

Experiences from the Implementation of a Structured- Entity-Relationship Modeling Method in a Student Project

Thilo Maximilian Glässner, Florian Heumann,
Luca Keßler, Felix Härer, Andreas Steffan,
Hans-Georg Fill

Experiences from the Implementation of SERM

1. Course Design
2. Project Phases
3. Experiences and Recommendations

1 Course Design

Motivation

- ❖ Conceptualization of modeling methods
- ❖ Illustration of a domain-specific method
- ❖ Hands-on experience with meta modeling tools

Course Objectives

- ❖ Conceptualization of an IT-based modeling method using ADOxx
- ❖ Implementation of an up-to-date modeling tool for the SERM approach
- ❖ Functional requirements: design data model, derive executable SQL code

<https://www.uni-bamberg.de/seda/studium/lehrveranstaltungen-im-ss-2017/masterseminar-metamodellierung/>

1 Course Design

Meta-modeling project seminar

- ❖ Seminar in curriculum of IS Master of Science
- ❖ Carried out as a project
- ❖ Small team of master students
- ❖ Integrated theory and hands-on sessions → instantaneous feedback

Phases

1. Preliminary – tools and modeling language
2. Foundations – modeling method conceptualization
3. Formalization – SERM in FDMM
4. Design and Implementation – SERM on the ADOxx platform

2.1 Preliminary – Tools and Modeling Language

Preliminary → Foundations → Formalization → Design & Implementation

Tools

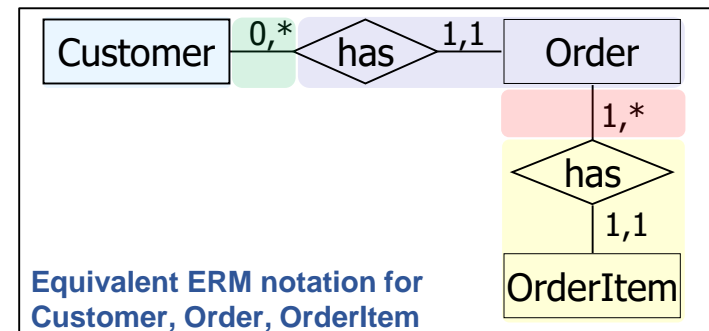
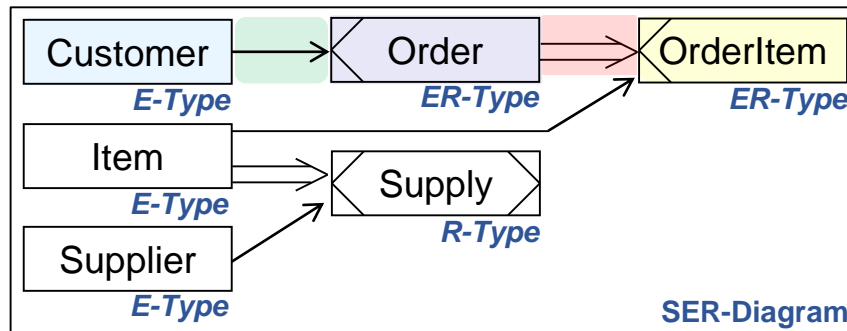
- ❖ Introduction to software tools for meta modeling and programming
- ❖ Meta modeling with ADOxx Client-Server 1.5
- ❖ GraphRep tools for creating notation
- ❖ Notepad++ text editor with ADOscript syntax highlighting

2.1 Preliminary – Tools and Modeling Language

Preliminary → Foundations → Formalization → Design & Implementation

Modeling Language

- ❖ Domain: Data Modeling
- ❖ Language: *Structured Entity-Relationship Model (SERM)* by Sinz¹ (1992)
- ❖ Re-cap of language known from Bachelor-level courses
- ❖ Models contain E-, R-Type (like ERM) and ER-Type with edges for cardinalities

