

Theoretical Analysis of Creative Service Management  
and its Application to Open model Initiative



www.openmodels.at

# OMI: The Implementation of the JCS-Toolkit on the ADOxx Framework

## ADOxx Introduction

Workshop  
Kyoto



universität  
wien

universität  
wien  
Faculty of Computer Science

o.Univ.-Prof. Dr. Dimitris Karagiannis

**DKE**

What is ADOxx®?

**ADOxx® is a metamodelling  
development and configuration  
platform for implementing  
modelling methods.**

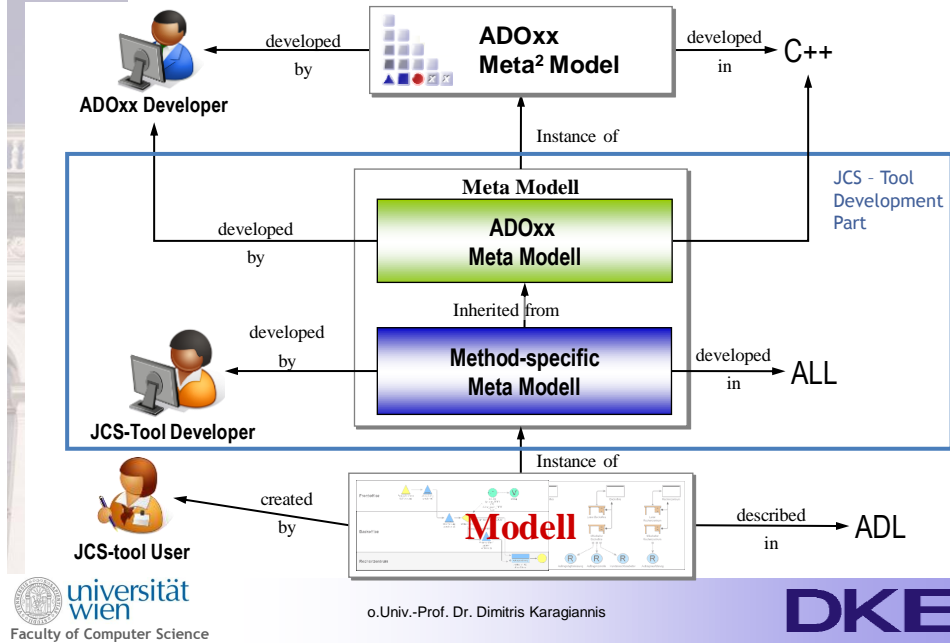
universität  
wien  
Faculty of Computer Science

o.Univ.-Prof. Dr. Dimitris Karagiannis

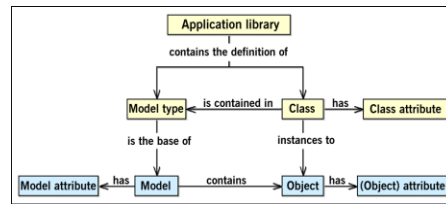
**DKE**

2

## Metamodelling Platforms Hierarchy: ADOxx®



## Introduction of ADOxx model types, classes, attributes and relations



### 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"

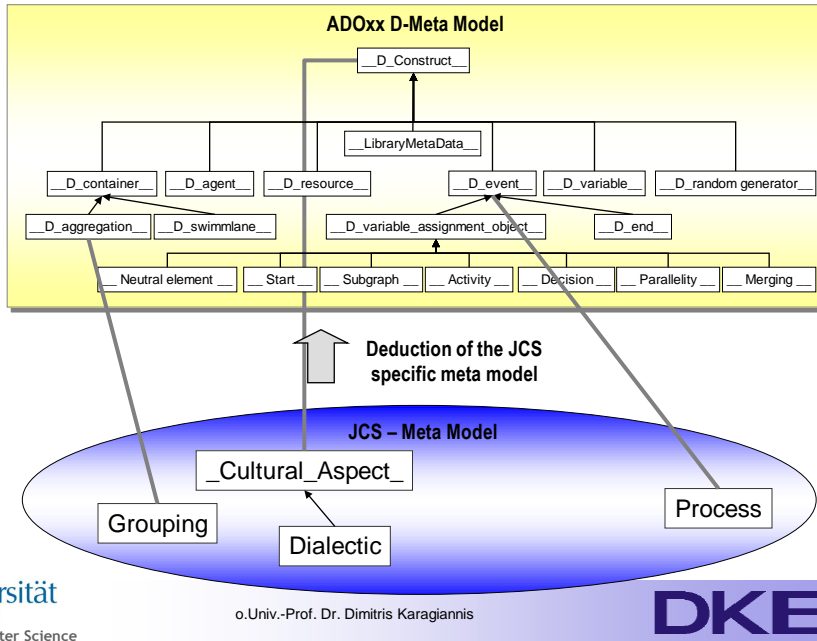
### Attributes

- 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.

