ACM/IEEE 15th International Conference
on Model Driven Engineering Languages & Systems
MODELS 2012

MODELS 2012
innsbruck

FROM MODEL EDITORS TO MODELLING TOOLS: OPERATIONALIZING MODELLING METHODS WITH ADOxx

Tutorial 2

Innsbruck, October 1, 2012

Dimitris Karagiannis, Hans-Georg Fill, Srdjan Zivkovic, Wilfrid Utz

[dk@dke.univie.ac.at | hans-georg.fill@dke.univie.ac.at |
srdjan.zivkovic@dke.univie.ac.at | wilfrid.utz@dke.unive.ac.at]

OMiLAB®
www.omilab.org

www.adoxx.org

universität
wien

OMiLAB: Approach

OMiLAB
www.omilab.org

- A **research and experimental laboratory** for the conceptualization, development and deployment of modelling methods and the models designed with them.
- Project space for Engineering of modelling methods and **modelling tools**
- A space for a community of researchers and practitioners sharing a common understanding about **model value**

Organisation: University of Vienna,
Faculty of Computer Science

Research Group: Knowledge Engineering

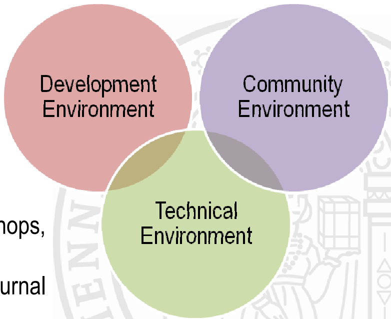
OMiLAB@Faculty of Computer Science
Währinger Str. 29

inf Fakultät für Informatik

OMiLAB: Environment

OMiLAB
www.omilab.org

- Development environment consists of
  - Core (Open Use): ADOxx on OMiLAB
  - Add-Ons (Open Source): implemented community tools such as Model Annotator, GraphRep Generator, Model Publisher, Method Publisher, OM-Repository, Meta-Model Browser, MLEA – Modelling Language Engineering Assistant
- Technical environment supports
  - virtual and physical accessibility
  - packaging and deployment capabilities
- Community environment provides
  - Web-platform based on Liferay
  - Community events like conferences, workshops, summer schools
  - Publications like books, conference and journal papers
  - Project networking activities
  - Newsletters, media and OM-TV



www.omilab.org

Agenda

www.omilab.org

- **Model Value**
- Definition of Model Structure on ADOxx
- Processing of Model Structure on ADOxx
  - Visualisation Functionality
  - Transformation Functionality
  - Analysis Functionality
- Conclusion

Model Definitions

- **Model as mappings of reality**
  …models as mappings of parts of reality for a particular purpose…
- **Model as a construction**
  ...the result of a construction of a modeler who declares for model users a representation of an original as significant at a given time using a language…

(Source: Schütte and Becker, 1998)

Model with Different Values

Representation Characteristic

“Models as a representation of natural or artifical originals, that again can be models.“ [1] (translated)

Abstraction Characteristic

“Models in general do not capture all attributes of the represented original, but only those that seem relevant to the modeller or model user.“[1] (translated)

Pragmatic Characteristic

Models meet their substituion function for specific subjects, within a pre-determined time interval and with limitations on defined intellectual and/or real operations. [1] (translated)

Source: Stachowiak 1973

Introduction of Terms

- **Modelling Language:**
  Modelling constructs (object types) and their relations (relation types) to each other to declare a model.
- **Metamodel:**
  The model of the syntax of the modelling language
- **Meta^2 Model:**
  Model of abstract syntax of a language to describe meta models.
- **Modelling Technique:**
  A modelling language and proceeding instructions for creation of a model in this modelling language.
- **Mechanisms und Algorithms:**
  Provision of functionalities to process models such as manipulation, visualisation, query, transformation or simulation depending on the modelling language and modelling procedure.

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006; Kühn 2004; Karagiannis and Visic, 2011)

Model Values: An Example

THE RESULTS OF MODELLING CAN BE USED FOR GENERATING SOFTWARE, BUT ALSO ACT AS A BASIS OF ENTERPRISE KNOWLEDGE PLATFORMS

MACHINE PROCESSABLE

Cf. (Karagiannis, 2012 – Presentation at FInES – "Translating Knowledge Into Growth: Views from ICT Research to Support Future Business Innovation")

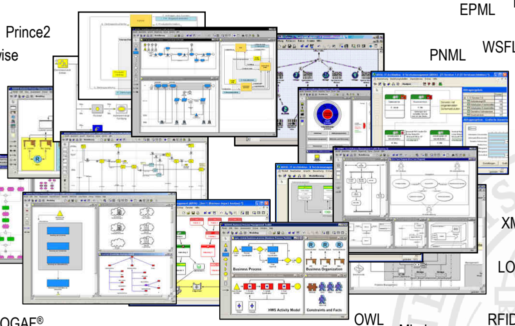
Some machine-processable formats …

CMMI$^{®}$
DoDAF
Goals
Business Models
BPSS
SCM-SCOR
ADONIS
BPDM
Balanced Scorecard
SITM framework
BPEL
BPMN
EPC
OWL-S
BPML
WS-CDL
Business Processes
. . .
Projects
EPML
LOVEM
IDEF
Prince2
. . .
Casewise
PNML
WSFL
Reference Processes
KPIs
XPDL
Skills
Topics Maps
E-BPMS
CRM
WSCI
Knowledge Management
E-Learning
eTOM
XLANG
WSCL
XKMS
Zachmann
FEAF
GRID
XML
UML
Ontology
SAML
. . .
Internet
W-LAN
LOOM
TCP/IP
SAMM
UML
OPM3
. . .
TOGAF$^{®}$
OWL
RFID
LegalXML
CobiT$^{®}$
Mind Maps
ISMS
. . .
DMAL-S
EVM3
HumanML
RDF
EuroSOX
Metamodelling
Web Services
GSM
P-CMM
GPRS
BASEII
XBRL
ISO 27 002
Agents
MOF
BS 15 000
SOX
Workflow
SW-CMM
WiMax
ISO 27 001
KonotraG
Semantic Web
ERP
BS7799-2
UMTS
ISO 20 000
WiFi
ITIL$^{®}$
ISO 27001
HR-XML
ebXML
LM3

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006)

… from an editor implementation, to …

SCM-SCOR Business Models DoDAF CMMI® Goals BPSS ADONIS
BPMN OWL-S Balanced Scorecard SITM framework BPDM EPC BPEL
. . . BPML WS-CDL Business Processes
Projects EPML LOVEM
Prince2 IDEF . . .
Casewise PNML WSFL Reference Processes
KPIs XPDL
Topics Maps Skills CRM E-BPMS
Knowledge Management WSCI E-Learning
eTOM XLANG WSCL
Zachmann XKMS
FEAF XML GRID
UML SAML Ontology
. . . Internet W-LAN
LOOM TCP/IP
OPM3 SAMM UML . . .

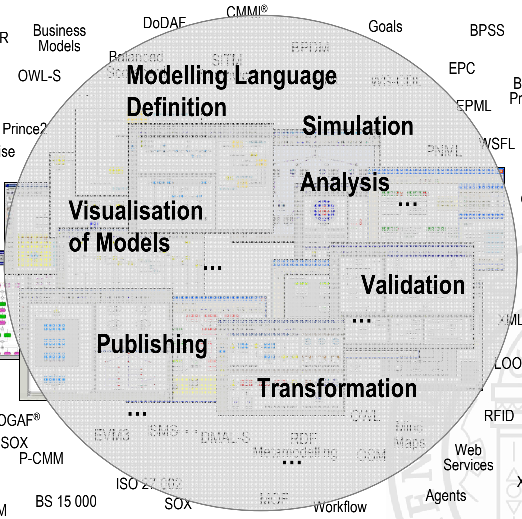


TOGAF® OWL RFID
CobiT® EVM3 ISMS . . . DMAL-S RDF Mind Maps LegalXML
EuroSOX Metamodelling GSM Web Services HumanML
P-CMM
BASEII ISO 27 002 XBRL GPRS
SW-CMM BS 15 000 SOX MOF Workflow Agents WiMax
ISO 27 001 KonotraG
BS7799-2 Semantic Web ERP UMTS
ITIL® ISO 27001 ISO 20 000 ebXML WiFi HR-XML
LM3

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006)

… to full-fledged modelling tool support.

SCM-SCOR Business Models DoDAF CMMI® Goals BPSS ADONIS
BPMN OWL-S Balanced Scorecard SITM BPDM WS-CDL EPC BPEL
EPML Business Processes
Projects LOVEM
Prince2 PNML WSFL IDEF . . .
Casewise Reference Processes
KPIs XPDL
Topics Maps Skills CRM E-BPMS
Knowledge Management WSCI E-Learning
eTOM XLANG WSCL
Zachmann XKMS
FEAF XML GRID
UML SAML Ontology
. . . Internet W-LAN
LOOM TCP/IP
OPM3 SAMM UML . . .



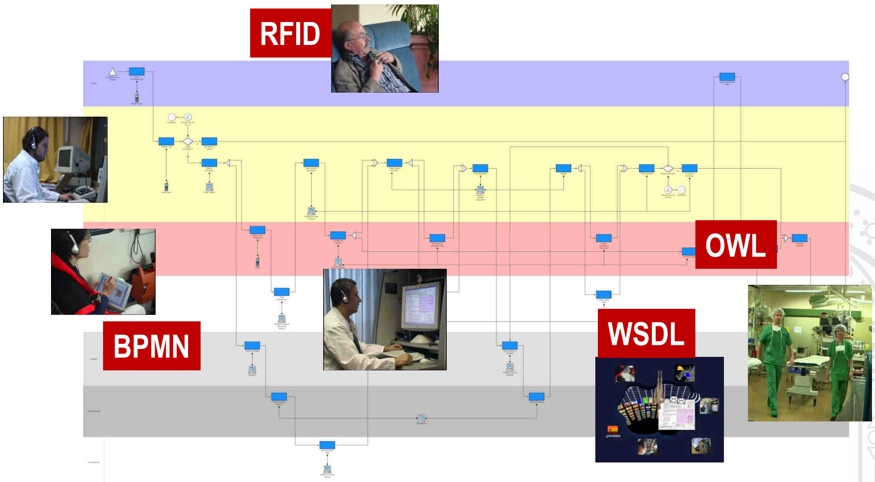
Modelling Language Definition
Simulation
Analysis
…
Visualisation of Models
…
Validation
…
Publishing
Transformation
…
…

TOGAF® OWL RFID
CobiT® EVM3 ISMS . . . DMAL-S RDF Mind Maps LegalXML
EuroSOX Metamodelling GSM Web Services HumanML
P-CMM
BASEII ISO 27 002 XBRL GPRS
SW-CMM BS 15 000 SOX MOF Workflow Agents WiMax
ISO 27 001 KonotraG
BS7799-2 Semantic Web ERP UMTS
ITIL® ISO 27001 ISO 20 000 ebXML WiFi HR-XML
LM3

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006)

Scenario: Mobile eHealth Analysis and Simulation

AKOGRIMO Project

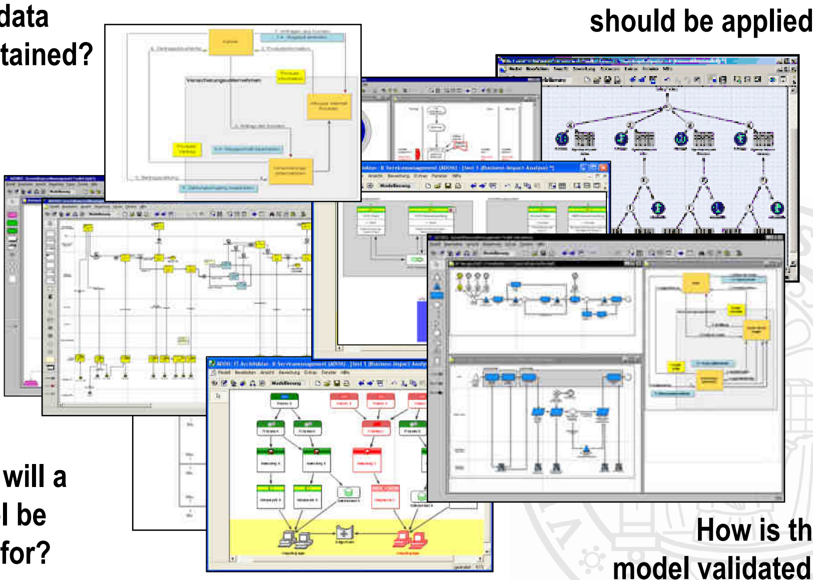


E-health scenario, for more information see video on
http://www.mobilegrid.org

The necessary information for model processing

What data
is contained?

Which algorithms
should be applied?



What will a
model be
used for?

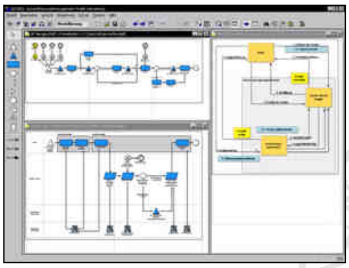
How is the
model validated?

Some functionalities of modelling tools

Visualisation of models

User interaction like: drag and drop, zoom, grid snap, print, etc.