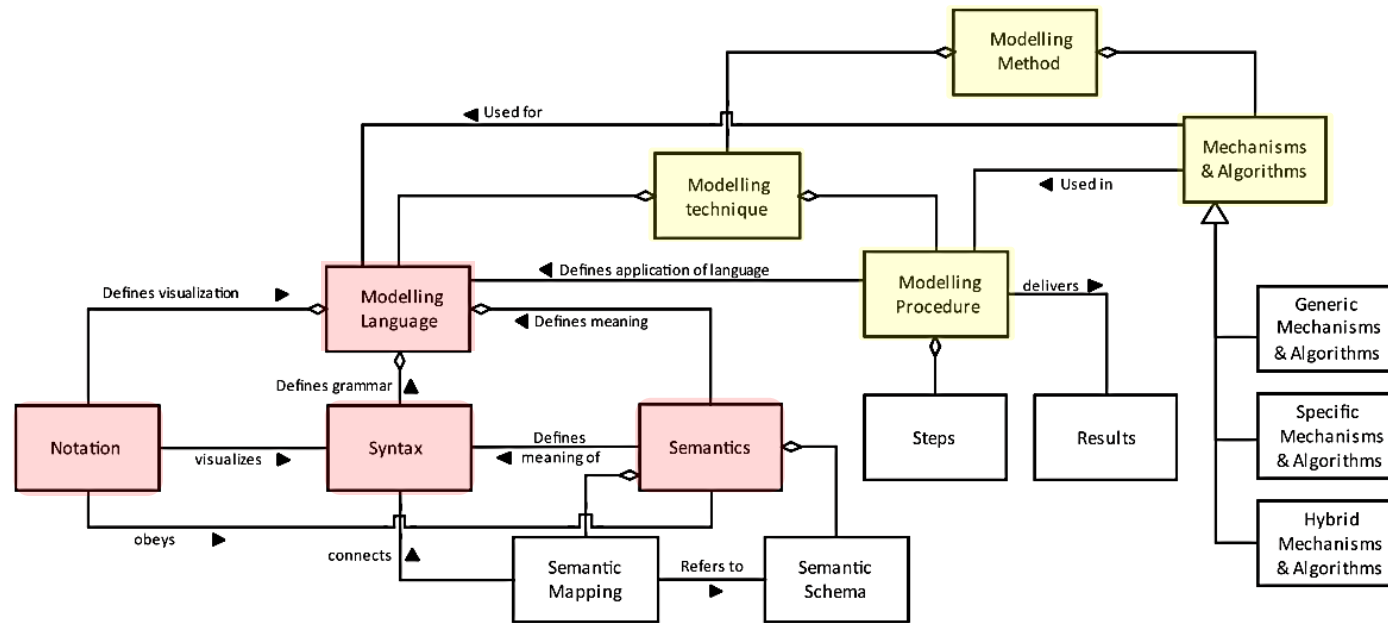


Dependencies of E-/R-/ER-Types

2.2 Foundations – Method Conceptualization

Preliminary → Foundations → Formalization → Design & Implementation

- ❖ Conceptualization of modeling method according to Karagiannis and Kühn (2002), Fill and Karagiannis (2013)

