   String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer (Metamodel)            String (STRING)
    Position (Metamodel)                 String (STRING)
    VisibleAttrs (Metamodel)             String (STRING)
    WF_Trans (Metamodel)                 String (STRING)
Relation classes
  aRb                                    A --> B
  dRp                                    DOC --> PatientD
  dRp2                                   DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “Customization feature”
2. Select New Attribute
3. Create following Attributes:
   1. Name (STRING)
   2. Description (STRING)
   3. Type (ENUMERATION)
   4. Quantity (DOUBLE)
   5. Measurement (ENUMERATION)
   6. Cost (DOUBLE)
   7. Price (DOUBLE)
   8. Is a product (INTERREF)
   9. Annotation (INTERREF)
   10. SL2 Annotation I (ENUMERATIONLIST)
   11. SL2 Annotation II (STRING)

1/2

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types



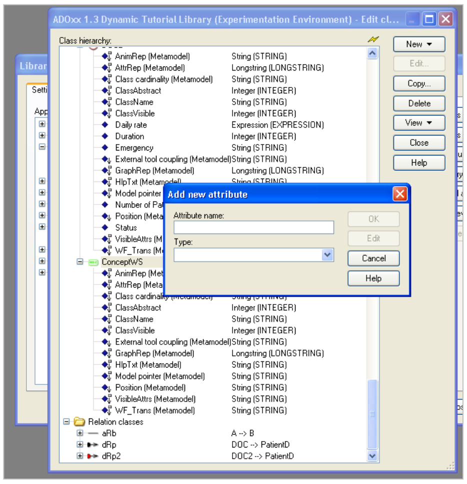
```
AnimRep (Metamodel)                  String (STRING)
AttrRep (Metamodel)                  Longstring (LONGSTRING)
Class cardinality (Metamodel)        String (STRING)
ClassAbstract                        Integer (INTEGER)
ClassName                            String (STRING)
ClassVisible                         Integer (INTEGER)
Daily rate                           Expression (EXPRESSION)
Duration                             Integer (INTEGER)
Emergency                            String (STRING)
External tool coupling (Metamodel)   String (STRING)
GraphRep (Metamodel)                 Longstring (LONGSTRING)
HlpTxt (Metamodel)                   String (STRING)
Model pointer
Number of Pa
Position (Meta
Status
VisibleAttrs (M
WF_Trans (M
ConceptWS
  AnimRep (Met
  AttrRep (Meta
  Class cardinality (Metamodel)
  ClassAbstract                      Integer (INTEGER)
  ClassName                          String (STRING)
  ClassVisible                       Integer (INTEGER)
  External tool coupling (Metamodel) String (STRING)
  GraphRep (Metamodel)               Longstring (LONGSTRING)
  HlpTxt (Metamodel)                 String (STRING)
  Model pointer (Metamodel)          String (STRING)
  Position (Metamodel)               String (STRING)
  VisibleAttrs (Metamodel)           String (STRING)
  WF_Trans (Metamodel)               String (STRING)
Relation classes
  aRb                                A --> B
  dRp                                DOC --> PatientD
  dRp2                               DOC2 --> PatientD
```

12. Semantic Transit Model (INTERREF)
13. Select Concept (PROGRAMCALL)
14. Detailed Annotation (INTERREF)
15. Text File (STRING)
16. Load Concepts (PROGRAMCALL)
17. Detailed Annotation 2 (STRING)
18. WebService (STRING)
19. Load WebService Concepts (PROGRAMCALL)
20. WS Annotation (STRING)
21. SL6 Annotation (STRING)
22. Semantic Transit Model - WS (INTERREF)
23. WS Annotation II (STRING)