Simulation of models



Modelling language definition

Publishing in multiple formats

Transformation of models

Analyse models and evaluate the results

Exchange of models

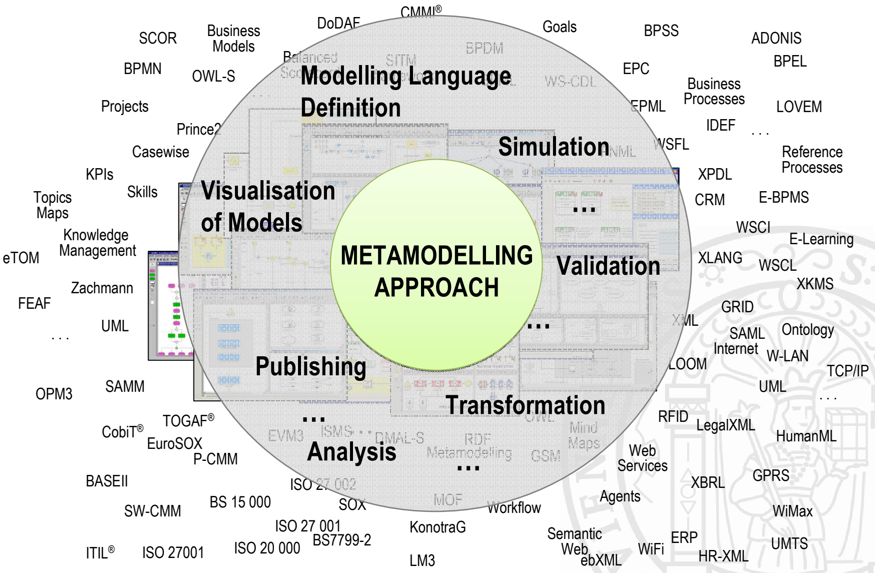
User access rights

Storage and Manipulation of Models

Security and Safety

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006; Fill, 2009)

A Metamodel-based Realisation Approach



METAMODELLING APPROACH

Modelling Language Definition

Simulation

...

Validation

...

Transformation

...

Analysis

...

Publishing

Visualisation of Models

Cf. (Karagiannis and Kühn, 2002; Karagiannis and Höfferer, 2006)

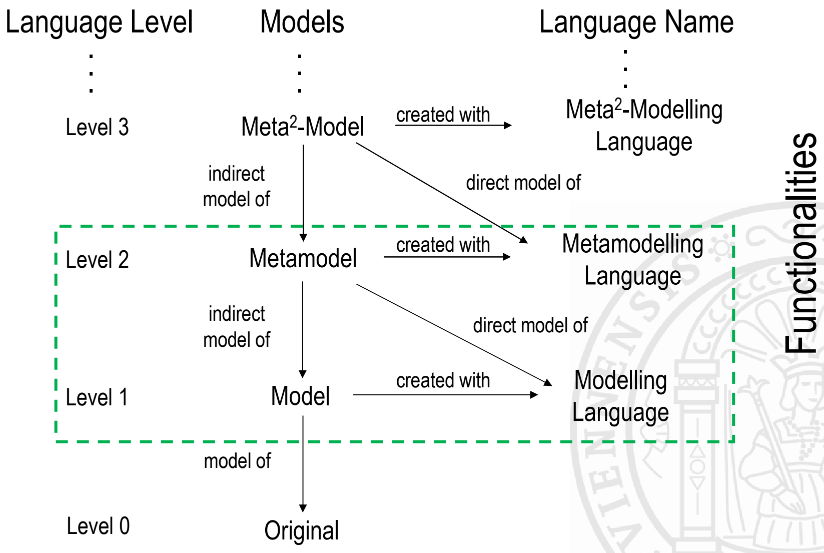
...FOCUS OF TUTORIAL

Agenda

OMILAB®
www.omilab.org

- Model Value
- **Definition of Model Structure on ADOxx**
- Processing of Model Structure on ADOxx
  - Visualisation Functionality
  - Transformation Functionality
  - Analysis Functionality
- Conclusion

Definition of Model Structure and Functionalities



Language Level | Models | Language Name

Level 3 — Meta^2-Model — created with → Meta^2-Modelling Language

Meta^2-Model — indirect model of → Metamodel; Meta^2-Model — direct model of → Metamodelling Language

Level 2 — Metamodel — created with → Metamodelling Language

Metamodel — indirect model of → Model; Metamodel — direct model of → Modelling Language

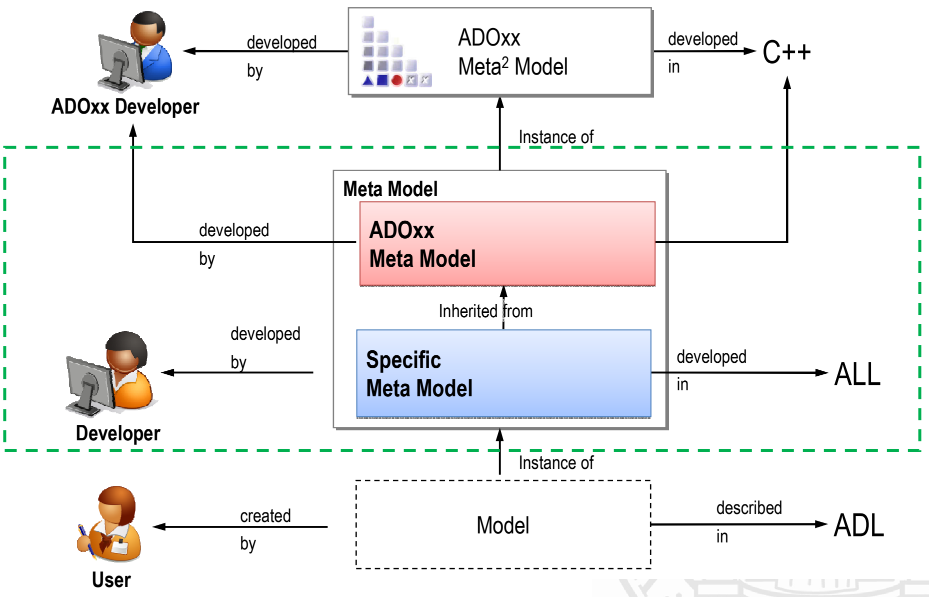
Level 1 — Model — created with → Modelling Language

Model — model of → Original

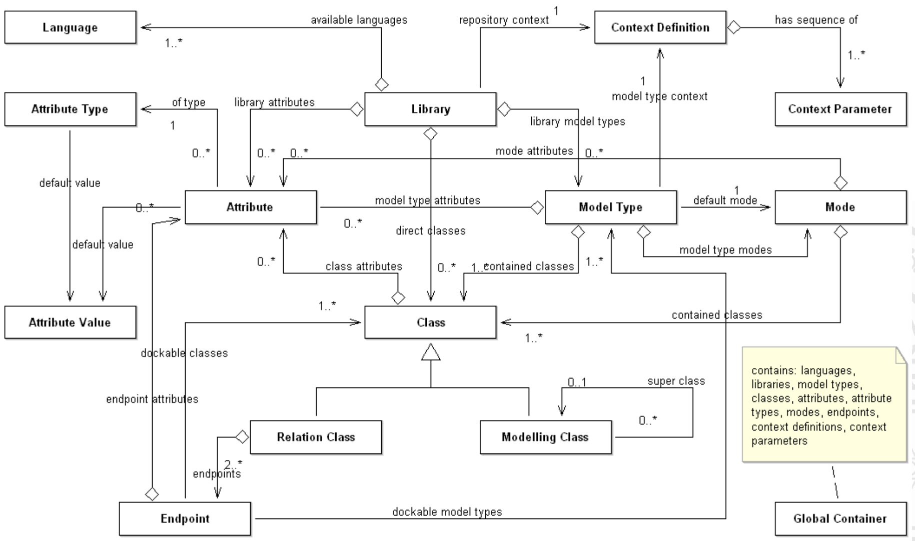
Level 0 — Original

Functionalities

ADOxx Platforms Hierarchy



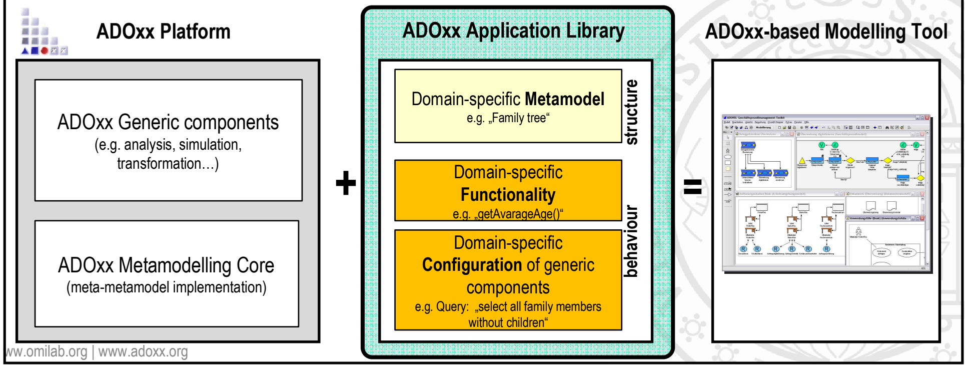
ADOxx Meta²-Model



contains: languages, libraries, model types, classes, attributes, attribute types, modes, endpoints, context definitions, context parameters

The ADOxx Library Concept:
More than a Metamodel

- The ADOxx Application Library is a concept which encapsulates both the structural and the behavioural aspects of metamodelling
- The ADOxx Application Library consists of:
  - Structural part: Metamodel definition
  - Behavioural part: Metamodel-specific functionality, Configuration of generic functionalities
- The ADOxx Application Library is a self-contained platform configuration package containing all neccessary artefacts to configure a fully-fledged modelling tool



ADOxx Platform

ADOxx Generic components
(e.g. analysis, simulation, transformation…)

ADOxx Metamodelling Core
(meta-metamodel implementation)

+

ADOxx Application Library

Domain-specific **Metamodel**
e.g. „Family tree"

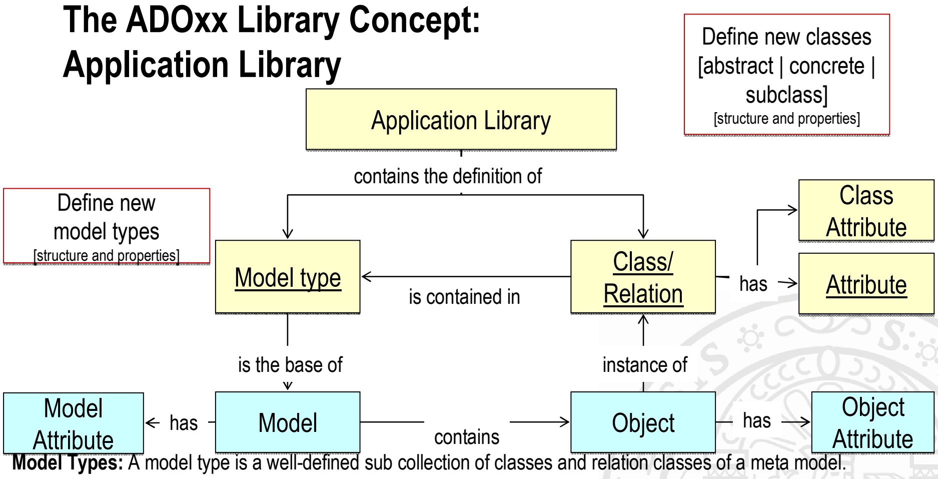
Domain-specific **Functionality**
e.g. „getAvarageAge()"

Domain-specific **Configuration** of generic components
e.g. Query: „select all family members without children"

=

ADOxx-based Modelling Tool

The ADOxx Library Concept:
Application Library



Define new classes [abstract | concrete | subclass]
[structure and properties]

Application Library

contains the definition of

Define new model types
[structure and properties]

Model type

is contained in

Class/ Relation

has

Class Attribute

Attribute

is the base of

instance of

Model Attribute

has

Model

contains

Object

has

Object Attribute

Model Types: A model type is a well-defined sub collection of classes and relation classes of a meta model.

Classes: A class is a construct that is used as a template to create objects of that class. The objects of a class are alternatively called "instances"

Attributes: An attribute is a property of a modelling construct such as a model, object or relation. Each attribute has a type and a value.

Relations: A relation class is a construct that is used as a template to create relations between objects. A relation class is defined between classes. A relation is always a directed connection between objects, i.e. each relation has a from-side and a to-side.