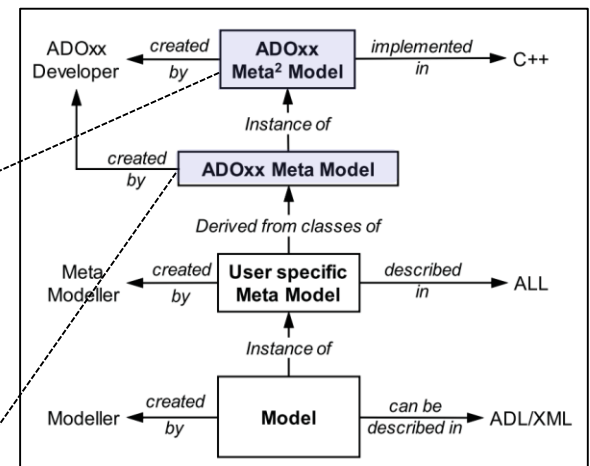
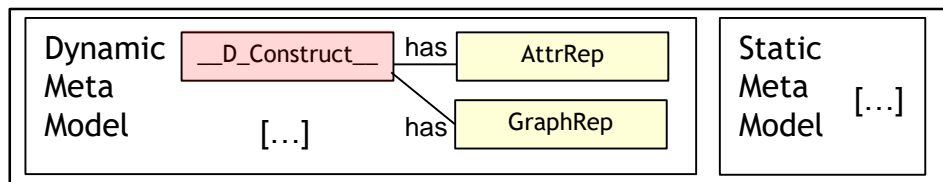
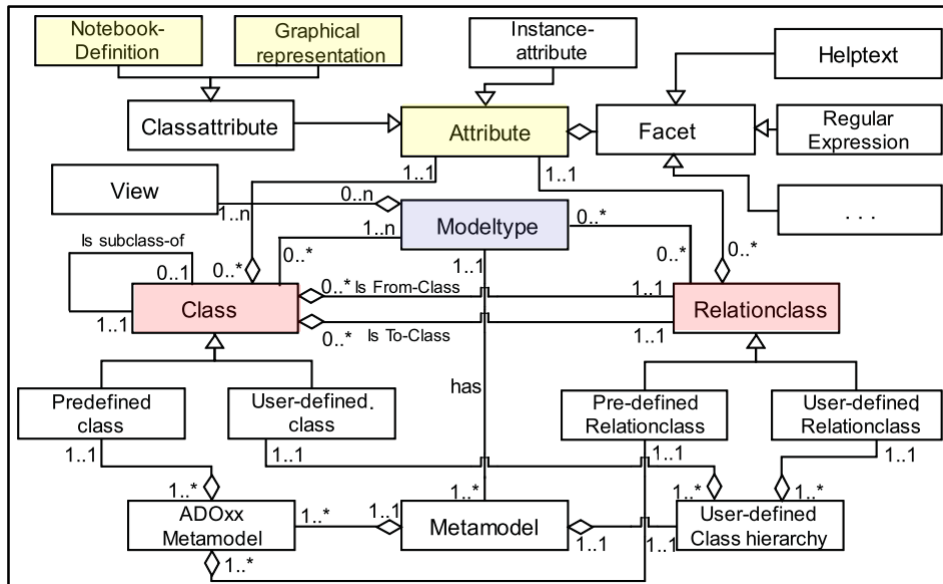


Components of modelling methods (Karagiannis and Kühn 2002)

2.2 Foundations – Method Conceptualization

Preliminary → Foundations → Formalization → Design & Implementation

- ❖ Conceptualization of modeling method according to Karagiannis and Kühn (2002), Fill and Karagiannis (2013)