## Application Library Language (ALL)



## Excursion: Application Library Language (ALL) Definition 1/3

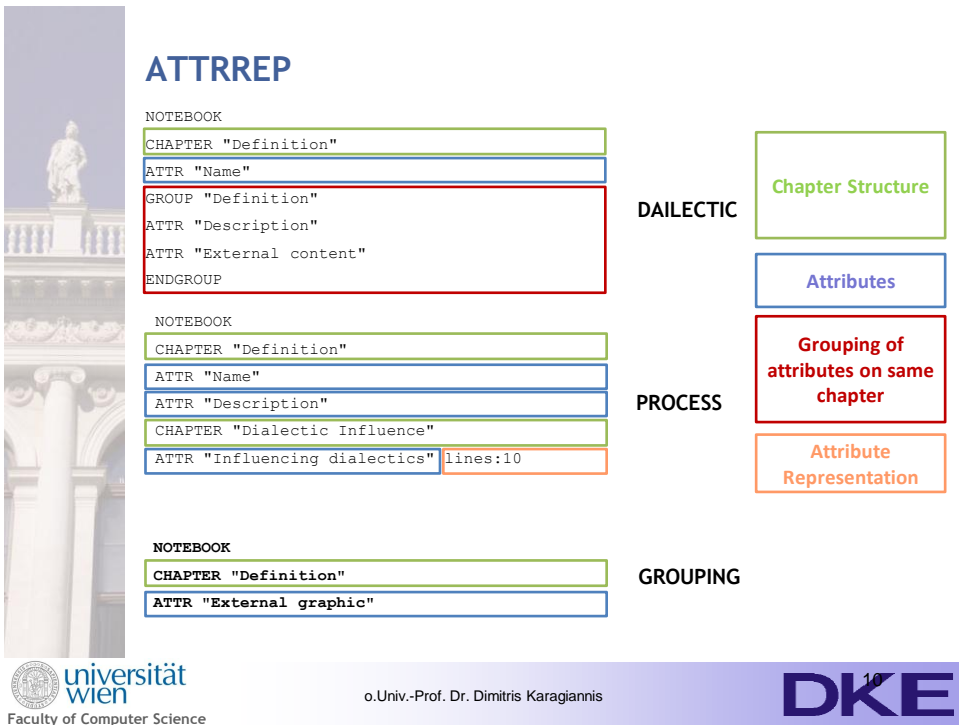
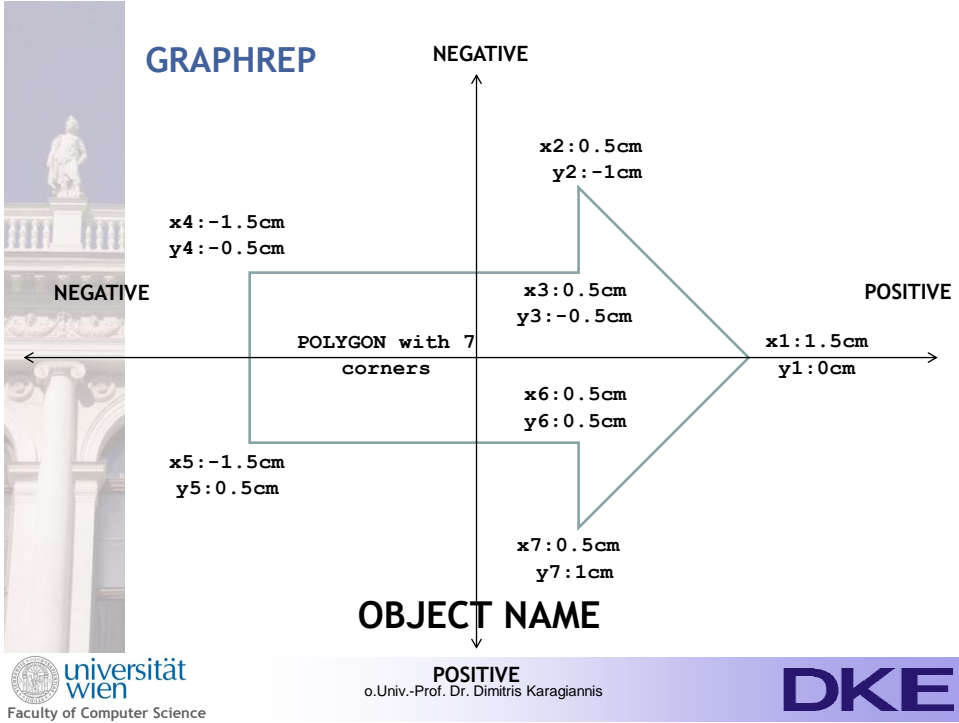
<i>all-file</i> :	[ <i>version</i> ] <i>libraries application-libraries</i>   [ <i>version</i> ] <i>newlib</i> .
<i>version</i> :	<b>VERSION</b> <i>identifier</i> .
<i>newlib</i> :	<i>applicationlib-definition</i> { <i>recordclass</i> } { <i>attrprof</i> } <i>bplib welib</i> .
<i>applicationlib-definition</i> :	<b>APPLICATION LIBRARY</b> <i>identifier</i> .
<i>recordclass</i> :	<b>RECORDCLASS</b> <i>identifier</i> ':' <i>identifier</i> { <i>attribute</i> } .
<i>attrprof</i> :	<b>ATTRIBUTEPROFILECLASS</b> <i>identifier</i> ':' <i>identifier</i> { <i>attribute</i> } .
<i>bplib</i> :	<b>BUSINESS PROCESS LIBRARY</b> <i>identifier</i> { <i>instanceattribute-setting</i> } <i>library</i> .
<i>welib</i> :	<b>WORKING ENVIRONMENT LIBRARY</b> <i>identifier</i> { <i>instanceattribute-setting</i> } <i>library</i> .
<i>libraries</i> :	{ <i>we-library</i>   <i>bp-library</i> } .
<i>we-library</i> :	<i>we-library-definition</i> { <i>instanceattribute-setting</i> } <i>library</i> .