Cf. (Junginger et al., 2000; Kühn, 2004; Fill, Redmond, Karagiannis, 2012)

Class Types in ADOxx

Define new classes
[abstract | concrete | subclass]
[structure and properties]

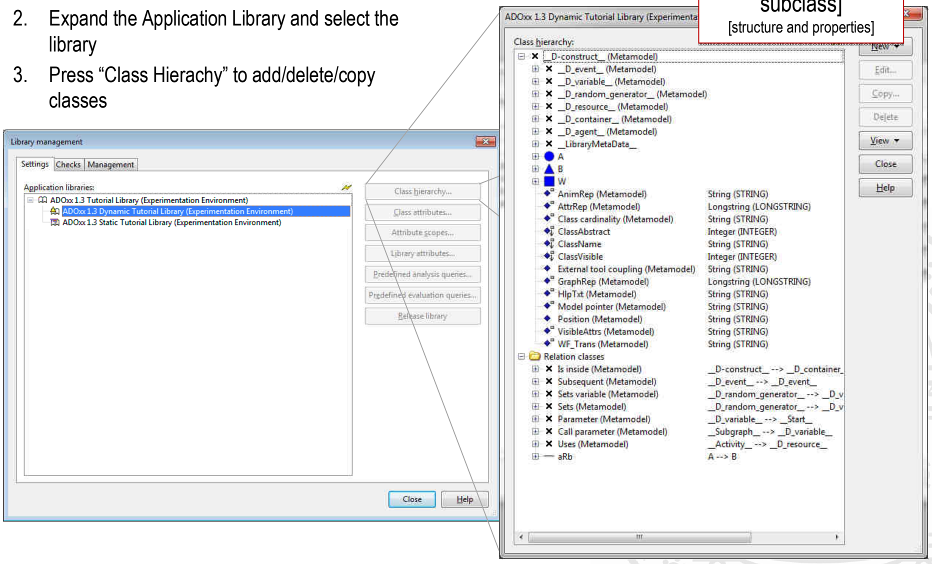
- **Abstract Classes**
  - – Abstract classes are self-defined classes enabling to structure the meta model and define syntax in form of attributes and semantic, which is inherited by sub-classes.
  - – Abstract classes either inherit from the root class of the meta model, or from any other class of the meta model. Hence, they inherit the behaviour from their super-class – which is often a pre-defined abstract class from the ADOxx meta model.
  - – Abstract classes enable an efficient meta model, hence they may not be in every ADOxx meta model.
  - – Nomenclature: _ Class Name _
- **(Concrete) Classes**
  - – Classes are self-defined classes defining a concrete modelling class that can be used, when applying the corresponding modelling language. Hence all model objects in every model created on ADOxx is an instance of a class.
  - – Classes inherit the semantic and the attributes from the Pre-defined abstract class and additionally - in case of inheriting - from the abstract class.
  - – Classes enable the realisation of a concrete meta model.
  - – Nomenclature: Class Name

Cf. (Fill, Redmond, Karagiannis, 2012)

Demonstration: Class Definition 1

Define new classes
[abstract | concrete | subclass]
[structure and properties]

1. Open the Library Management Component
2. Expand the Application Library and select the library
3. Press “Class Hierachy” to add/delete/copy classes



Library management

Application libraries:

- ADOxx 1.3 Tutorial Library (Experimentation Environment)
  - ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment)
  - ADOxx 1.3 Static Tutorial Library (Experimentation Environment)

ADOxx 1.3 Dynamic Tutorial Library (Experimenta

Class hierarchy:

- ✕ _D-construct_ (Metamodel)
  - ✕ _D_event_ (Metamodel)
  - ✕ _D_variable_ (Metamodel)
  - ✕ _D_random_generator_ (Metamodel)
  - ✕ _D_resource_ (Metamodel)
  - ✕ _D_container_ (Metamodel)
  - ✕ _D_agent_ (Metamodel)
  - ✕ _LibraryMetaData_
  - ● A
  - ▲ B
  - ■ W

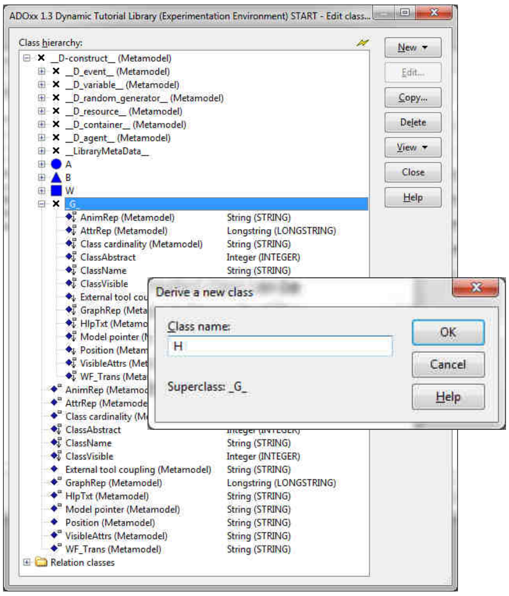
AnimRep (Metamodel)	String (STRING)
AttrRep (Metamodel)	Longstring (LONGSTRING)
Class cardinality (Metamodel)	String (STRING)
ClassAbstract	Integer (INTEGER)
ClassName	String (STRING)
ClassVisible	Integer (INTEGER)
External tool coupling (Metamodel)	String (STRING)
GraphRep (Metamodel)	Longstring (LONGSTRING)
HlpTxt (Metamodel)	String (STRING)
Model pointer (Metamodel)	String (STRING)
Position (Metamodel)	String (STRING)
VisibleAttrs (Metamodel)	String (STRING)
WF_Trans (Metamodel)	String (STRING)

Relation classes

✕ Is inside (Metamodel)	_D-construct_ --> _D_container_
✕ Subsequent (Metamodel)	_D_event_ --> _D_event_
✕ Sets variable (Metamodel)	_D_random_generator_ --> _D_v
✕ Sets (Metamodel)	_D_random_generator_ --> _D_v
✕ Parameter (Metamodel)	_D_variable_ --> _Start_
✕ Call parameter (Metamodel)	_Subgraph_ --> _D_variable_
✕ Uses (Metamodel)	_Activity_ --> _D_resource_
— aRb	A --> B

Demonstration: Class Definition 2

Define new classes [abstract | concrete | subclass]
[structure and properties]


ADOxx 1.3 Dynamic Tutorial Library (Experimentation Environment) START - Edit class...

Class hierarchy:

```
_D-construct_ (Metamodel)
  _D_event_ (Metamodel)
  _D_variable_ (Metamodel)
  _D_random_generator_ (Metamodel)
  _D_resource_ (Metamodel)
  _D_container_ (Metamodel)
  _D_agent_ (Metamodel)
  _LibraryMetaData_
  A
  B
  W
  _G_
    AnimRep (Metamodel)              String (STRING)
    AttrRep (Metamodel)              Longstring (LONGSTRING)
    Class cardinality (Metamodel)    String (STRING)
    ClassAbstract                    Integer (INTEGER)
    ClassName                        String (STRING)
    ClassVisible
    External tool cou
    GraphRep (Meta
    HlpTxt (Metamo
    Model pointer (I
    Position (Metam
    VisibleAttrs (Met
    WF_Trans (Meta
  AnimRep (Metamod
  AttrRep (Metamode
  Class cardinality (M
  ClassAbstract
  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
```

Derive a new class

Class name: H

Superclass: _G_

1. Add a new concrete class below the abstract element that is used to define a concrete class
2. Select the abstract class, click "New" -> "New class"
3. Name the new class

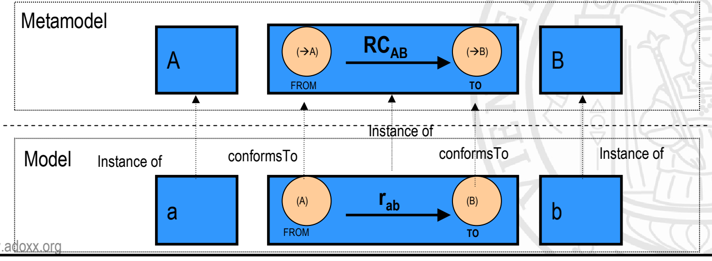
The new created class can be identified on instance level by the "Name" attribute. This attribute is automatically/implict available for each class

---

Relation Types

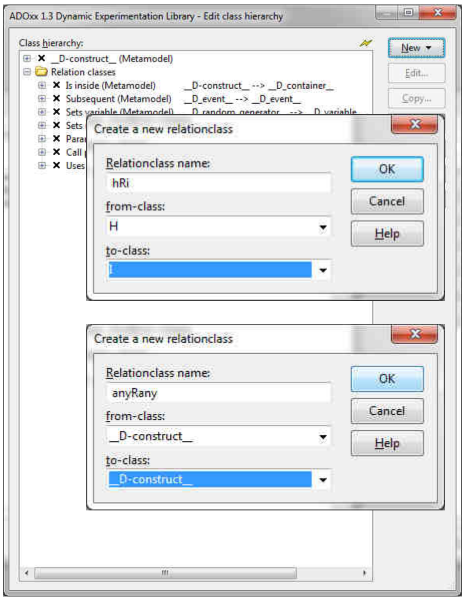
Define new classes [abstract | concrete | subclass]
[structure and properties]

- Relations in ADOxx are expressed either as a class "Relation Class" or as a pointer in form of an attribute called "InterRef".
- Relation as Class "RC"
  - describes relationship between two objects from two or more classes within one model.
  - has start and endpoints define which (abstract) classes a relation can connect
  - Cardinality and attribute defined the semantic of the relations class
- Relation as Attribute "InterRef"
  - Is a special configuration of a Relation Class and describes the relationship between two objects from two or more classes within or across models.
  - Is a pointer represented as an attributed in the class the relation starts from, with defined classes the relation can point to.
  - Cardinality defines the semantic of the InterRef



Demonstration: Relation Class Definition

Define new classes
[abstract | concrete | subclass]
[structure and properties]


ADOxx 1.3 Dynamic Experimentation Library - Edit class hierarchy
Class hierarchy:
__D-construct__ (Metamodel)
Relation classes
Is inside (Metamodel) __D-construct__ --> __D_container__
Subsequent (Metamodel) __D_event__ --> __D_event__
Sets variable (Metamodel) __D_random_generator__ --> __D_variable
Sets
Para
Call
Uses
New
Edit...
Copy...
Create a new relationclass
OK
Cancel
Help
Create a new relationclass
OK
Cancel
Help

Relationclass name: hRi
from-class: H
to-class: I

Relationclass name: anyRany
from-class: __D-construct__
to-class: __D-construct__

- Add two new relation classes to connect classes
  - Click “New” -> “New relation class”
  - Name new relation class
  - Define from-class
  - Define to-class

Definition of Attributes

Define new classes
[abstract | concrete | subclass]
[structure and properties]

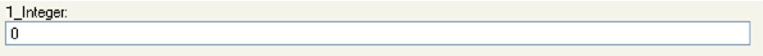
- Attributes for classes and relation classes have to b the definition section of the class/relation class with
- The following attribute types are possible:

INTEGER	**integer**
DOUBLE	**floating number**
STRING	**string – max. 3699 symbols**
LONGSTRING	string – max. 32000 symbols
ENUMERATION	enumeration for selecting a characteristic
ENUMERATIONLIST	enumeration for selecting one or several characteristics
TIME	time
DATE	date
DATETIME	date and time
PROGRAMCALL	***enumeration for selecting a program***
RECORD	a table of attributes
EXPRESSION	a formula
INTERREF	***reference on a model or an instance***
ATTRPROFREF	a preset set of attribuite values

Demonstration:
Attribute Types and their Appearance

Numerical Attributes: Integer (INTEGER)

Define new classes
[abstract | concrete | subclass]
[structure and properties]



1_Integer:
0

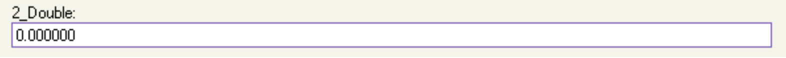
- An attribute of the type "Integer" is defined as an integer from -1,999,999,999 to 1,999,999,999.
- An ADOxx integer is limited to 10 digits plus an optional sign ('+' or '-')
- The standard value of attributes of this type is "0" or a value defined

---

Demonstration:
Attribute Types and their appearance

Numerical Attributes: Floating number (DOUBLE)

Define new classes
[abstract | concrete | subclass]
[structure and properties]



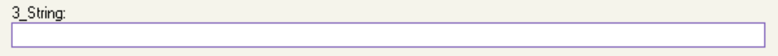
2_Double:
0.000000

- The amount of decimal places is defined by the attribute definition
- An attribute of the type "Double" is defined for a float within +/- 999,999,999,999,999 for an integer (without decimal places) or +/- 999,999,999.999999 for figures with 6 decimals.
- The corresponding attribute value is displayed to 6 decimal places. That means that a double value should not exceed a total of 15 significant digits with at last 6 decimal digits!
- The standard value of attributes of this type is "0.000000" or a value defined in the application library.

Demonstration:
Attribute Types and their appearance

String attributes: String (STRING)

Define new classes [abstract | concrete | subclass]
[structure and properties]


3_String:

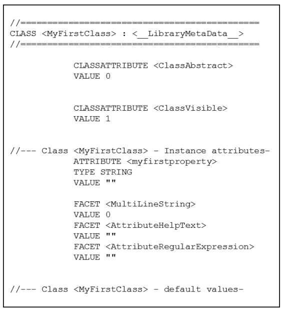
- An attribute of the type "String" is defined for texts up to 3700 characters of any type.
  - Hint: The maximum number of characters is 250 for name. That concerns classes, relation, instances, attributes, application models, libraries and application libraries.
  - Model names have a special rule!
- The standard value of attributes of this type is "" (no entry) or a value defined in the application library.