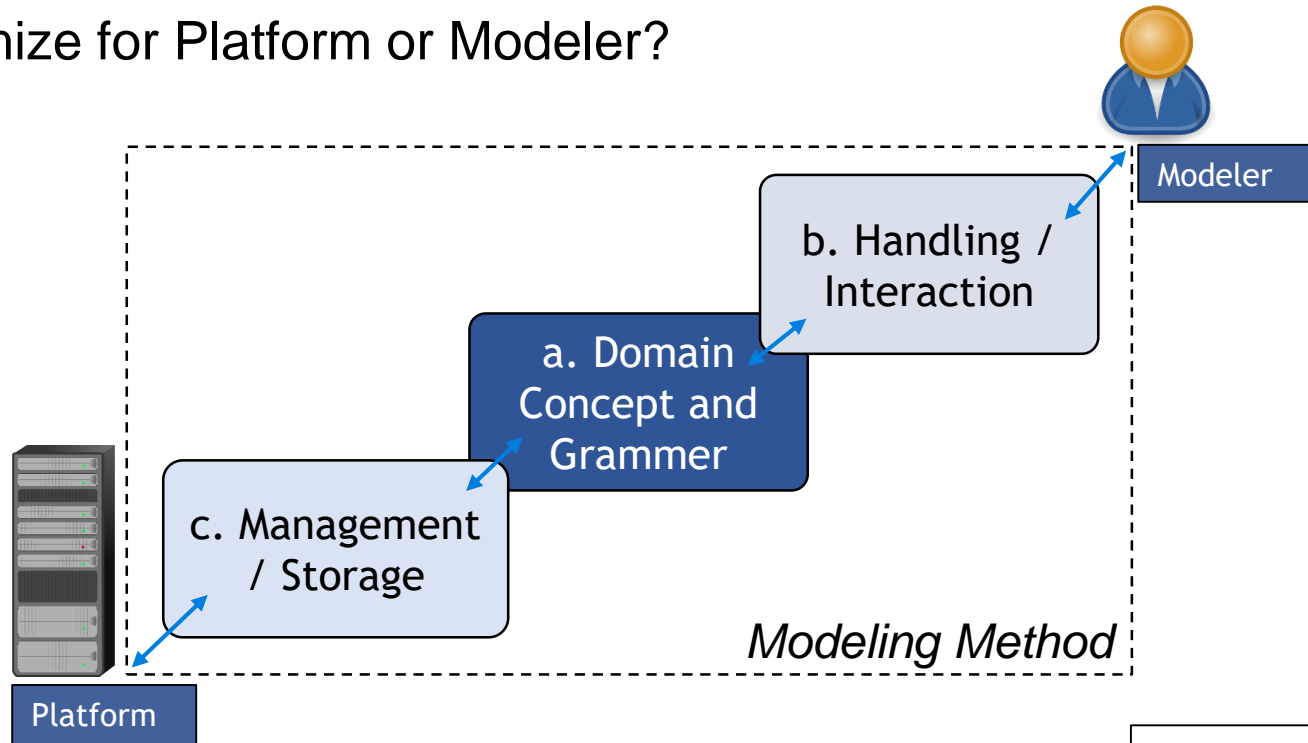
Roles and Languages in ADOxx

Fill and Karagiannis (2013): On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform

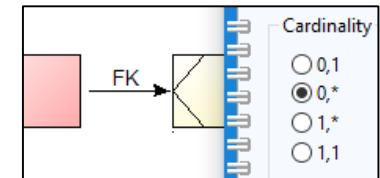
2.2 Foundations – Method Conceptualization

Preliminary → Foundations → Formalization → Design & Implementation

❖ Optimize for Platform or Modeler?



- ❖ Conflict between b. and c., depends on instance
- ❖ E.g. represent each of the 4 SERM edges as 4 separate Relation Class or one Relation Class with attributes?