2/2

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

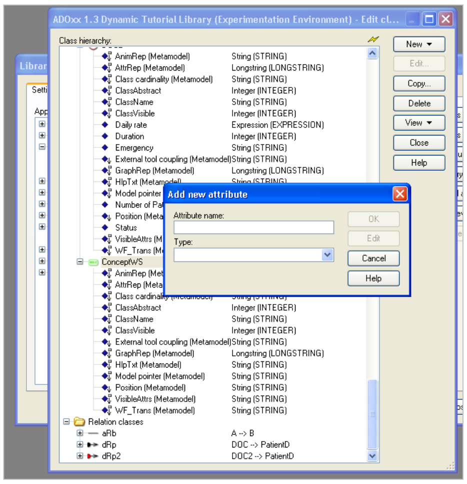


```
AnimRep (Metamodel)                  String (STRING)
AttrRep (Metamodel)                  Longstring (LONGSTRING)
Class cardinality (Metamodel)        String (STRING)
ClassAbstract                        Integer (INTEGER)
ClassName                            String (STRING)
ClassVisible                         Integer (INTEGER)
Daily rate                           Expression (EXPRESSION)
Duration                             Integer (INTEGER)
Emergency                            String (STRING)
External tool coupling (Metamodel)   String (STRING)
GraphRep (Metamodel)                 Longstring (LONGSTRING)
HlpTxt (Metamodel)                   String (STRING)
Model pointer
Number of Pa
Position (Meta
Status
VisibleAttrs (M
WF_Trans (M
ConceptWS
  AnimRep (Met
  AttrRep (Meta
  Class cardinality (Metamodel)
  ClassAbstract                      Integer (INTEGER)
  ClassName                          String (STRING)
  ClassVisible                       Integer (INTEGER)
  External tool coupling (Metamodel) String (STRING)
  GraphRep (Metamodel)               Longstring (LONGSTRING)
  HlpTxt (Metamodel)                 String (STRING)
  Model pointer (Metamodel)          String (STRING)
  Position (Metamodel)               String (STRING)
  VisibleAttrs (Metamodel)           String (STRING)
  WF_Trans (Metamodel)               String (STRING)
Relation classes
  aRb                                A --> B
  dRp                                DOC --> PatientD
  dRp2                               DOC2 --> PatientD
```

1. For Class “Concept”
2. Select New Attribute
3. Create following Attributes:
    1. Concept type (INTERREF)

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

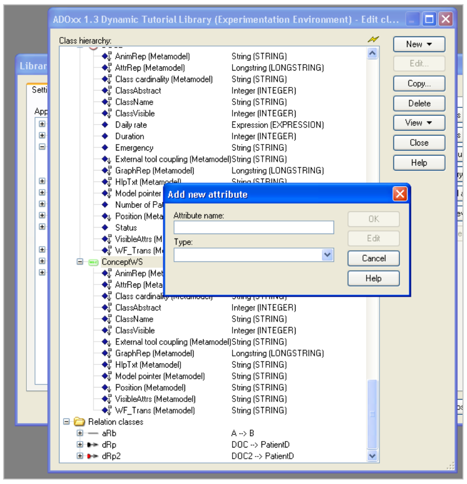


```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                  String (STRING)
    AttrRep (Metamodel)                  Longstring (LONGSTRING)
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    Daily rate                           Expression (EXPRESSION)
    Duration                             Integer (INTEGER)
    Emergency                            String (STRING)
    External tool coupling (Metamodel)String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinali
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    External tool coupling (Metamodel)String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer (Metamodel)            String (STRING)
    Position (Metamodel)                 String (STRING)
    VisibleAttrs (Metamodel)             String (STRING)
    WF_Trans (Metamodel)                 String (STRING)
Relation classes
  aRb                                    A --> B
  dRp                                    DOC --> PatientD
  dRp2                                   DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “ConceptWS”
2. Select New Attribute
3. Create following Attributes:
    1. WS Endpoint (STRING)

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

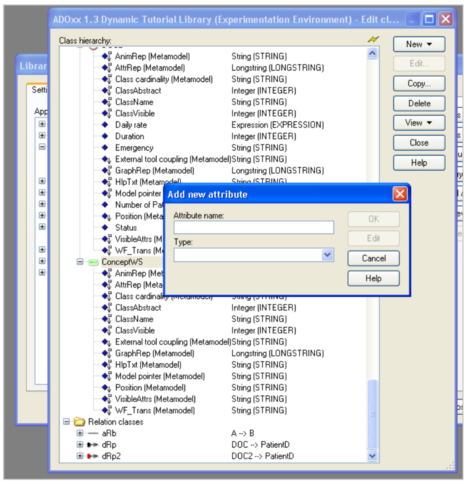