<i>bp-library</i> :	<i>bp-library-definition</i> { <i>instanceattribute-setting</i> } <i>library</i> .
<i>we-library-definition</i> :	<b>WORKING ENVIRONMENT LIBRARY</b> <i>identifier</i> ':' <i>identifier</i> .
<i>bp-library-definition</i> :	<b>BUSINESS PROCESS LIBRARY</b> <i>identifier</i> ':' <i>identifier</i> .
<i>library</i> :	{ <i>class</i> } { <i>relationclass</i> } .
<i>class</i> :	<i>class-definition</i> { <i>attribute</i> }   <i>redefclass-definition</i> { <i>redefattribute</i> } .
<i>class-definition</i> :	<b>CLASS</b> <i>identifier</i> ':' <i>identifier</i> .
<i>redefclass-definition</i> :	<b>CLASS</b> <i>identifier</i> .
<i>relationclass</i> :	<i>relationclass-definition</i> { <i>instanceattribute</i> }   <i>redefrelationclass-definition</i> { <i>redefinstanceattribute</i> } .
<i>relationclass-definition</i> :	<b>RELATIONCLASS</b> <i>identifier</i> <b>FROM</b> <i>identifier</i> <b>TO</b> <i>identifier</i> .
<i>redefrelationclass-definition</i> :	<b>RELATIONCLASS</b> <i>identifier</i> .
<i>attribute</i> :	<i>attribute-definition</i> { <i>facet-definition</i> } .