Wrap Up: Definition of Model Structure on ADOxx

- Introduction of ADOxx Library Concept
- Demonstration of implementation of model structure

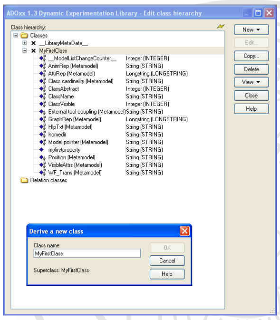
RESULT ACCOMPLISHED:

- Implemented model structure on ADOxx



```
//=============================================
CLASS <MyFirstClass> : <__LibraryMetaData__>
//=============================================

           CLASSATTRIBUTE <ClassAbstract>
           VALUE 0


           CLASSATTRIBUTE <ClassVisible>
           VALUE 1


//--- Class <MyFirstClass> - Instance attributes-
           ATTRIBUTE <myfirstproperty>
           TYPE STRING
           VALUE ""

           FACET <MultiLineString>
           VALUE 0
           FACET <AttributeHelpText>
           VALUE ""
           FACET <AttributeRegularExpression>
           VALUE ""


//--- Class <MyFirstClass> - default values-
```

Model Structure Definition using ADOxx Library Language (ALL)



Classes
__LibraryMetaData__
MyFirstClass

__ModelListChangeCounter__	Integer (INTEGER)
AnimRep (Metamodel)	String (STRING)
AttrRep (Metamodel)	Longstring (LONGSTRING)
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)
homedir	String (STRING)
Model pointer (Metamodel)	String (STRING)
myfirstproperty	String (STRING)
Position (Metamodel)	String (STRING)
VisibleAttrs (Metamodel)	String (STRING)
WF_Trans (Metamodel)	String (STRING)

Relation classes

Class name: MyFirstClass

Superclass: MyFirstClass

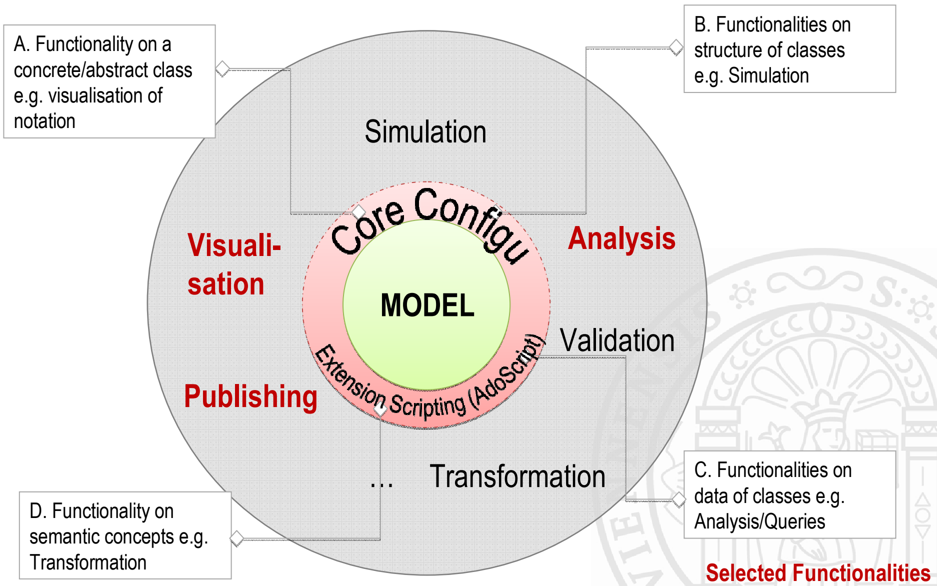
Model Structure Definition using Development Environment

Agenda

www.omilab.org

- Model Value
- Definition of Model Structure on ADOxx
- **Processing of Model Structure on ADOxx**
  - Visualisation Functionality
  - Transformation Functionality
  - Analysis Functionality
- Conclusion

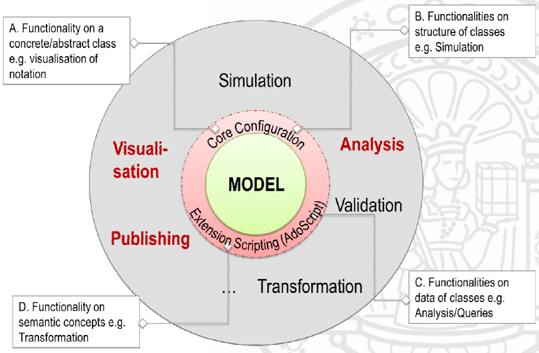
Model Processing Classification


A. Functionality on a concrete/abstract class e.g. visualisation of notation
B. Functionalities on structure of classes e.g. Simulation
Simulation
Core Configu
Visuali-sation
Analysis
MODEL
Validation
Extension Scripting (AdoScript)
Publishing
… Transformation
C. Functionalities on data of classes e.g. Analysis/Queries
D. Functionality on semantic concepts e.g. Transformation

Selected Functionalities for Tutorial

Core Configuration

- User and Access Right Management
- File Management
- Library Persistence (DB and File Persistence)
- Model Persistence (DB and File Persistence)
- Serialization Functionality (Import/Export)
- …



A. Functionality on a concrete/abstract class e.g. visualisation of notation

B. Functionalities on structure of classes e.g. Simulation

C. Functionalities on data of classes e.g. Analysis/Queries

D. Functionality on semantic concepts e.g. Transformation

Simulation

Core Configuration

MODEL

Extension Scripting (AdoScript)

Visuali-sation

Analysis

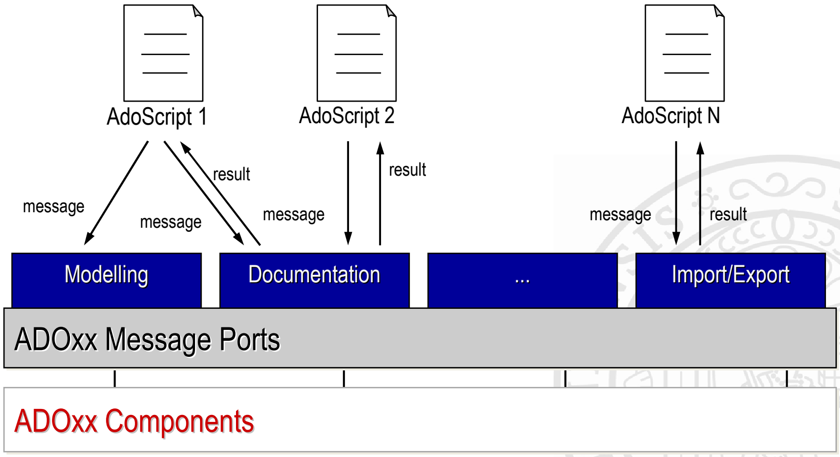
Validation

Publishing

… Transformation

Extension Scripting (AdoScript)

AdoScript: The ADOxx DSL



AdoScript 1

AdoScript 2

AdoScript N

message

message

result

message

result

message

result

Modelling

Documentation

...

Import/Export

ADOxx Message Ports

ADOxx Components

Programmable through Scripting APIs

- Method-specific development of functionalities through scripting
- Function calls/APIs of the platform are possible through scripting.

Component APIs

Messageport **Acquisition**
Messageport **Modeling**
Messageport **Analysis**
Messageport **Simulation**
Messageport **Evaluation**
Messageport **ImportExport**
Messageport **Documentation**
Messageport **AQL**

UI APIs

Messageport **AdoScript**
Messageport **CoreUI**
Messageport **Explorer**

Manipulation APIs

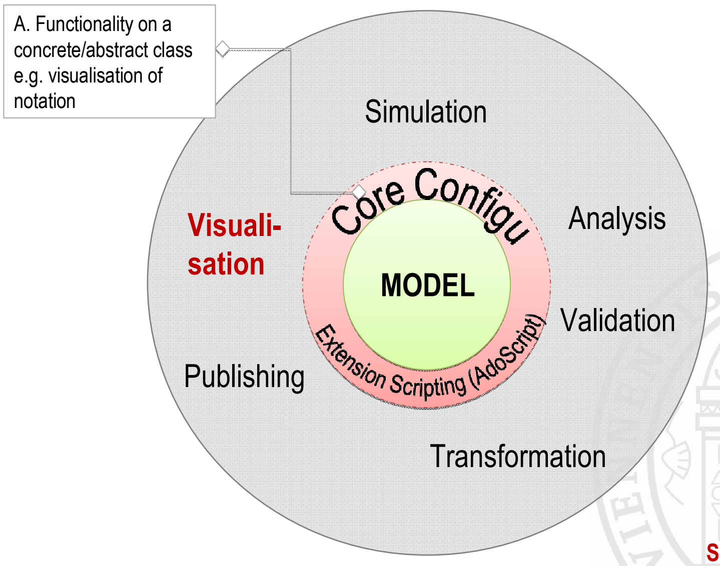
Messageport **Core**
Messageport **DB**
Messageport **UsrMgt**

Application APIs

Messageport **Drawing**
Messageport **Application**

About 400 APIs are available.

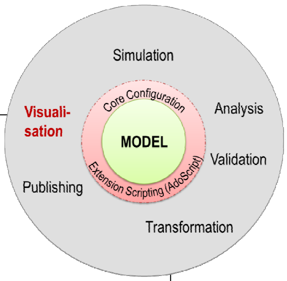
Model Processing Functionality: Visualisation



A. Functionality on a concrete/abstract class e.g. visualisation of notation

Selected Functionalities for Tutorial

Object Visualisation



Platform Functionality

- Object visualisation
- Model visualisation
  - Tabular view incl. view concept
  - Graphical view incl. view concept
  - Machine-generated models
    - Model analysis visualisation
    - Information visualisation
  - Human-generated models
    - Support functionality (automatic & user-defined)

OMiLAB Development Tools

- OMITool GraphRepGenerator
- OMITool AdoScript Syntax Highlighter

OPEN SOURCE

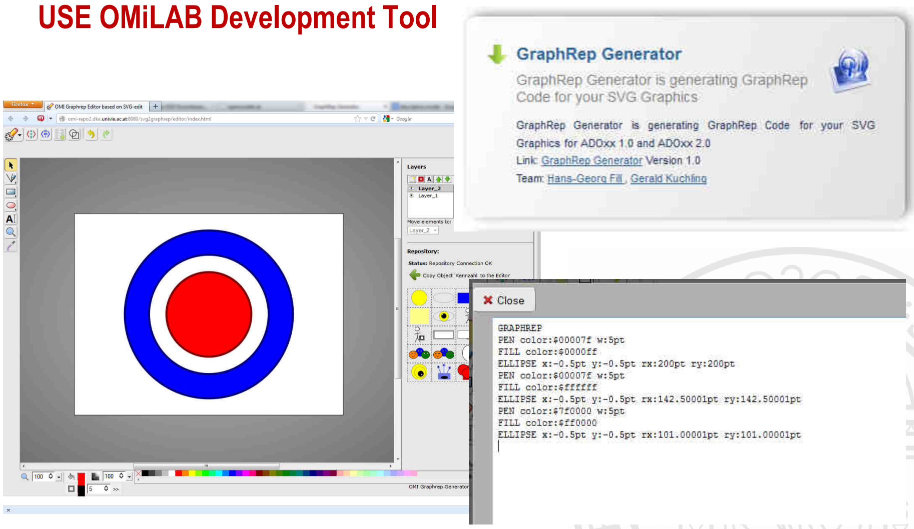
Platform Technologies

- GraphRep
- AdoScript

OPEN USE

Demonstration: Implementation of Object Visualisation

USE OMiLAB Development Tool



Status: Repository Connection OK

Copy Object 'Kennzahl' to the Editor

GraphRep Generator

GraphRep Generator is generating GraphRep Code for your SVG Graphics

GraphRep Generator is generating GraphRep Code for your SVG Graphics for ADOxx 1.0 and ADOxx 2.0

Link: GraphRep Generator Version 1.0

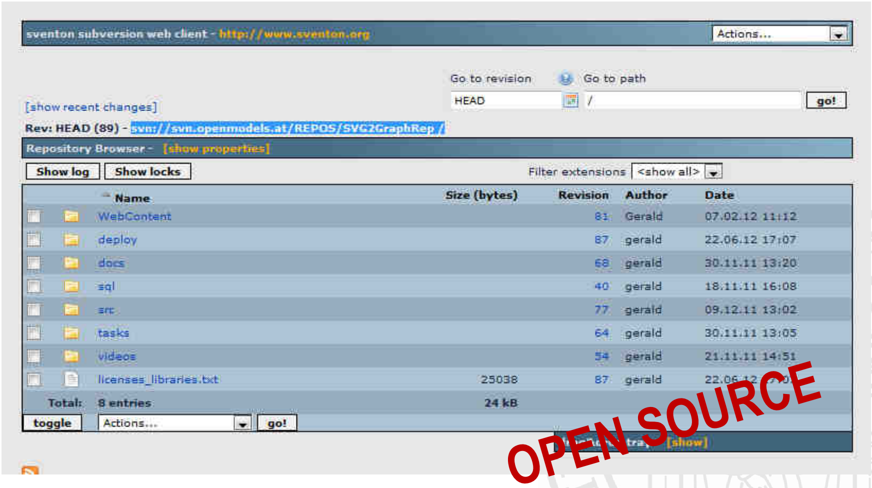
Team: Hans-Georg Fill , Gerald Kuchling

```
GRAPHREP
PEN color:$00007f w:5pt
FILL color:$0000ff
ELLIPSE x:-0.5pt y:-0.5pt rx:200pt ry:200pt
PEN color:$00007f w:5pt
FILL color:$ffffff
ELLIPSE x:-0.5pt y:-0.5pt rx:142.50001pt ry:142.50001pt
PEN color:$7f0000 w:5pt
FILL color:$ff0000
ELLIPSE x:-0.5pt y:-0.5pt rx:101.00001pt ry:101.00001pt
```

http://omi-repo2.dke.univie.ac.at:8080/svg2graphrep/editor/index.html

Demonstration: Implementation of Object Visualisation

CONTRIBUTE to OMiLAB Development



sventon subversion web client - http://www.sventon.org

Go to revision: HEAD

Go to path: /

[show recent changes]

Rev: HEAD (89) - svn://svn.openmodels.at/REPOS/SVG2GraphRep /

Repository Browser - [show properties]

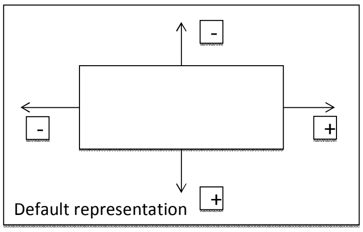
Filter extensions: <show all>

Name	Size (bytes)	Revision	Author	Date
WebContent		81	Gerald	07.02.12 11:12
deploy		87	gerald	22.06.12 17:07
docs		68	gerald	30.11.11 13:20
sql		40	gerald	18.11.11 16:08
src		77	gerald	09.12.11 13:02
tasks		64	gerald	30.11.11 13:05
videos		54	gerald	21.11.11 14:51
licenses_libraries.txt	25038	87	gerald	22.06.12
Total: 8 entries	**24 kB**			

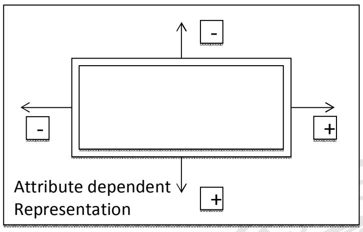
HTTP: http://omi-repo2.dke.univie.ac.at:8080/svg2graphrep/svn.htm
SVN: svn://svn.openmodels.at/REPOS/SVG2GraphRep /