```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                  String (STRING)
    AttrRep (Metamodel)                  Longstring (LONGSTRING)
    Class cardinality (Metamodel)        String (STRING)
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    Daily rate                           Expression (EXPRESSION)
    Duration                             Integer (INTEGER)
    Emergency                            String (STRING)
    External tool coupling (Metamodel)String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinali
    ClassAbstract                        Integer (INTEGER)
    ClassName                            String (STRING)
    ClassVisible                         Integer (INTEGER)
    External tool coupling (Metamodel)String (STRING)
    GraphRep (Metamodel)                 Longstring (LONGSTRING)
    HlpTxt (Metamodel)                   String (STRING)
    Model pointer (Metamodel)            String (STRING)
    Position (Metamodel)                 String (STRING)
    VisibleAttrs (Metamodel)             String (STRING)
    WF_Trans (Metamodel)                 String (STRING)
Relation classes
  aRb                                    A --> B
  dRp                                    DOC --> PatientD
  dRp2                                   DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “Instance”
2. Select New Attribute
3. Create following Attributes:
    1. Concept type (INTERREF)
    2. Description (STRING)

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

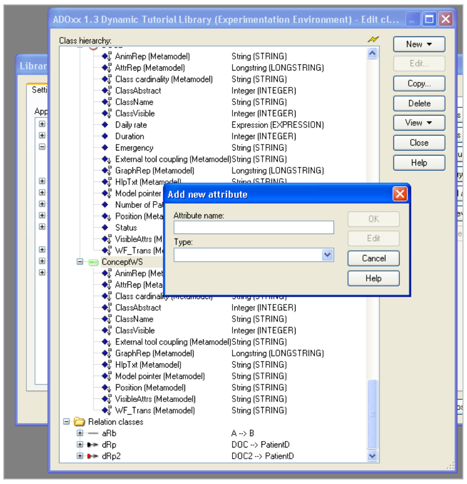


```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                    String (STRING)
    AttrRep (Metamodel)                    Longstring (LONGSTRING)
    Class cardinality (Metamodel)          String (STRING)
    ClassAbstract                          Integer (INTEGER)
    ClassName                              String (STRING)
    ClassVisible                           Integer (INTEGER)
    Daily rate                             Expression (EXPRESSION)
    Duration                               Integer (INTEGER)
    Emergency                              String (STRING)
    External tool coupling (Metamodel)     String (STRING)
    GraphRep (Metamodel)                   Longstring (LONGSTRING)
    HlpTxt (Metamodel)                     String (STRING)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinality (Metamodel)          String (STRING)
    ClassAbstract                          Integer (INTEGER)
    ClassName                              String (STRING)
    ClassVisible                           Integer (INTEGER)
    External tool coupling (Metamodel)     String (STRING)
    GraphRep (Metamodel)                   Longstring (LONGSTRING)
    HlpTxt (Metamodel)                     String (STRING)
    Model pointer (Metamodel)              String (STRING)
    Position (Metamodel)                   String (STRING)
    VisibleAttrs (Metamodel)               String (STRING)
    WF_Trans (Metamodel)                   String (STRING)
Relation classes
  aRb                                      A --> B
  dRp                                      DOC --> PatientD
  dRp2                                     DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “Root concept”
2. Select New Attribute
3. Create following Attributes:
    1. Description (STRING)

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

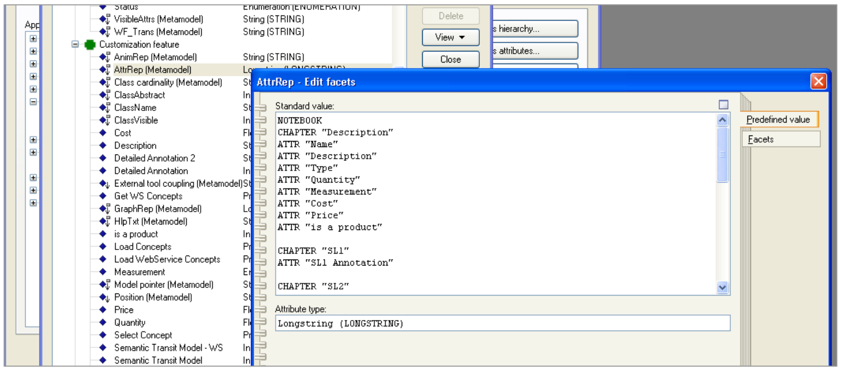