## Excursion: Application Library Language (ALL) Definition 2/3

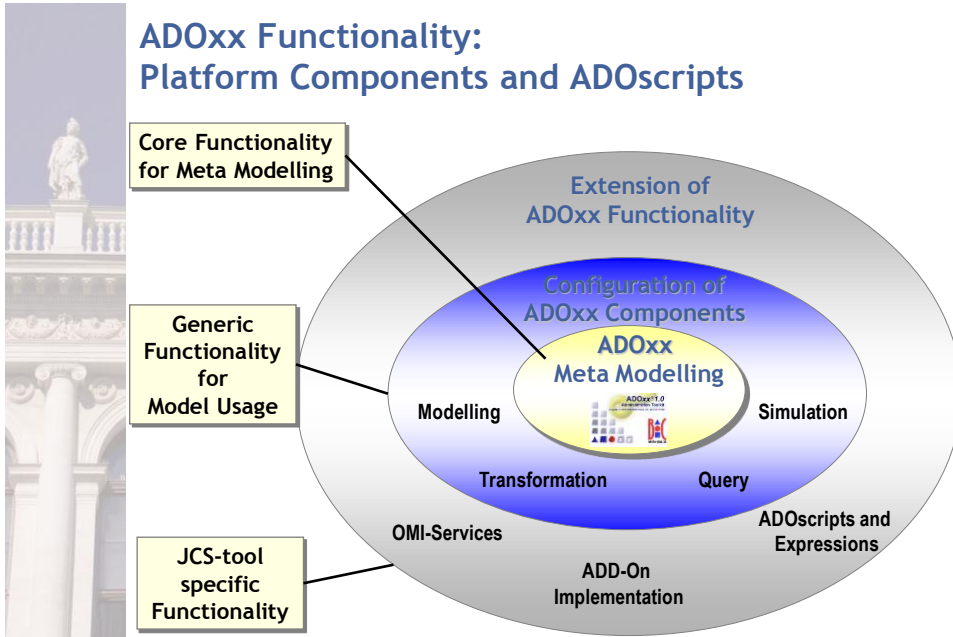
<i>redefattribute</i> :	<i>redefattribute-definition</i> { <i>facet-definition</i> } .
<i>instanceattribute</i> :	<i>instanceattribute-definition</i> { <i>facet-definition</i> } .
<i>redefininstanceattribute</i> :	<i>redefininstanceattribute-definition</i> { <i>facet-definition</i> } .
<i>redefattribute-definition</i> :	<i>redefininstanceattribute-definition</i>   <i>redefclassattribute-definition</i> .
<i>redefclassattribute-definition</i> :	<b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i> <b>VALUE</b> <i>val</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>VALUE</b> <i>val</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <b>RECORD</b>   <b>CLASSATTRIBUTE</b> <i>identifier</i> .
<i>redefininstanceattribute-definition</i> :	<b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i> <b>VALUE</b> <i>val</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>VALUE</b> <i>val</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <b>RECORD</b>   <b>ATTRIBUTE</b> <i>identifier</i> .
<i>attribute-definition</i> :	<i>instanceattribute-definition</i>   <i>classattribute-definition</i> .
<i>classattribute-definition</i> :	<b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i> <b>VALUE</b> <i>val</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>VALUE</b> <i>val</i>   <b>CLASSATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <b>RECORD</b> .
<i>instanceattribute-definition</i> :	<b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i> <b>VALUE</b> <i>val</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>VALUE</b> <i>val</i>   <b>ATTRIBUTE</b> <i>identifier</i> <b>TYPE</b> <b>RECORD</b> .
<i>instanceattribute-setting</i> :	<b>ATTRIBUTE</b> <i>identifier</i> <b>VALUE</b> <i>val</i> .
<i>facet-definition</i> :	<b>FACET</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i>   <b>FACET</b> <i>identifier</i> <b>TYPE</b> <i>typeidentifier</i> <b>VALUE</b> <i>val</i>   <b>FACET</b> <i>identifier</i> <b>VALUE</b> <i>val</i> .
<i>typeidentifier</i> :	<b>INTEGER</b>

## Application Library Language (ALL) Definition 3

<i>typeidentifier</i> :	<b>INTEGER</b>   <b>DOUBLE</b>   <b>STRING</b>   <b>DISTRIBUTION</b>   <b>TIME</b>   <b>ENUMERATION</b>   <b>ENUMERATIONLIST</b>   <b>PROGRAMCALL</b>   <b>INTERREF</b>   <b>EXPRESSION</b>   <b>ATTRPROFPREF</b> .
<i>application-libraries</i> :	{ <i>application-library-definition</i> <i>library-attachments</i> } .
<i>application-library-definition</i> :	<b>APPLICATION LIBRARY</b> <i>identifier</i> ' ' <i>identifier</i> .
<i>library-attachments</i> :	<b>WORKING ENVIRONMENT LIBRARY</b> <i>identifier</i> [ <b>BUSINESS PROCESS LIBRARY</b> <i>identifier</i> ]   <b>BUSINESS PROCESS LIBRARY</b> <i>identifier</i> [ <b>WORKING ENVIRONMENT LIBRARY</b> <i>identifier</i> ] .
<i>identifier</i> :	'<' <i>any_characters_but_newline</i> '>' .
<i>val</i> :	<i>number</i>   "' ' <i>any_characters</i> "' .