Demonstration: Implementation of Object Visualisation

DEVELOPMENT on ADOxx Platform



Default representation



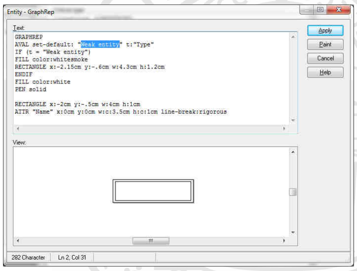
Attribute dependent Representation

```
GRAPHREP
AVAL t:"Type"
IF (t = "Weak entity")
  FILL color:whitesmoke
  RECTANGLE x:-2.15cm y:-.6cm  w:4.3cm h:1.2cm
ENDIF
FILL color:white
PEN solid

RECTANGLE x:-2cm y:-.5cm w:4cm h:1cm
ATTR "Name" x:0cm y:0cm w:c:3.5cm h:c:1cm line-
break:rigorous
```

Conditional representation

Default representation

Name representation



Entity - GraphRep

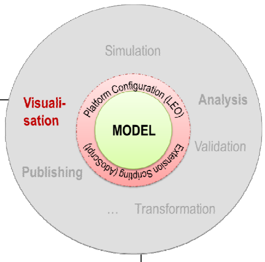
Text:

```
GRAPHREP
AVAL set-default: "Weak entity" t:"Type"
IF (t = "Weak entity")
FILL color:whitesmoke
RECTANGLE x:-2.15cm y:-.6cm w:4.3cm h:1.2cm
ENDIF
FILL color:white
PEN solid

RECTANGLE x:-2cm y:-.5cm w:4cm h:1cm
ATTR "Name" x:0cm y:0cm w:c:3.5cm h:c:1cm line-break:rigorous
```

View:

282 Character | Ln 2, Col 31

Model Analysis Visualisation



Platform Functionality

- Object visualisation
- Model visualisation
    - Tabular view incl. view concept
    - Graphical view incl. view concept
    - Machine-generated models
        - Model analysis visualisation
        - Information visualisation
    - Human-generated models
        - Support functionality (automatic & user-defined)

OMiLAB Development Tools

- OMITool GraphRepGenerator
- OMITool AdoScript Syntax Highlighter

OPEN SOURCE

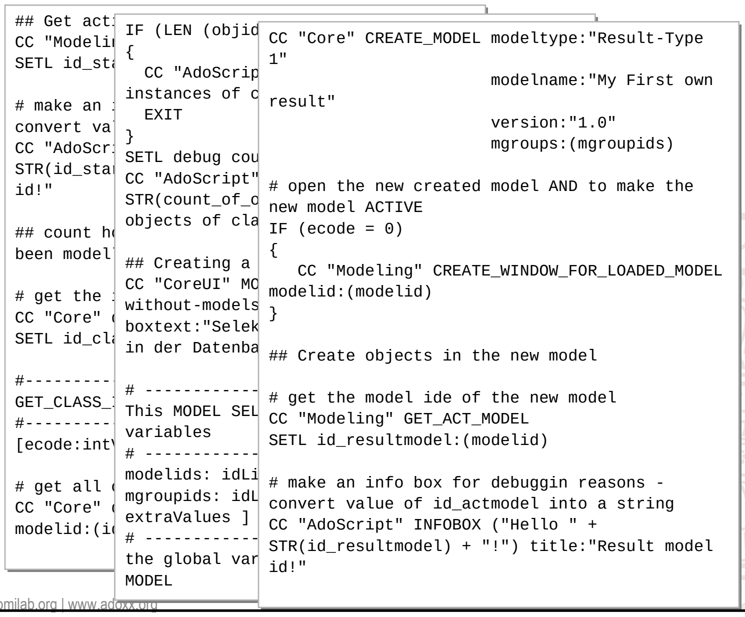
Platform Technologies

- GraphRep
- AdoScript

OPEN USE

Demonstration: Model Analysis Visualisation

DEVELOPMENT on ADOxx Platform

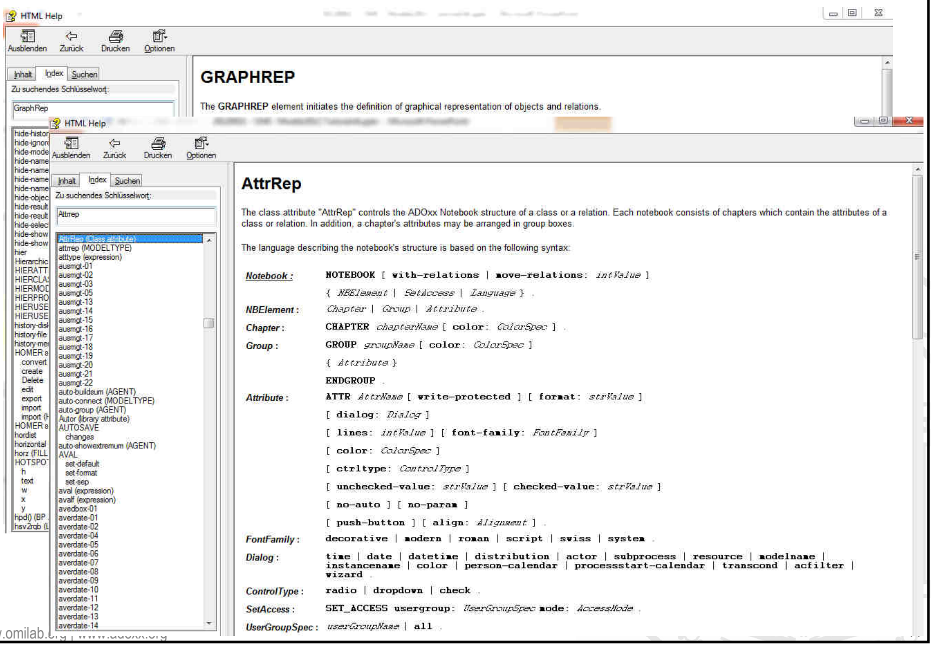


```
## Get acti
CC "Modelir
SETL id_sta

# make an i
convert va
CC "AdoScri
STR(id_star
id!"

## count ho
been model

# get the i
CC "Core" c
SETL id_cla

#---------
GET_CLASS_I
#---------
[ecode:intV

# get all c
CC "Core" c
modelid:(ic
```

```
IF (LEN (objid
{
  CC "AdoScrip
instances of c
  EXIT
}
SETL debug cou
CC "AdoScript"
STR(count_of_o
objects of cla

## Creating a
CC "CoreUI" MO
without-models
boxtext:"Selek
in der Datenba

# ------------
This MODEL SEL
variables
# ------------
modelids: idLi
mgroupids: idL
extraValues ]
# ------------
the global var
MODEL
```

```
CC "Core" CREATE_MODEL modeltype:"Result-Type
1"
                       modelname:"My First own
result"
                       version:"1.0"
                       mgroups:(mgroupids)

# open the new created model AND to make the
new model ACTIVE
IF (ecode = 0)
{
   CC "Modeling" CREATE_WINDOW_FOR_LOADED_MODEL
modelid:(modelid)
}

## Create objects in the new model

# get the model ide of the new model
CC "Modeling" GET_ACT_MODEL
SETL id_resultmodel:(modelid)

# make an info box for debuggin reasons -
convert value of id_actmodel into a string
CC "AdoScript" INFOBOX ("Hello " +
STR(id_resultmodel) + "!") title:"Result model
id!"
```

Summary: GraphRep & AttrRep Syntax



GRAPHREP

The **GRAPHREP** element initiates the definition of graphical representation of objects and relations.

AttrRep

The class attribute "AttrRep" controls the ADOxx Notebook structure of a class or a relation. Each notebook consists of chapters which contain the attributes of a class or relation. In addition, a chapter's attributes may be arranged in group boxes.

The language describing the notebook's structure is based on the following syntax:

```
Notebook :       NOTEBOOK [ with-relations | move-relations: intValue ]
                 { NBElement | SetAccess | Language } .
NBElement :      Chapter | Group | Attribute .
Chapter :        CHAPTER chapterName [ color: ColorSpec ] .
Group :          GROUP groupName [ color: ColorSpec ]
                 { Attribute }
                 ENDGROUP .
Attribute :      ATTR AttrName [ write-protected ] [ format: strValue ]
                 [ dialog: Dialog ]
                 [ lines: intValue ] [ font-family: FontFamily ]
                 [ color: ColorSpec ]
                 [ ctrltype: ControlType ]
                 [ unchecked-value: strValue ] [ checked-value: strValue ]
                 [ no-auto ] [ no-param ]
                 [ push-button ] [ align: Alignment ] .
FontFamily :     decorative | modern | roman | script | swiss | system .
Dialog :         time | date | datetime | distribution | actor | subprocess | resource | modelname |
                 instancename | color | person-calendar | processstart-calendar | transcond | acfilter |
                 wizard .
ControlType :    radio | dropdown | check .
SetAccess :      SET_ACCESS usergroup: UserGroupSpec mode: AccessMode .
UserGroupSpec :  userGroupName | all .
```

More details available on www.omilab.org

GRAPHREP
ATTREP

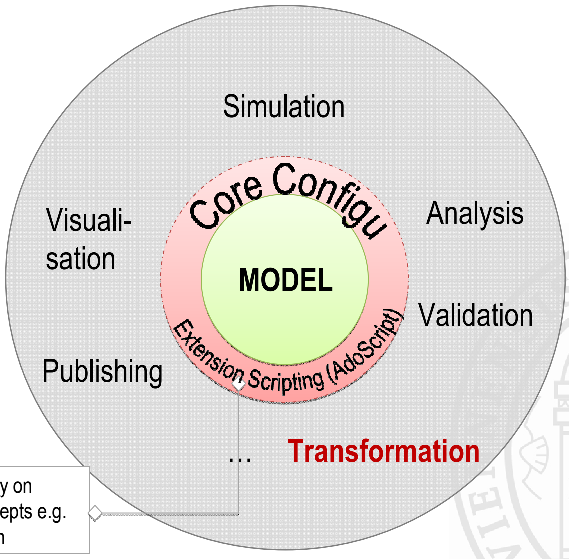
Wrap Up: Visualisation Functionality

- Introduction of visualisation platform functionality
- Definition of tools and services to support development
- Technology Overview to support visualisation functionality

RESULT ACCOMPLISHED:

- Implemented Object Visualisation
- Implemented Script Functionality
- Modeltypes and View Definition
- Attribute Representation

Model Processing Functionality: Transformation



Simulation

Visuali-
sation

Analysis

Core Configu

MODEL

Extension Scripting (AdoScript)

Validation

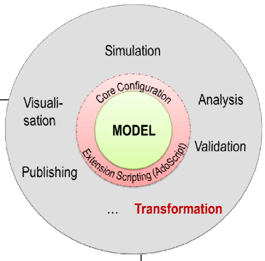
Publishing

…

Transformation

D. Functionality on semantic concepts e.g. Transformation

Model Transformation



Platform Functionality

- Objecttransformation
- Modeltransformation
  - SAME PLATFORM
  - DIFFERENT PLATFORM
    - Inbound Interfaces
    - Outbound Interfaces

OMiLAB Development Tools

- Model Publishing Engine
- AdoScript Syntax Highlighter

OPEN SOURCE

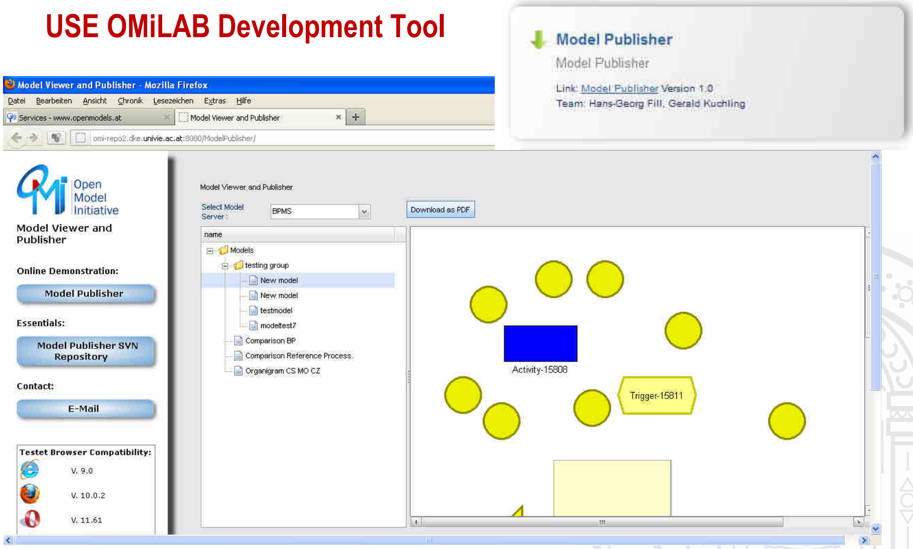
Platform Technologies

- Platform configuration
- AdoScript

OPEN USE

Demonstration: Implementation of Model transformation

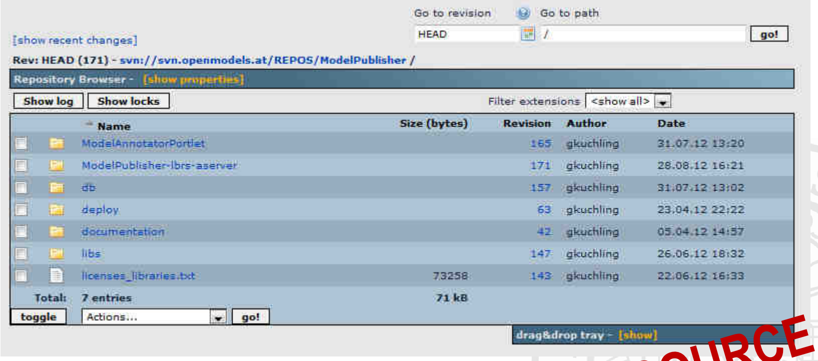
USE OMiLAB Development Tool


Model Publisher
Model Publisher
Link: Model Publisher Version 1.0
Team: Hans-Georg Fill, Gerald Kuchling
Model Viewer and Publisher - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Services - www.openmodels.at
Model Viewer and Publisher
omi-repo2.dke.univie.ac.at:8080/ModelPublisher/
Open Model Initiative
Model Viewer and Publisher
Online Demonstration:
Model Publisher
Essentials:
Model Publisher SVN Repository
Contact:
E-Mail
Testet Browser Compatibility:
V. 9.0
V. 10.0.2
V. 11.61
Model Viewer and Publisher
Select Model Server :
BPMS
Download as PDF
name
Models
testing group
New model
New model
testmodel
modeltest7
Comparison BP
Comparison Reference Process
Organigram CS MO CZ
Activity-15808
Trigger-15811