```
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Edit cl...
Class hierarchy:
    AnimRep (Metamodel)                    String (STRING)
    AttrRep (Metamodel)                    Longstring (LONGSTRING)
    Class cardinality (Metamodel)          String (STRING)
    ClassAbstract                          Integer (INTEGER)
    ClassName                              String (STRING)
    ClassVisible                           Integer (INTEGER)
    Daily rate                             Expression (EXPRESSION)
    Duration                               Integer (INTEGER)
    Emergency                              String (STRING)
    External tool coupling (Metamodel)     String (STRING)
    GraphRep (Metamodel)                   Longstring (LONGSTRING)
    HlpTxt (Metamodel)                     String (STRING)
    Model pointer
    Number of Pa
    Position (Meta
    Status
    VisibleAttrs (M
    WF_Trans (M
  ConceptWS
    AnimRep (Met
    AttrRep (Meta
    Class cardinality (Metamodel)          String (STRING)
    ClassAbstract                          Integer (INTEGER)
    ClassName                              String (STRING)
    ClassVisible                           Integer (INTEGER)
    External tool coupling (Metamodel)     String (STRING)
    GraphRep (Metamodel)                   Longstring (LONGSTRING)
    HlpTxt (Metamodel)                     String (STRING)
    Model pointer (Metamodel)              String (STRING)
    Position (Metamodel)                   String (STRING)
    VisibleAttrs (Metamodel)               String (STRING)
    WF_Trans (Metamodel)                   String (STRING)
Relation classes
  aRb                                      A --> B
  dRp                                      DOC --> PatientD
  dRp2                                     DOC2 --> PatientD

Add new attribute
Attribute name:
Type:
```

1. For Class “Note”
2. Select New Attribute
3. Create following Attributes:
    1. Note
    2. Annotated

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types


AttrRep - Edit facets

Standard value:

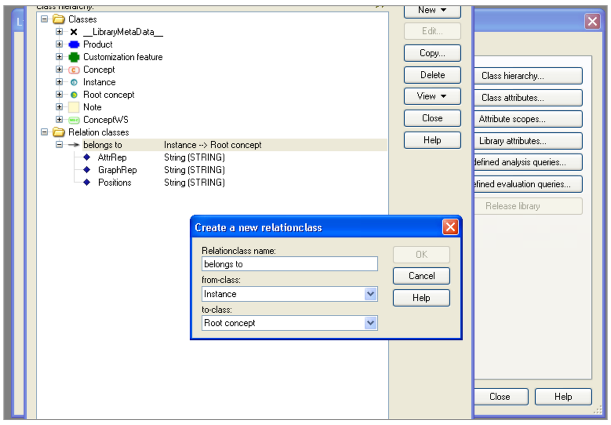
```
NOTEBOOK
CHAPTER "Description"
ATTR "Name"
ATTR "Description"
ATTR "Type"
ATTR "Quantity"
ATTR "Measurement"
ATTR "Cost"
ATTR "Price"
ATTR "is a product"

CHAPTER "SL1"
ATTR "SL1 Annotation"