## ADOxx Functionality: Platform Components and ADOscripts

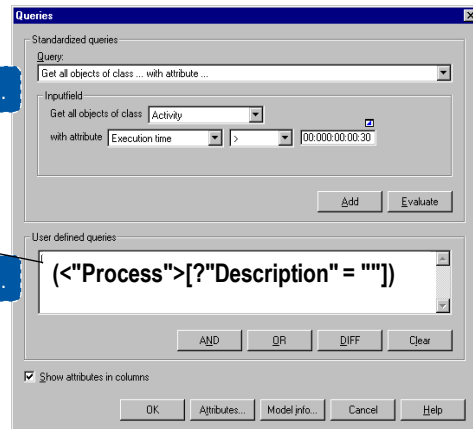


## ADOxx Platform Component Sample: Query with ADOxx Query Language (AQL)

Query Component can be switched included into the JCS-Tool. Default and User-defined queries are provided.  
Pre-defined queries for the JCS-Tool can be configured.

1. Default query
2. User-defined query

"List all Processes without a Description"

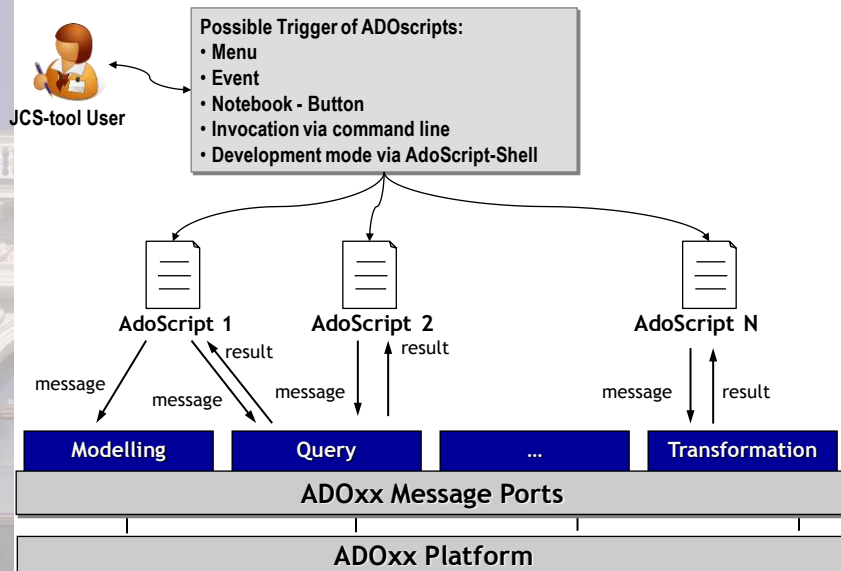


## ADOxx Platform Component Sample: Pre-defined Query Configuration

User_Defined_Queries :	{ Modeltype } .
Modeltype :	<b>MODELTYPE</b> "Modeltype-name" { MenuItem } .
MenuItem :	<b>MENUITEM</b> "MenuItem-name" { Query } .
Query :	<b>QUERY</b> "Query-name" Inputfields AqlExpressions Resultattributes .
Inputfields :	<b>FIELD</b> { Text  Edit   ClassList   AttributeList   AttributeValueList   AttributeEnumerationList   ValueList } .
Text :	<b>TEXT</b> "Text"
Edit :	<b>EDIT</b> length: numval value:"Default-value" [ dialog ] [ type:Type ] .
Type :	<b>double</b>   <b>integer</b>   <b>string</b>   <b>time</b>   <b>date</b>   <b>datetime</b> .
ClassList :	<b>CLASSLIST</b> .
AttributeList :	<b>ATTRLIST</b> "Classname" .
AttributeValueList :	<b>ATTRVALLIST</b> class:"Classname" attr:"Attributename" length: numval .
AttributeEnumerationList :	<b>ATTRENUMLIST</b> class:"Classname"   relclass:"Relationclassname" attr:"Attributename" .
ValueList :	<b>VALLIST</b> val1:"First value" val2:"Second value" val3:"Third value" ...
AqlExpressions :	<b>AQL</b> [ PLAN title:"Titlename" column: numval ] { Expression   Reference } .
Expression :	<b>EXPR</b> "Aql expression part" .
Reference :	<b>REF</b> numval .
Resultattributes :	<b>RESULT</b> { Class   Relationclass } .
Class :	<b>CLASS</b> "Classname" [ modeltype:"Modeltypename" ] { ATTR "Attributename" } .
Relationclass :	<b>RELCLASS</b> "Relationclassname" [ modeltype:"Modeltypename" ] { ATTR "Attributename" } .
numval :	any valid number .