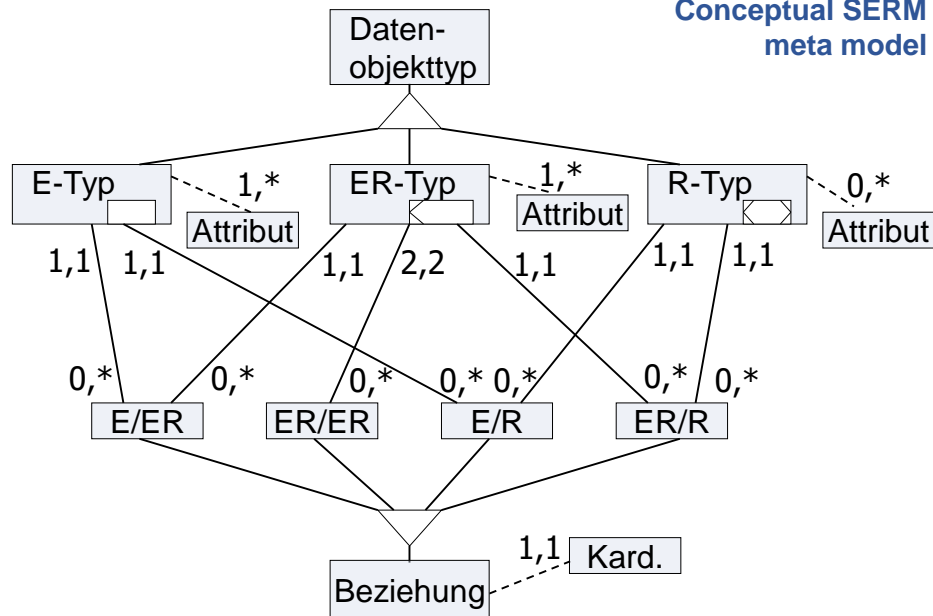


2.3 Formalization – SERM in FDMM

Preliminary → Foundations → **Formalization** → Design & Implementation

- ❖ Introductory lecture (45 minutes) by example of FDMM publication (Fill 2013)
- ❖ No FDMM experience, student draft, revision
- ❖ Technology-independent, conscious design choices

Conceptual SERM
meta model



Model Type

FDMM

$$MT_{SERM} = \langle O_{SERM}^T, D_{SERM}^T, A_{SERM} \rangle$$

Object Type

$$O_{SERM}^T = \{ DOT, EType, ERType, RType, DOTRelation, Triangle, DotToTriangle, TriangleToER, RecordAttributes, RecordInheritedAttributes \}$$

Type DOT (data object type) as basis

$$EType \preceq DOT$$

$$ERType \preceq DOT$$

$$RType \preceq DOT$$

$$\text{domain}(\text{relatesFrom}) = \{ DOTRelation \}$$

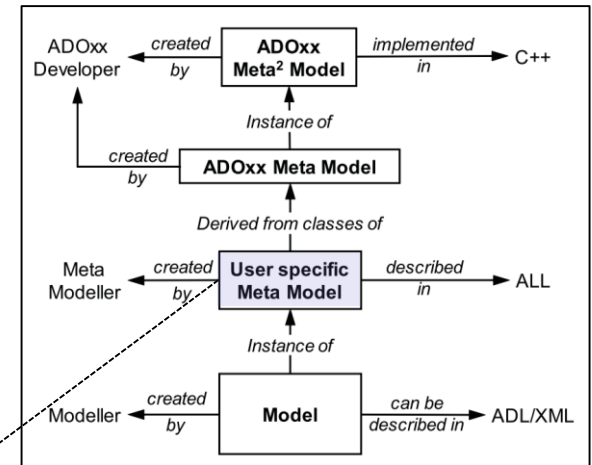
$$\text{range}(\text{relatesFrom}) = \{ DOT \}$$

$$\text{card}(DOTRelation, \text{relatesFrom}) = \langle 1, 1 \rangle \quad [\dots]$$