CHAPTER "SL2"
```

Attribute type: Longstring (LONGSTRING)

1. Select AttRep Attribute of created classes
2. Write AttRep code to make created attributes visible in the Notebook

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types


Create a new relationclass

Relationclass name: belongs to

from-class: Instance

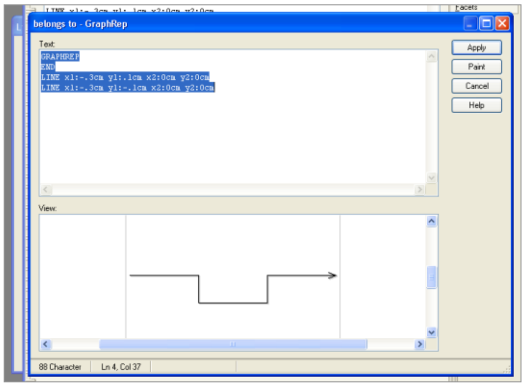
to-class: Root concept

belongs to	Instance --> Root concept
AttrRep	String (STRING)
GraphRep	String (STRING)
Positions	String (STRING)

1. Define a Relationclass
2. “belongs to” between Instance -> Root concept

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types

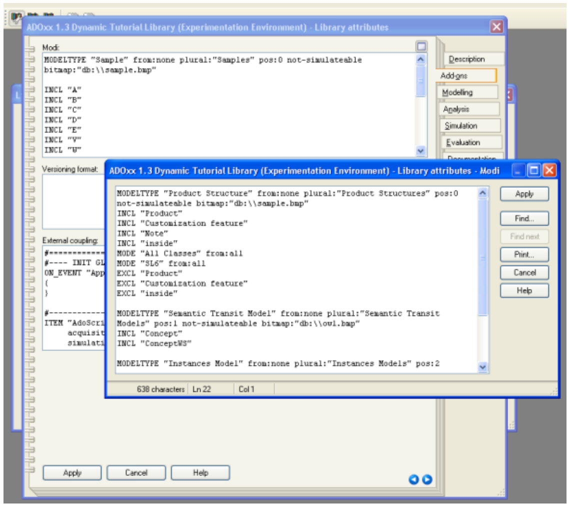


```
GRAPHREP
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm
```

1. Select GraphRep Attribute
2. Open Dialog
3. Write GraphRep code to visualise the Relationclass

Definition of Modelling Language on ADOxx

Define Classes, Attributes and Model Types


ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Library attributes
Modi:
Description
Add-ons
Modelling
Analysis
Simulation
Evaluation
Versioning format:
External coupling:
ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) - Library attributes - Modi
Apply
Find...
Find next
Print...
Cancel
Help
638 characters
Ln 22
Col 1
Apply
Cancel
Help

```
MODELTYPE "Sample" from:none plural:"Samples" pos:0 not-simulateable
bitmap:"db:\\sample.bmp"

INCL "A"
INCL "B"
INCL "C"
INCL "D"
INCL "E"
INCL "V"
INCL "W"
```

```
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"

MODELTYPE "Semantic Transit Model" from:none plural:"Semantic Transit
Models" pos:1 not-simulateable bitmap:"db:\\owl.bmp"
INCL "Concept"
INCL "ConceptWS"