HTTP: http://omi-repo2.dke.univie.ac.at:8080/ModelPublisher/

Demonstration: Implementation of Model transformation

CONTRIBUTE to OMiLAB Development



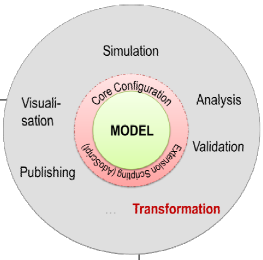
Rev: HEAD (171) - svn://svn.openmodels.at/REPOS/ModelPublisher /

Name	Size (bytes)	Revision	Author	Date
ModelAnnotatorPortlet		165	gkuchling	31.07.12 13:20
ModelPublisher-lbrs-aserver		171	gkuchling	28.08.12 16:21
db		157	gkuchling	31.07.12 13:02
deploy		63	gkuchling	23.04.12 22:22
documentation		42	gkuchling	05.04.12 14:57
libs		147	gkuchling	26.06.12 18:32
licenses_libraries.txt	73258	143	gkuchling	22.06.12 16:33
Total: 7 entries	71 kB			

OPEN SOURCE

HTTP: http://omi-repo2.dke.univie.ac.at:8080/ModelPublisher/
SVN: svn://svn.openmodels.at/REPOS/ModelPublisher /

Model Transformation (Different Platform)



Platform Functionality

- Objecttransformation
- Modeltransformation
    - SAME PLATFORM
    - DIFFERENT PLATFORM
        - Inbound Interfaces
        - Outbound Interfaces

OMiLAB Development Tools

- Model Publishing Engine
- AdoScript Syntax Highlighter

OPEN SOURCE

Platform Technologies

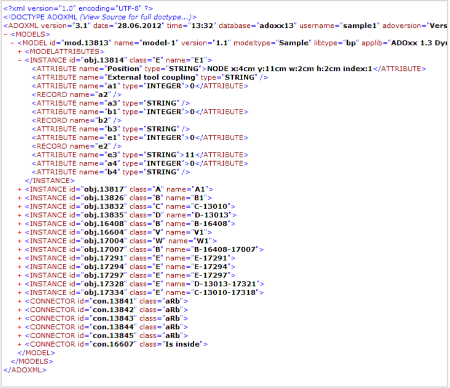
- Platform configuration
- AdoScript

OPEN USE

Demonstration:
Core Functionality for Serialisation as XML and ADL

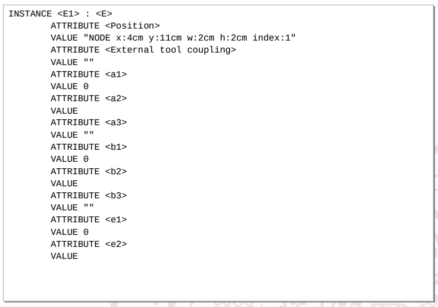
USE functionality on ADOxx Platform

XML Export Sample



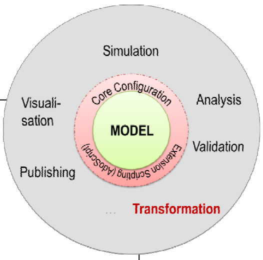
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ADOXML (View Source for full doctype...)>
<ADOXML version="3.1" date="28.06.2012" time="13:32" database="adoxx13" username="sample1" adoversion="Vers
- <MODELS>
  - <MODEL id="mod.13813" name="model-1" version="1.1" modeltype="Sample" libtype="bp" applib="ADOxx 1.3 Dy
    + <MODELATTRIBUTES>
    - <INSTANCE id="obj.13814" class="E" name="E1">
        <ATTRIBUTE name="Position" type="STRING">NODE x:4cm y:11cm w:2cm h:2cm index:1</ATTRIBUTE>
        <ATTRIBUTE name="External tool coupling" type="STRING" />
        <ATTRIBUTE name="a1" type="INTEGER">0</ATTRIBUTE>
        <RECORD name="a2" />
        <ATTRIBUTE name="a3" type="STRING" />
        <ATTRIBUTE name="b1" type="INTEGER">0</ATTRIBUTE>
        <RECORD name="b2" />
        <ATTRIBUTE name="b3" type="STRING" />
        <ATTRIBUTE name="e1" type="INTEGER">0</ATTRIBUTE>
        <RECORD name="e2" />
        <ATTRIBUTE name="e3" type="STRING">11</ATTRIBUTE>
        <ATTRIBUTE name="a4" type="INTEGER">0</ATTRIBUTE>
        <ATTRIBUTE name="b4" type="STRING" />
      </INSTANCE>
    + <INSTANCE id="obj.13817" class="A" name="A1">
    + <INSTANCE id="obj.13826" class="B" name="B1">
    + <INSTANCE id="obj.13832" class="C" name="C-13010">
    + <INSTANCE id="obj.13835" class="D" name="D-13013">
    + <INSTANCE id="obj.16408" class="B" name="B-16408">
    + <INSTANCE id="obj.16604" class="V" name="V1">
    + <INSTANCE id="obj.17004" class="W" name="W1">
    + <INSTANCE id="obj.17007" class="B" name="B-16408-17007">
    + <INSTANCE id="obj.17291" class="E" name="E-17291">
    + <INSTANCE id="obj.17294" class="E" name="E-17294">
    + <INSTANCE id="obj.17297" class="E" name="E-17297">
    + <INSTANCE id="obj.17328" class="E" name="D-13013-17321">
    + <INSTANCE id="obj.17334" class="E" name="C-13010-17318">
    + <CONNECTOR id="con.13841" class="aRb">
    + <CONNECTOR id="con.13842" class="aRb">
    + <CONNECTOR id="con.13843" class="aRb">
    + <CONNECTOR id="con.13844" class="aRb">
    + <CONNECTOR id="con.13845" class="aRb">
    + <CONNECTOR id="con.16607" class="Is inside">
    </MODEL>
  </MODELS>
</ADOXML>
```

ADL Export Sample



```
INSTANCE <E1> : <E>
        ATTRIBUTE <Position>
        VALUE "NODE x:4cm y:11cm w:2cm h:2cm index:1"
        ATTRIBUTE <External tool coupling>
        VALUE ""
        ATTRIBUTE <a1>
        VALUE 0
        ATTRIBUTE <a2>
        VALUE
        ATTRIBUTE <a3>
        VALUE ""
        ATTRIBUTE <b1>
        VALUE 0
        ATTRIBUTE <b2>
        VALUE
        ATTRIBUTE <b3>
        VALUE ""
        ATTRIBUTE <e1>
        VALUE 0
        ATTRIBUTE <e2>
        VALUE
```

Model Transformation (Same Platform)



Platform Functionality

- •Objecttransformation
- •Modeltransformation
  - SAME PLATFORM
  - DIFFERENT PLATFORM
    - Inbound Interfaces
    - Outbound Interfaces

OMiLAB Development Tools

- •Model Publishing Engine
- •AdoScript Syntax Highlighter

OPEN SOURCE

Platform Technologies

- •Platform configuration
- •AdoScript

OPEN USE

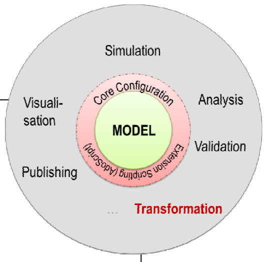
Demonstration: Implementation of Model transformation 1
DEVELOP on ADOxx Platform

```
## Open Model
CC "Modeling" GET_ACT_MODEL
SETL id_source_model:(modelid)

SETL s_classname_source:("A")
SETL s_classname_target:("E")

# BEGIN set new model
CC "CoreUI" MODEL_SELECT_BOX mgroup-sel without-models title:"Zielmodellgruppe"
                             boxtext:"Selektieren Sie die Ziel-Modellgruppe in der
    Datenbank:"

CC "Core" CREATE_MODEL modeltype:"Sample"
                       modelname:"My First sample"
                       version:"1.0"
                       mgroups:(mgroupids)
SETL id_target_model:(modelid)

# END set new model

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(id_source_model)
    classname:(s_classname_source)
SETL id_objects:(objids)
```

Demonstration: Implementation of Model transformation 2
DEVELOP on ADOxx Platform

```
# BEGIN set x, y pos
SETL xoffset:5cm
SETL yoffset:5cm
SETL xpos:5.0cm
SETL ypos:5.0cm
SETL counter:1
FOR id_object in:(id_objects)
{
    # get class ID from class name
    CC "Core" GET_CLASS_ID classname:(s_classname_source)

    # get all Notebook attributes
    CC "Core" GET_ALL_NB_ATTRS classid:(classid)

    # and show them
    CC "AdoScript" INFOBOX (attrids)
    CC "Core" GET_ATTR_VAL objid:(VAL (id_object)) attrid:(VAL ("9"))
    SETL s_attr_name:(val)