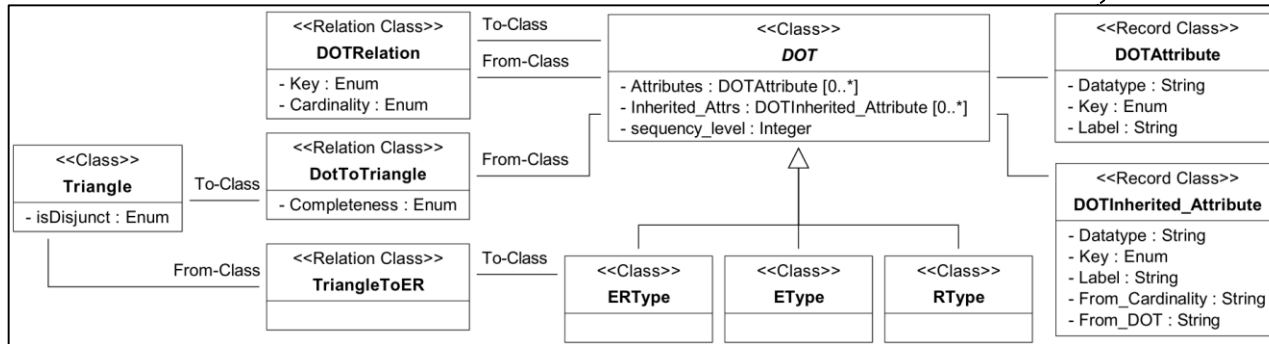
2.4 Implementation – SERM on ADOxx

Preliminary → Foundations → Formalization → Design & Implementation

- ❖ Extension of ADOxx meta model for abstract syntax
- ❖ Class DOT („data object type“) with sub-types of SERM model elements EType, EType, RType
- ❖ Design choice for SERM generalization concept as multiple sub-classes of Relation Class
- ❖ Attributes Key, Datatype for SQL derivation