MODELTYPE "Instances Model" from:none plural:"Instances Models" pos:2
```

1. Select Library attributes
2. Click on tab Add-ons
3. Expand the “Modi”
4. Write the Model Type definition Code for “Product Structure, Instances Model, Semantic Transit Model”

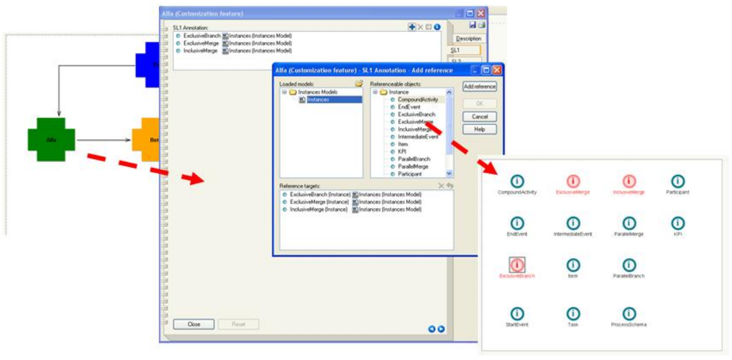
Agenda

www.omilab.org

- SCM Scenario
- Hybrid Modelling Approach Introduction
- Semantic Lifting
- **Hands On Sessions:**
  - Prerequisites
  - Definition of Model Structure on ADOxx
  - **Semantic Lifting**
- Q&A

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.



SL1 Annotation:
- ExclusiveBranch Instances (Instances Model)
- ExclusiveMerge Instances (Instances Model)
- InclusiveMerge Instances (Instances Model)

Loaded models:
- Instances Models
  - Instances

Referenceable objects:
- Instance
  - CompoundActivity
  - EndEvent
  - ExclusiveBranch
  - ExclusiveMerge
  - InclusiveMerge
  - IntermediateEvent
  - Item
  - KPI
  - ParallelBranch
  - ParallelMerge
  - Participant

Reference targets:
- ExclusiveBranch (Instance) Instances (Instances Model)
- ExclusiveMerge (Instance) Instances (Instances Model)
- InclusiveMerge (Instance) Instances (Instances Model)

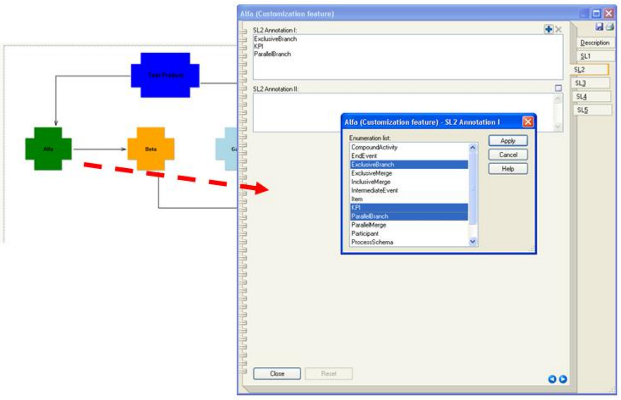
SL1 - Model to Model Instance with Interref: Code

ALL Code

```
ATTRIBUTE <SL1 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"
"
```

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



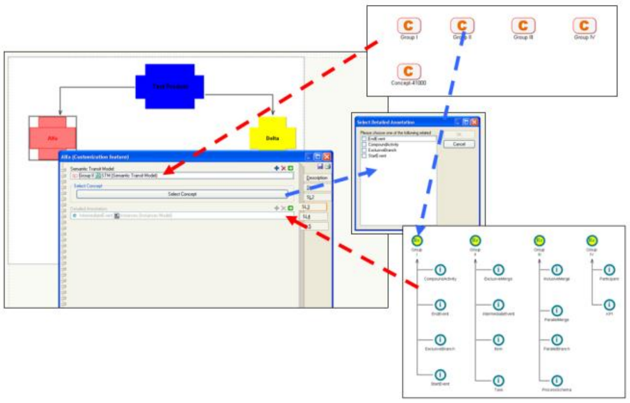
SL2 - Textual Annotation within Model: Code

ALL Code

```
            ATTRIBUTE <SL2 Annotation I>
            TYPE ENUMERATIONLIST

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

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

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.



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

AdoScript Code

```
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)
```

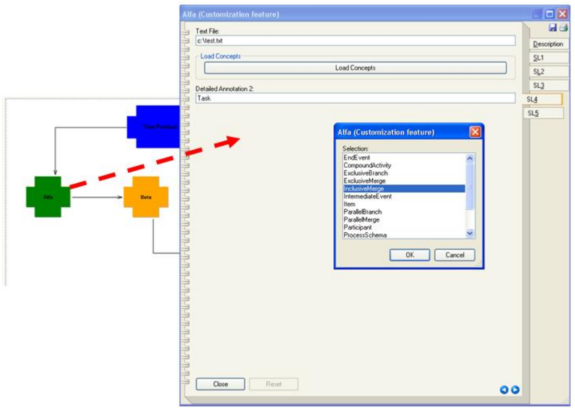
SL3 - Model to Model Instance through the Semantic Transit Model: Code (2/2)