    # Make new model
    CC "Core" GET_CLASS_ID classname:(s_classname_target)
    SETL id_class_target:(classid)

    CC "Core" debug CREATE_OBJ modelid:(id_target_model) classid:(id_class_target)
    objname:(s_attr_name)
    CC "Modeling" debug SET_OBJ_POS objid:(objid) x:"5cm" y:"5cm"
}
```

Object Transformation



Platform Functionality

- Objecttransformation
- Modeltransformation
  - SAME PLATFORM
  - DIFFERENT PLATFORM
    - Inbound Interfaces
    - Outbound Interfaces

OMiLAB Development Tools

- Model Publishing Engine
- AdoScript Syntax Highlighter

OPEN SOURCE

Platform Technologies

- Platform configuration
- AdoScript

OPEN USE

Objecttransformation using CONVERSION

DEVELOP on ADOxx Platform

- If you define __*Conversion*__ for the class “A” with

  CLASS “B”

  ATTR “ba1”

  ATTR “ba2” from: “aa3”

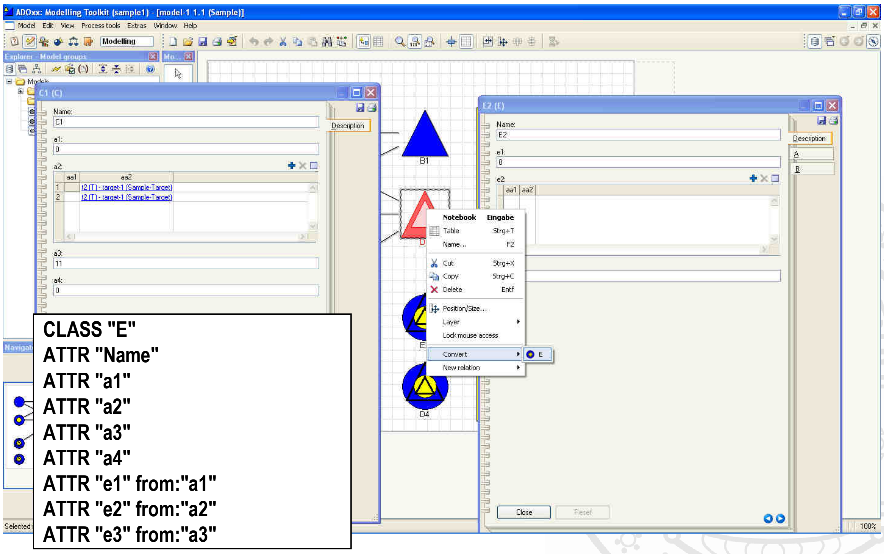
Conversion :	{ *ClassConversion* } .
ClassConversion :	**CLASS** *className* { *AttrConversion* } .
AttrConversion :	**ATTR** *attrName* [**from:***attrName*] .

- this means that
  - objects of class “A” can be converted to objects of class “B”,
  - the aa1 is assigned from A to ba1 in B as the have the same name,
  - the aa3 from A is assigned to Ba2 from B as they have different names,

Demonstration: Objecttransformation

DEVELOP on ADOxx Platform

Instances of C->E



C1 (C)

Name: C1

a1: 0

a2:

	aa1	aa2
1		t2 (T) - target-1 (Sample-Target)
2		t2 (T) - target-1 (Sample-Target)

a3: 11

a4: 0

E2 (E)

Name: E2

e1: 0

e2:

aa1	aa2

CLASS "E"
ATTR "Name"
ATTR "a1"
ATTR "a2"
ATTR "a3"
ATTR "a4"
ATTR "e1" from:"a1"
ATTR "e2" from:"a2"
ATTR "e3" from:"a3"

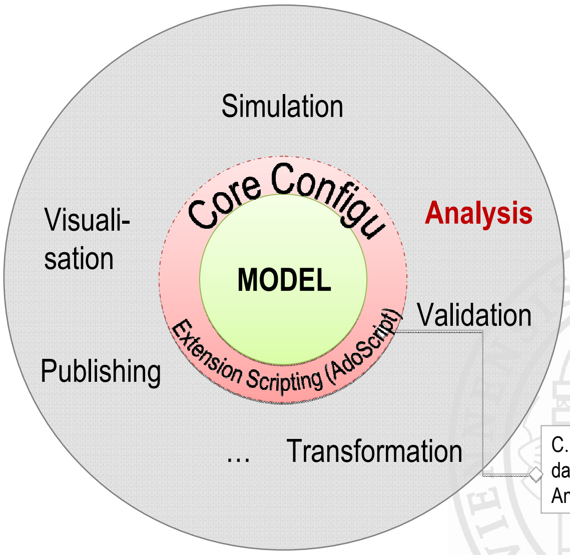
Wrap Up: Transformation Functionality

- Use the basic transformation mechanisms to use it for publishing
- Use scripting mechanisms for model transformation

RESULT ACCOMPLISHED:

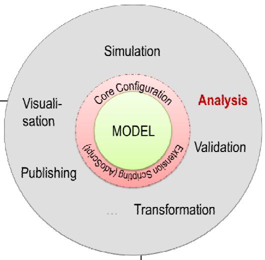
- Publishing example using the OMiLAB service
- Transformation of scripts

Model Processing Functionality: Analysis



C. Functionalities on data of classes e.g. Analysis/Queries

Query of Model Content



Platform Functionality

- Query of model content
- Quality validation
- Consistency checks
- Population through analysis

OMiLAB Development Tools

- ReST API

OPEN SOURCE

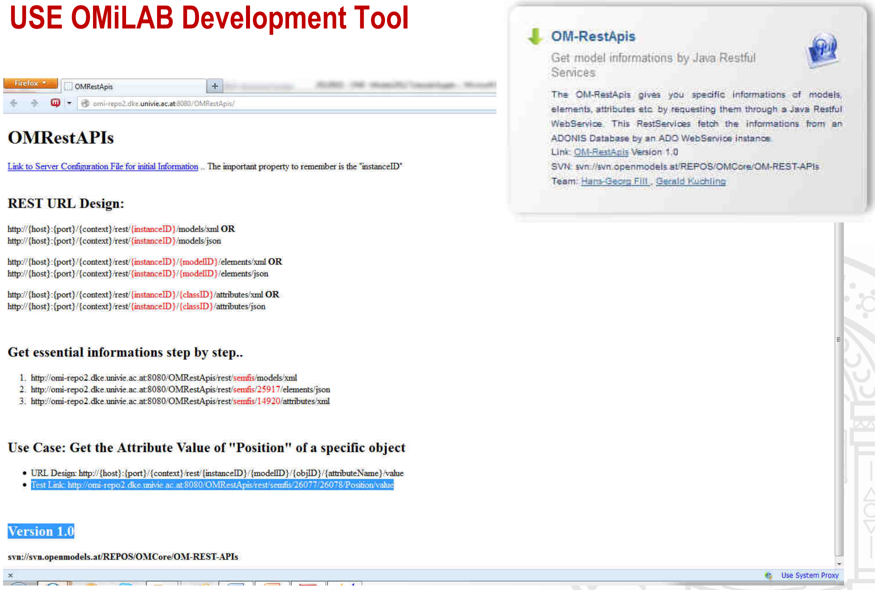
Platform Technologies

- AQL
- AdoScript

OPEN USE

Demonstration: Analysis Functionality

USE OMiLAB Development Tool



OMRestAPIs

Link to Server Configuration File for initial Information .. The important property to remember is the "instanceID"

REST URL Design:

http://{host}:{port}/{context}/rest/{instanceID}/models/xml **OR**
http://{host}:{port}/{context}/rest/{instanceID}/models/json

http://{host}:{port}/{context}/rest/{instanceID}/{modelID}/elements/xml **OR**
http://{host}:{port}/{context}/rest/{instanceID}/{modelID}/elements/json

http://{host}:{port}/{context}/rest/{instanceID}/{classID}/attributes/xml **OR**
http://{host}:{port}/{context}/rest/{instanceID}/{classID}/attributes/json

Get essential informations step by step..

1. http://omi-repo2.dke.univie.ac.at:8080/OMRestApis/rest/semfis/models/xml
2. http://omi-repo2.dke.univie.ac.at:8080/OMRestApis/rest/semfis/25917/elements/json
3. http://omi-repo2.dke.univie.ac.at:8080/OMRestApis/rest/semfis/14920/attributes/xml

Use Case: Get the Attribute Value of "Position" of a specific object

- URL Design: http://{host}:{port}/{context}/rest/{instanceID}/{modelID}/{objID}/{attributeName}/value
- Test Link: http://omi-repo2.dke.univie.ac.at:8080/OMRestApis/rest/semfis/26077/26078/Position/value

Version 1.0

svn://svn.openmodels.at/REPOS/OMCore/OM-REST-APIs

OM-RestApis

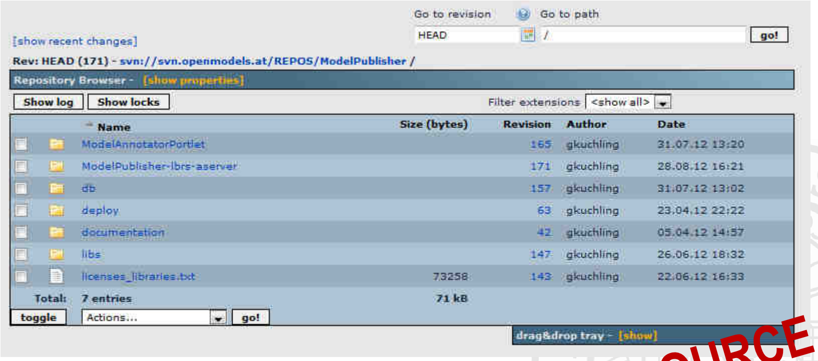
Get model informations by Java Restful Services

The OM-RestApis gives you specific informations of models, elements, attributes etc. by requesting them through a Java Restful WebService. This RestServices fetch the informations from an ADONIS Database by an ADO WebService instance.
Link: OM-RestApis Version 1.0
SVN: svn://svn.openmodels.at/REPOS/OMCore/OM-REST-APIs
Team: Hans-Georg Fill , Gerald Kuchling

HTTP: svn://svn.openmodels.at/REPOS/OMCore/OM-REST-APIs

Demonstration: Analysis Functionality

CONTRIBUTE to OMiLAB Development



Rev: HEAD (171) - svn://svn.openmodels.at/REPOS/ModelPublisher /

	Name	Size (bytes)	Revision	Author	Date
☐	ModelAnnotatorPortlet		165	gkuchling	31.07.12 13:20
☐	ModelPublisher-lbrs-aserver		171	gkuchling	28.08.12 16:21
☐	db		157	gkuchling	31.07.12 13:02
☐	deploy		63	gkuchling	23.04.12 22:22
☐	documentation		42	gkuchling	05.04.12 14:57
☐	libs		147	gkuchling	26.06.12 18:32
☐	licenses_libraries.txt	73258	143	gkuchling	22.06.12 16:33
Total:	7 entries	71 kB			

OPEN SOURCE

HTTP: svn://svn.openmodels.at/REPOS/OMCore/OM-REST-APIs
SVN: svn://svn.openmodels.at/REPOS/OMCore/OM-REST-APIs

Demonstration: Analysis Functionality

DEVELOP on ADOxx Platform

Example 1: Get all objects of class "A" in a certain model

```
CC "AQL" EVAL_AQL_EXPRESSION expr:"<\"A\">" modelid:(modelid)

IF (ecode = 0)
{
    CC "AdoScript" INFOBOX ("Found objects: " + objids)
}
ELSE
{
    CC "AdoScript" INFOBOX "An error has occured!"
}
```

Example 2: Get all models of modeltype "Working Environment Model"

```
CC "AQL" EVAL_AQL_EXPRESSION expr:"<\„Sample\">" modelscope
IF (ecode = 0)
{
    CC "AdoScript" INFOBOX ("Found models: " + objids)
}
ELSE
{
    CC "AdoScript" INFOBOX "An error has occured!"
}
```

More details available on www.omilab.org

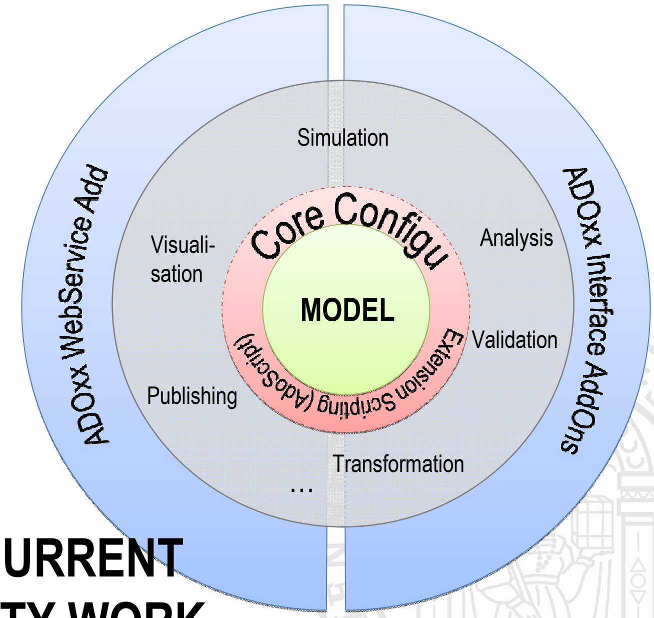
AQL

Wrap Up: Analysis Functionality

- ReST API for model analysis
- Script <-> AQL combination to run analysis

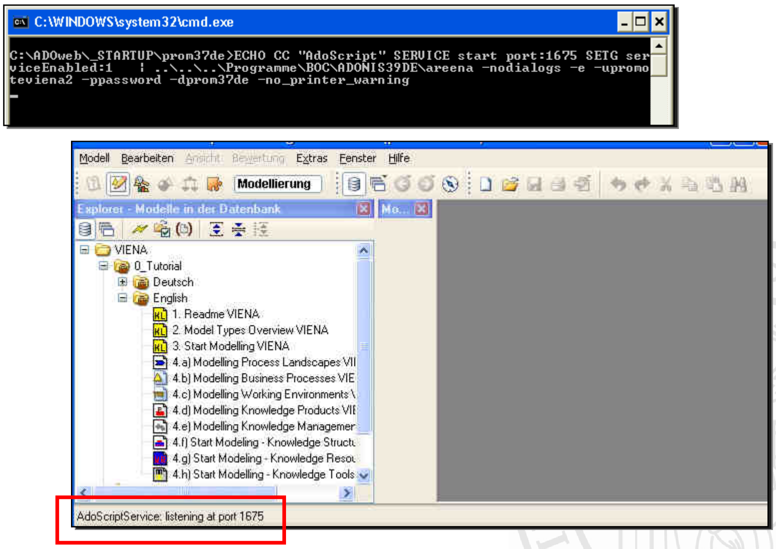
RESULT ACCOMPLISHED:

- Implemented API integration with demonstration environment
- AQL queries in script



ADOXX
SELECT CURRENT COMMUNITY WORK

ADOxx WebService Port

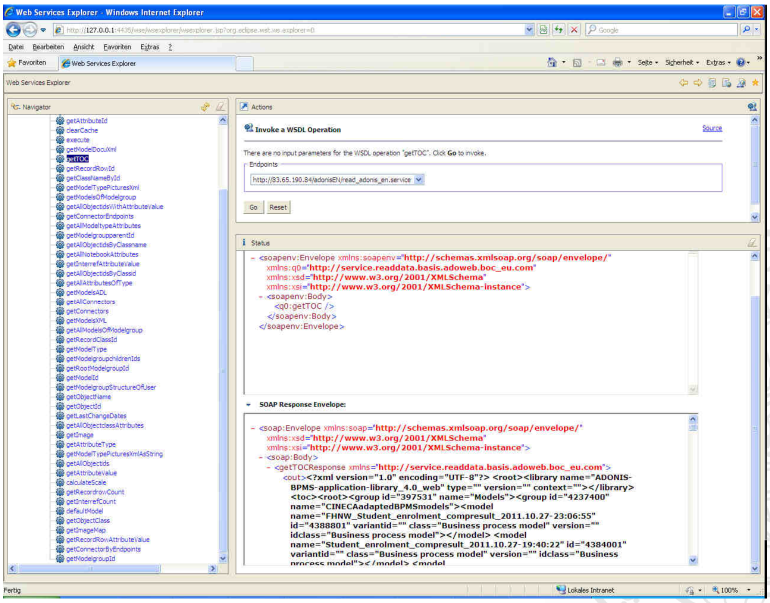