MODELTYPE „Process Ma"  
MENUITEM „Dialctic Update"  
QUERY „All Process influenced by specific dialect"

## ADOxx ADOscripts using Message Port Paradigm



## ADOxx ADOscripts Keywords and Definition

### Command execution

EXECUTE                      SEND  
CC            SYSTEM  
START        CALL

### Set Operations

SET        SETG  
SETL

### Controll elements

IF        ELSIF  
ELSE     WHILE  
FOR      BREAK  
NEXT     EXIT

### Definition of Procedures/Functions

PROCEDURE                      FUNCTION

### LEO (Return Format) Handling

LEO        LEO parse

StatementSeq :	{ Statement } .
Statement :	Execute   Send   CC   System   Start   Call   Set   SetL   SetG   Leo   IfStatement   WhileStatement   ForStatement   BreakStatement   ExitStatement   FunctionDefinition   ProcedureDefinition   ProcedureCall .
Execute :	ExecuteFile   ExecuteEx .
ExecuteFile :	EXECUTE file:scriptText [ scope:ScopeSpec ] .
ExecuteEx :	EXECUTE scriptText [ scope:ScopeSpec ] .
ScopeSpec :	separate   same   child
Send :	SEND msg:Text to:msg:portName [ answer:varName ] .
CC :	CC msg:portName [ debug ] [ raw ] anyLeoElement .
System :	SYSTEM str:Expr [ with-console-window ] [ hide ] [ result:varName ] .
Start :	START str:Expr [ cmdshow:CmdShow ] .
CmdShow :	showmaximized   showminimized   showinactive   shownormal .
Call :	CALL dll:str:Expr function:str:Expr { InputParam } [ result:varName ] [ free:mem:str:Value ] .
InputParam :	varName:anyExpr .
Set :	SET { VarAssignment } .
SetL :	SETL { VarAssignment } .
SetG :	SETG { VarAssignment } .
VarAssignment :	varName:anyExpr .
Leo :	LEO [ parseCmd ] { accessCmd } .
parseCmd :	parse:string:Expr .
accessCmd :	get-elem-count:varName   set-cur-elem-index:int:Expr   get-keyword:varName   is-contained:varName [ str:Expr ]   get-int-value:varName [ str:Expr ]   get-real-value:varName [ str:Expr ]   get-time-value:varName [ str:Expr ]   get-tmm-value:varName [ str:Expr ]   get-modifier:varName:str:Expr .
IfStatement :	IF boolean:Expr { StatementSequence } { ELSIF boolean:Expr { StatementSequence } } { ELSE { StatementSequence } } .
WhileStatement :	WHILE boolean:Expr { StatementSequence } .
ForStatement :	ForNumStatement   ForTokenStatement .
ForNumStatement :	FOR varName from:num:Expr to:num:Expr [ by:num:Expr ]

## Using OMI-Services or ADD-Ons

### File Exchange

- export of models as XML file out of ADOxx
- individual manipulation within individual tool
- import of result xml file into ADOxx

### ADOxx API

- Build ADOxx Script
- Embed the script into programming language of choice
- Execute ADOxx Script via the ADOxx Web-Service method „execute“





## OMI Development Environment:

- 1) Create OMI Account and apply for Project
- 2) Receive ADOxx Development Environment
- 3) Configure Project Page on OMI
- 4) Participate ADOxx Training or receive consultancy
- 5) Participate OMI Community