Roles and Languages in ADOxx

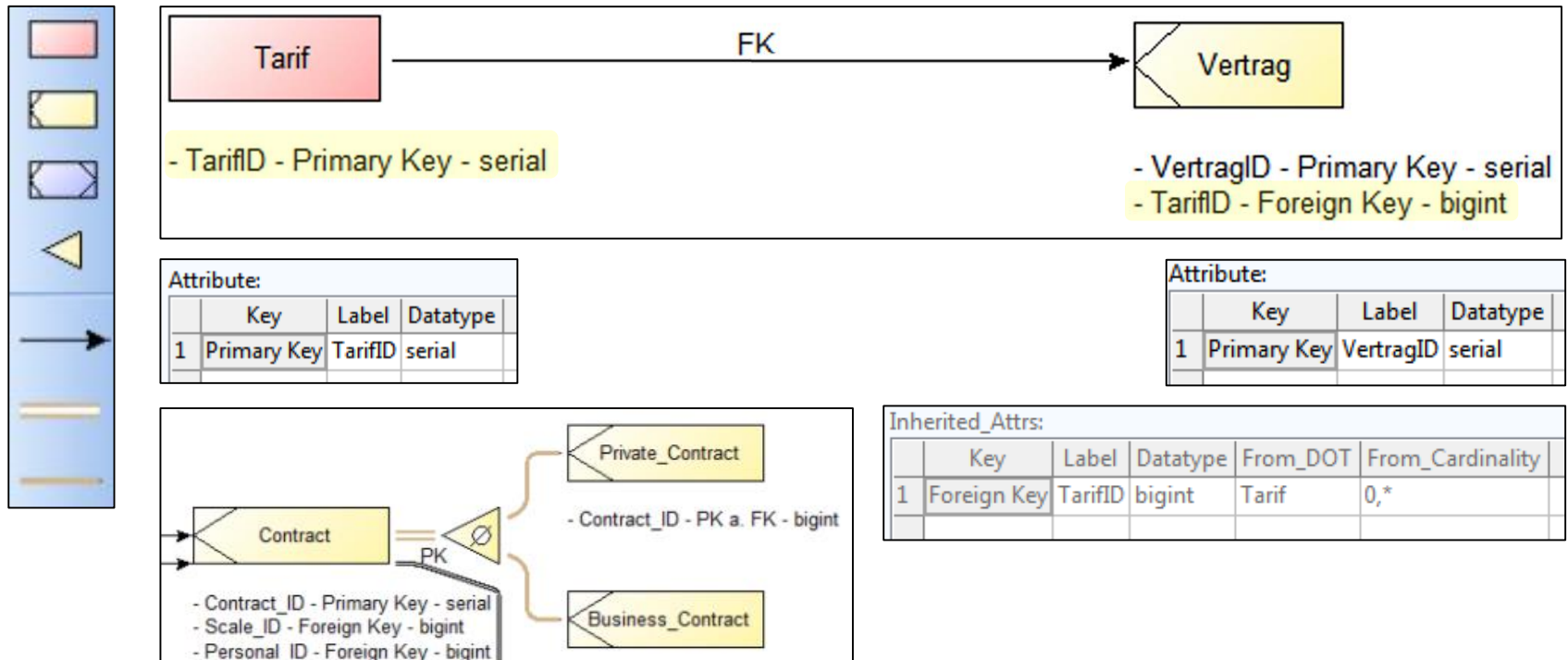


Fill and Karagiannis (2013): On the Conceptualization of Modelling Methods Using the ADOxx Meta Modelling Platform

2.4 Implementation – SERM on ADOxx

Preliminary → Foundations → Formalization → Design & Implementation

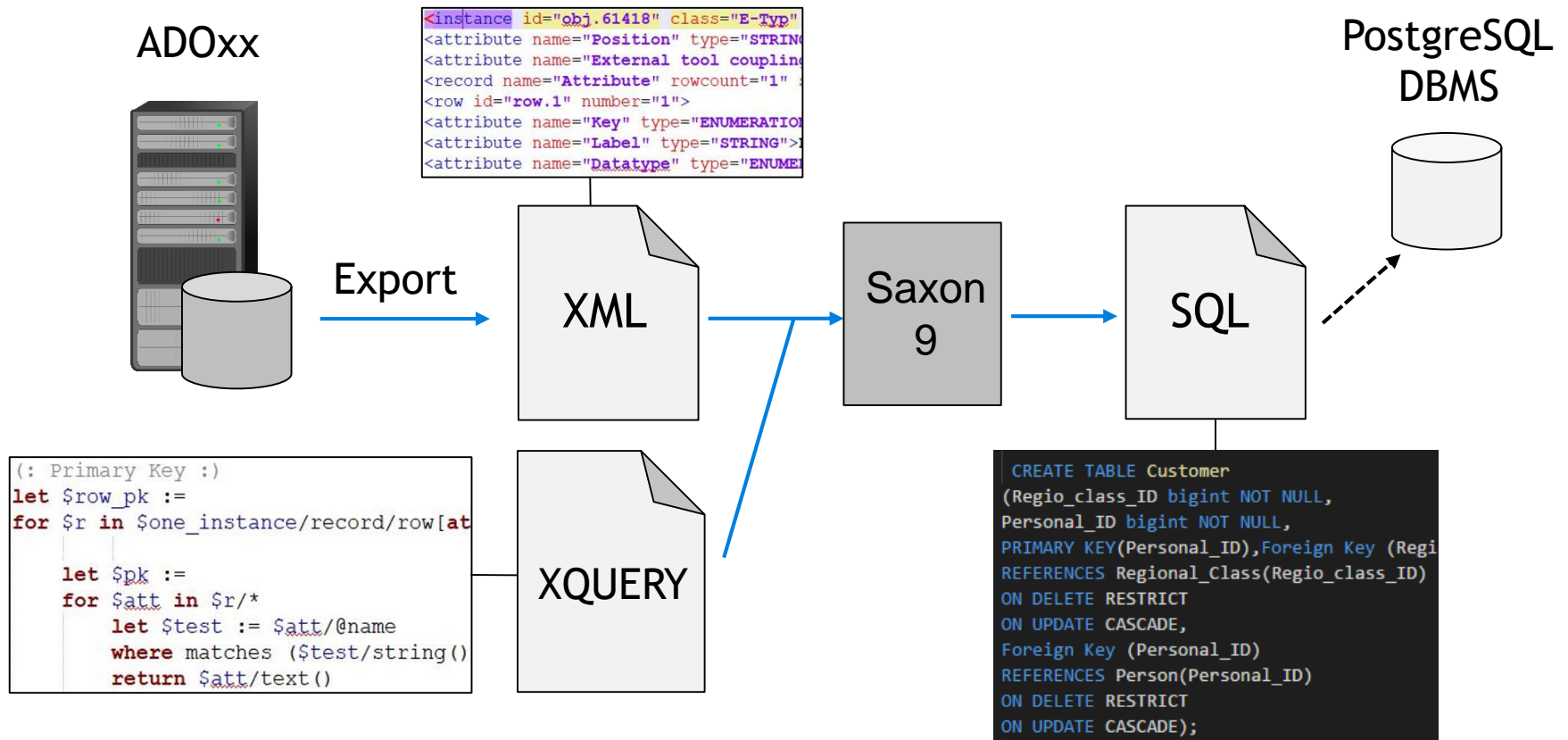
- ❖ Implementation of concrete syntax
- ❖ According GraphRep definitions for model elements, generalization