```
C:\ADOweb\_STARTUP\prom37de>ECHO CC "AdoScript" SERVICE start port:1675 SETG ser
viceEnabled:1    ¦ ..\..\..\Programme\BOC\ADONIS39DE\areena -nodialogs -e -upromo
teviena2 -ppassword -dprom37de -no_printer_warning
_
```

AdoScriptService: listening at port 1675

C:\WINDOWS\system32\cmd.exe
Modell
Bearbeiten
Ansicht
Bewertung
Extras
Fenster
Hilfe
Modellierung
Explorer - Modelle in der Datenbank
Mo...
VIENA
0_Tutorial
Deutsch
English
1. Readme VIENA
2. Model Types Overview VIENA
3. Start Modelling VIENA
4.a) Modelling Process Landscapes VII
4.b) Modelling Business Processes VIE
4.c) Modelling Working Environments \
4.d) Modelling Knowledge Products VII
4.e) Modelling Knowledge Managemer
4.f) Start Modeling - Knowledge Structu
4.g) Start Modeling - Knowledge Resou
4.h) Start Modelling - Knowledge Tools

ADOxx WebService Interaction



Invoke a WSDL Operation

There are no input parameters for the WSDL operation "getTOC". Click **Go** to invoke.

Endpoints: http://83.65.190.84/adonisEN/read_adonis_en.service

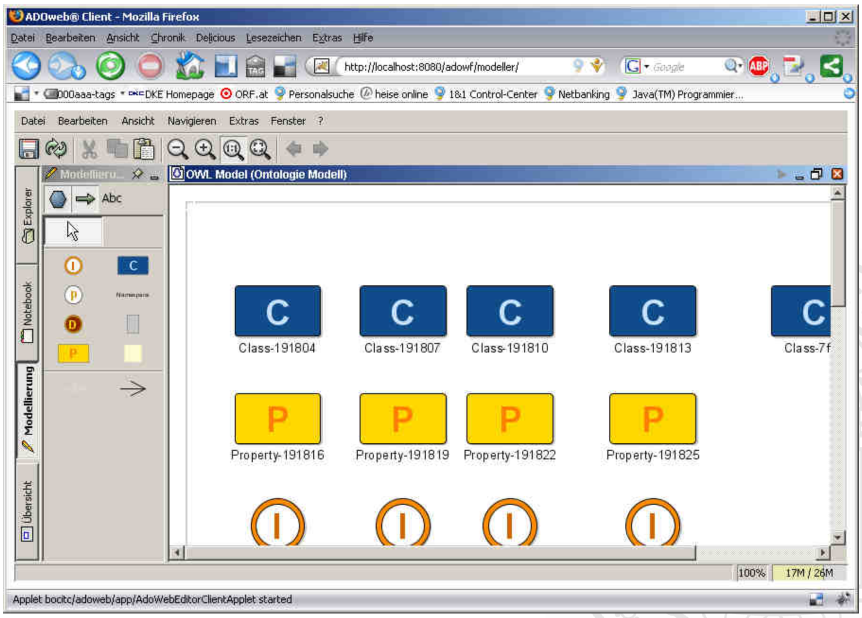
Status

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:q0="http://service.readdata.basis.adoweb.boc_eu.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <soapenv:Body>
      <q0:getTOC />
    </soapenv:Body>
  </soapenv:Envelope>
```

SOAP Response Envelope:

```
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <soap:Body>
    - <getTOCResponse xmlns="http://service.readdata.basis.adoweb.boc_eu.com">
        <out><?xml version="1.0" encoding="UTF-8"?> <root><library name="ADONIS-
          BPMS-application-library_4.0_web" type="" version="" context=""></library>
          <toc><root><group id="397531" name="Models"><group id="4237400"
          name="CINECAadaptedBPMSmodels"><model
          name="FHNW_Student_enrolment_compresult_2011.10.27-23:06:55"
          id="4388801" variantid="" class="Business process model" version=""
          idclass="Business process model"></model> <model
          name="Student_enrolment_compresult_2011.10.27-19:40:22" id="4384001"
          variantid="" class="Business process model" version="" idclass="Business
          process model"></model> <model
```

ADOxx Web Modeller


ADOweb® Client - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Delicious Lesezeichen Extras Hilfe
http://localhost:8080/adowf/modeller/
Google
000aaa-tags DKE Homepage ORF.at Personalsuche heise online 1&1 Control-Center Netbanking Java(TM) Programmier...
Datei Bearbeiten Ansicht Navigieren Extras Fenster ?
Modellieru...
Abc
Explorer
Notebook
Modellierung
Übersicht
OWL Model (Ontologie Modell)
C
Class-191804
C
Class-191807
C
Class-191810
C
Class-191813
C
Class-7f
P
Property-191816
P
Property-191819
P
Property-191822
P
Property-191825
I
100%
17M / 26M

Applet bocitc/adoweb/app/AdoWebEditorClientApplet started

CONCLUSION

The Method Conceptualisation Process …

- Capturing of fundamental concepts, relationships in between and properties adhering to them, usually obtained through the analysis of a selected domain.
- Description of such conceptualisations varies depending on the addressed audience, with different expectations, like End User, Modeller, Developer, ….
- From a development perspective, a method conceptualization needs to be formal enough to enable developer continue along the life-cycle
- A model of the method (language)
  that facilitates a coherent view on the core concepts involved

---

…results in a Modelling Method Tool

- When the realisation of a modelling method is expected to result in an application software/tool, a domain expert’s (i.e., method developer) viewpoint need to be “augmented” with the viewpoint of a software developer (i.e., method engineer).
- Typically, a method developer rarely considers design, implementation or deployment relevant artefacts when “conceptualising” a modelling method.
- A method engineer on the other hand is usually not an expert in the domain that is addressed by a certain modelling method.

Further Questions?

Dimitris Karagiannis
dk@dke.univie.ac.at

Hans-Georg Fill
hans-georg.fill@dke.univie.ac.at

Srdjan Zivkovic
srdjan.zivkovic@dke.univie.ac.at

Wilfrid Utz
wilfrid.utz@dke.unive.ac.at

OMILAB®
www.omilab.org

www.adoxx.org
tutorial@adoxx.org

REFERENCES

OMiLAB References on the ADOxx Meta Modeling Approach

- Fill, H.-G., T. Redmond, et al. (2012). FDMM: A Formalism for Describing ADOxx Meta Models and Models. Proceedings of ICEIS 2012, Wroclaw, Poland. L. Maciaszek, A. Cuzzocrea and J. Cordeiro. 3: 133-144.
- Junginger, S., H. Kühn, et al. (2000). "Ein Geschäftsprozeßmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen (German: ADONIS: A next generation business process management tool - Concepts and Applications)." Wirtschaftsinformatik 42(5): 392-401.
- Karagiannis, D., Visic, N. (2011). Next Generation of Modelling Platforms, BIR Conference 2011, Riga, Latvia, 6.
- Karagiannis D. (2012). Modelling Aspects for Next-Generation Modelling Systems. Presentation at FInES "Translating Knowledge Into Growth: Views from ICT Research to Support Future Business Innovation" - Panel 3 Next Generation Enterprise Systems - Characteristics, Properties, and Architectural Design Principles. Aalborg May 9, 2012. URL: http://de.slideshare.net/FInESCluster/p3-2dimitris-karagiannisv2
- Kühn, H. and S. Junginger (1999). An Approach to use UML for Business Process Modeling and Simulation in ADONIS. Proceedings of the 13th European Simulation Multiconference (ESM 99) - Modeling and Simulation: A Tool for the Next Millenium. H. Szczerbicka. Warsaw, Poland: 634-639.
- Kühn H. (2004). Methodenintegration im Business Engineering, PhD Thesis, University of Vienna, April 2004.
- Schwab, M., D. Karagiannis, et al. (2010). i* on ADOxx(R): A Case Study. Proceedings of the 4th International i* Workshop - iStar10 - CAiSE Workshop Proceedings, Springer.
- Utz W., Woitsch, R. and Karagiannis D. (2011): Conceptualisation of Hybrid Service Models: An Open Models Approach, The 5th International IEEE Workshop on Requirements Engineering for Services (REFS 2011), Munich, IEEE
- Wende, C., Assmann, U., Zivkovic, S., and Kühn, H. (2011). Feature-based Customisation of Tool Environments for Model-Driven Software Development. In: Almeida, E., Kishi, T., Schwanninger, C., John, I., and Schmid, K. (Eds.): Proceedings of the 15th International Software Product Lines Conference (SPLC 2011), Munich, August 21-26, 2011, IEEE Computer Society, pp. 45-54.
- Zivkovic, S.; Kühn, H.; Murzek, M.: An Architecture of Ontology-aware Metamodelling Platforms for Advanced Enterprise Repositories. In: Camp, O.; Hammoudi, S.; (Eds.): Proceedings of the 1st International Workshop on Advanced Enterprise Repositories (AER 2009), Milano, Italy, May 6th, 2009. pp. 95-104.

OMiLAB References on General Aspects of Modelling and Modelling Methods

- Bork, D. and E. Sinz (2010). Design of a SOM Business Process Modelling Tool based on the ADOxx meta-modelling Platform. Proceedings of the Fourth International Workshop on Graph-Based Tools, EASST.
- Fill, H.-G. (2009). Visualisation for Semantic Information Systems, Gabler.
- Fill, H.-G. (2011). On the Conceptualization of a Modeling Language for Semantic Model Annotations. Advanced Information Systems Engineering Workshops, CAiSE 2011. C. Salinesi and O. Pastor. London, UK, Springer. LNBIP Vol. 83: 134-148.
- Fill, H.-G. (2012). An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations. European Conference on Information Systems 2012, AIS.
- Fill, H.-G. (2012a). SeMFIS: A Tool for Managing Semantic Conceptual Models. Workshop on Graphical Modeling Language Development, Kgs. Lyngby, Denmark.
- Karagiannis, D. and H. Kühn (2002). Metamodeling Platforms. Third International Conference EC-Web 2002 – Dexa 2002. K. Bauknecht, A. Min Tjoa and G. Quirchmayr. Aix-en-Provence, France, Springer: 182.
- Karagiannis, D. and P. Höfferer (2006). Metamodels in Action: An Overview. ICSOFT 2006 - First International Conference on Software and Data Technologies. J. Filipe, B. Shishkov and M. Helfert. Setúbal, Insticc Press: IS-27-IS-36.
- Karagiannis, D., W. Grossmann, et al. (2008) "Open Model Initiative - A Feasibility Study.“, URL: http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study _SEPT_2008.pdf
- Karagiannis, D., H.-G. Fill, et al. (2008). Metamodeling: Some Application Areas in Information Systems. UNISCON. R. Kaschek and et al., Springer: 175-188.
- Karagiannis, D., Moser, C., Mostashari, A. (2012): “Compliance Evaluation with Heatmaps”, accepted for CAiSE 2012, Gdańsk, Poland, 25th – 29th June, 2012.
- Kühn, H. and M. Murzek (2005). Modelling: From Craftsmanship to Automation. Proceedings of the 4th International Conference on Business Informatics Research (BIR 2005). P. Backlund, S. Carlsson and E. Soederstroem.
- Rausch, T.; Kühn, H.; Murzek, M.; Brennan, T. (2011). BPMN for Business Professionals: Making BPMN 2.0 Fit for Full Business Use. In: Shapiro, R.; White, A. S.; Palmer, N.; zur Muehlen, M.; Allweyer, T.; Gagné, D. et al (Eds.): BPMN 2.0 Handbook, 2011, Future Strategies, pp. 167-180.

OMiLAB References on Industrial Applications of Modelling

- Abazi, F., H.-G. Fill, et al. (2011). Formalising Knowledge-intensive Nuclear Fuel Process Models Using Pattern Theory. KSEM 2011, Springer.
- Fill, H.-G., A. Gericke, et al. (2007). Modeling for Integrated Enterprise Balancing (German: Modellierung für Integrated Enterprise Balancing). Wirtschaftsinformatik 06/2007: 419-429.
- Fill, H.-G., A. Eberhart, et al. (2011). An Approach to Support the Performance Management of Public Health Authorities using an IT based Modeling Method. Proceedings of the 10th International Conference on Wirtschaftsinformatik WI 2.011, Zürich, CH.
- Karagiannis, D., J. Mylopoulos, et al. (2007). Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. 15th IEEE International RE Conference, 315-321, IEEE.
- Karagiannis, D., F. Ronaghi, et al. (2007). Business-oriented IT management: Developing e-business applications with E-BPMS. ICEC 2008, ACM.
- Kühn, H., Murzek, M., Specht, G., Zivkovic, S. (2010). Model-Driven Development of Interoperable, Inter-Organisational Business Processes, in: Yannis Charalabidis (Ed.), "Interoperability in the Digital Public Services and Administration: Bridging E-Government and E-Business", pp. 119-143, IGI Global.
- Orensanz, D. et al. (2011): D2.2: Immigration Policy 2.0 Definition of Services and Service Bundles, Public deliverable: http://www.immigrationpolicy2.eu/trunk/Deliverables/D.2.2_Definition%20of%20 Services%20and%20Service%20Bundles_v1.1.pdf (access 24-09-2012)
- Zivkovic, S., Miksa, K., and Kühn, H. (2011). A Modelling Method for Consistent Physical Devices Management: An ADOxx Case Study. In: Salinesi, C., Pastor, O. (Eds.): Proceedings of Advanced Information Systems Engineering Workshops. 1st International Workshop on Conceptualization of Modelling Methods (CMM 2011). 2011, LNBIP 83, Springer, pp. 104-118.

Additional References

- Harel, D. and B. Rumpe (2000). Modeling Languages: Syntax, Semantics and All That Stuff - Part I: The Basic Stuff. Rehovot, Israel, The Weizmann Institute of Science: 28p.
- Harel, D. and B. Rumpe (2004). "Meaningful Modeling: What's the Semantics of "Semantics"?" IEEE Computer October 2004: 64-72.
- Koch, S., S. Strecker, et al. (2006). Conceptual Modelling as a New Entry in the Bazaar: The Open Model Approach. Open Source Systems, IFIP International Federation for Information Processing. 203/2006: 9-20.
- Schmidt, D. (2006). "Model-Driven Engineering." IEEE Computer 39(2): 25-34.
- Schütte, R. and J. Becker (1998). Subjektivitätsmanagement bei Informationsmodellen (German: Management of subjectivity for information models). Modellierung 98, Münster, GI-Workshop.
- Schütte, R. and T. Rotthowe (1998). The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. ER'98. T. W. Ling, S. Ram and L. M. Lee, Springer: 240-254.
- Stachowiak, H. (1973). Allgemeine Modelltheorie (German: General Model Theory), Springer.