AdoScript Code

```
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!")
}
```

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)



Text File: c:\test.txt

Detailed Annotation 2: Task

Selection:
EndEvent
CompoundActivity
ExclusiveBranch
ExclusiveMerge
InclusiveMerge
IntermediateEvent
Item
ParallelBranch
ParallelMerge
Participant
ProcessSchema

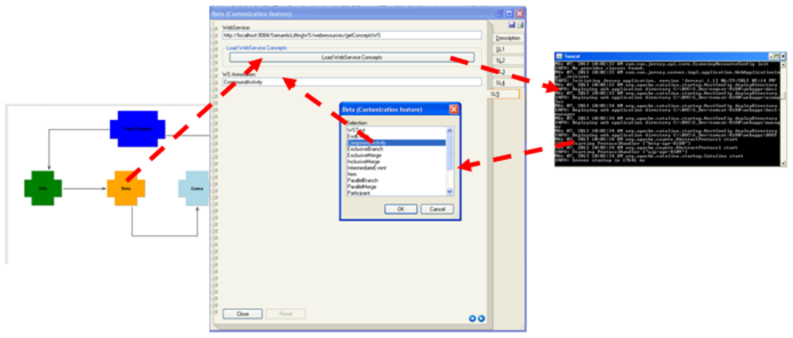
SL4 - Model to Text File with AdoScript: Code

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!")
}
```

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



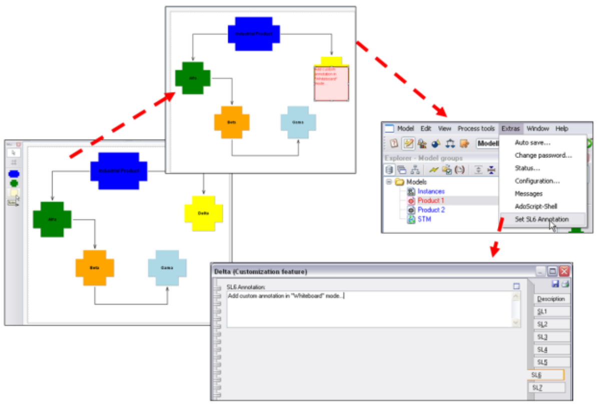
SL5 - Model to Web Service with AdoScript: Code

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")
```

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.



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

AdoScript Code

```
#-----------------------------------------------
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)

AdoScript Code

```
             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)
```

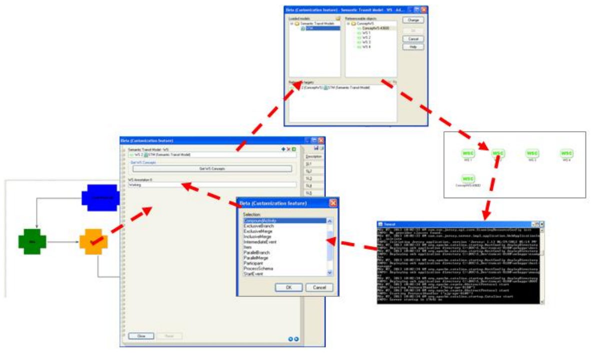
SL6 - Graphical Annotation within Model (Whiteboard): Code (3/3)

AdoScript Code

```
             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")))
}
```

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

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.



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

AdoScript 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")
```

Agenda

OMILAB®
www.omilab.org

- SCM Scenario
- Hybrid Modelling Approach Introduction
- Semantic Lifting
- Hands On Sessions:
  - Prerequisites
  - Definition of Model Structure on ADOxx
  - Semantic Lifting
- **Q&A**

Further Questions?

tutorial@adoxx.org

References

- Kühn H. (2004). Methodenintegration im Business Engineering, PhD Thesis, University of Vienna, April 2004.
- Keynote by Karagiannis, D. at The 4th IEEE International Workshop on Requirements Engineering For Services, COMPSAC 2010
- Open Model Initiative: A Feasibility Study, 2008