2.4 Implementation – SERM on ADOxx

Preliminary → Foundations → Formalization → Design & Implementation

❖ Transformation to SQL – Architecture

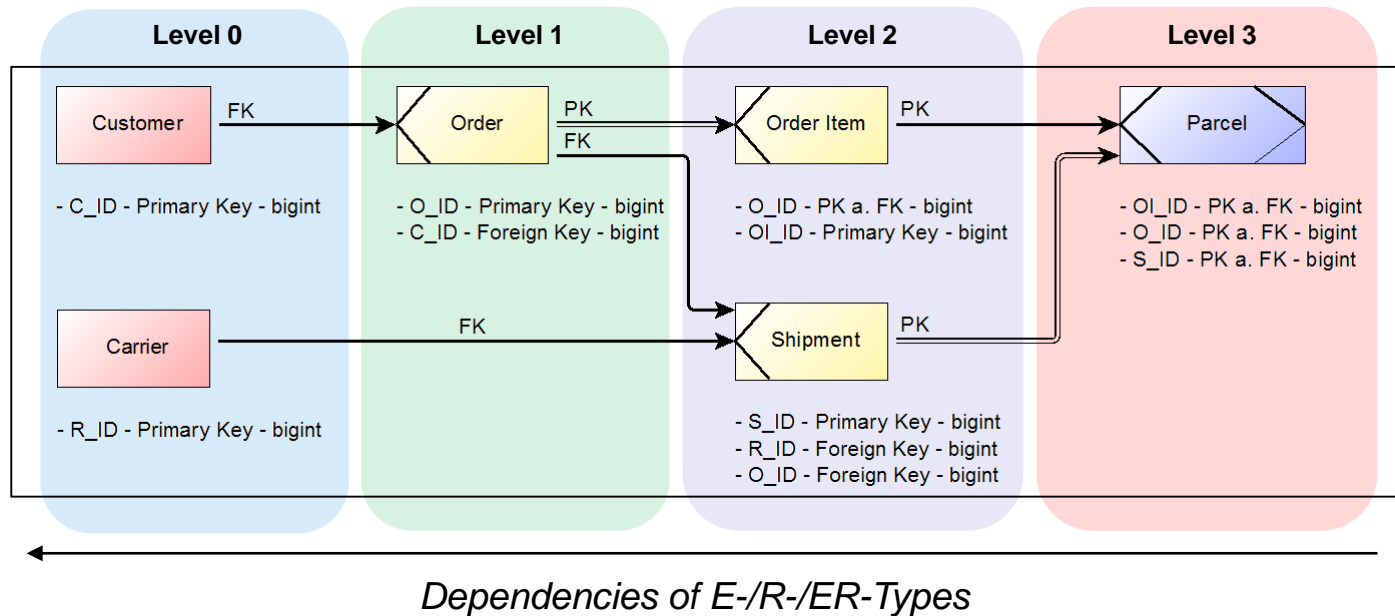


2.4 Implementation – SERM on ADOxx

Preliminary → Foundations → Formalization → Design & Implementation

❖ Transformation to SQL – Technologies and Algorithms

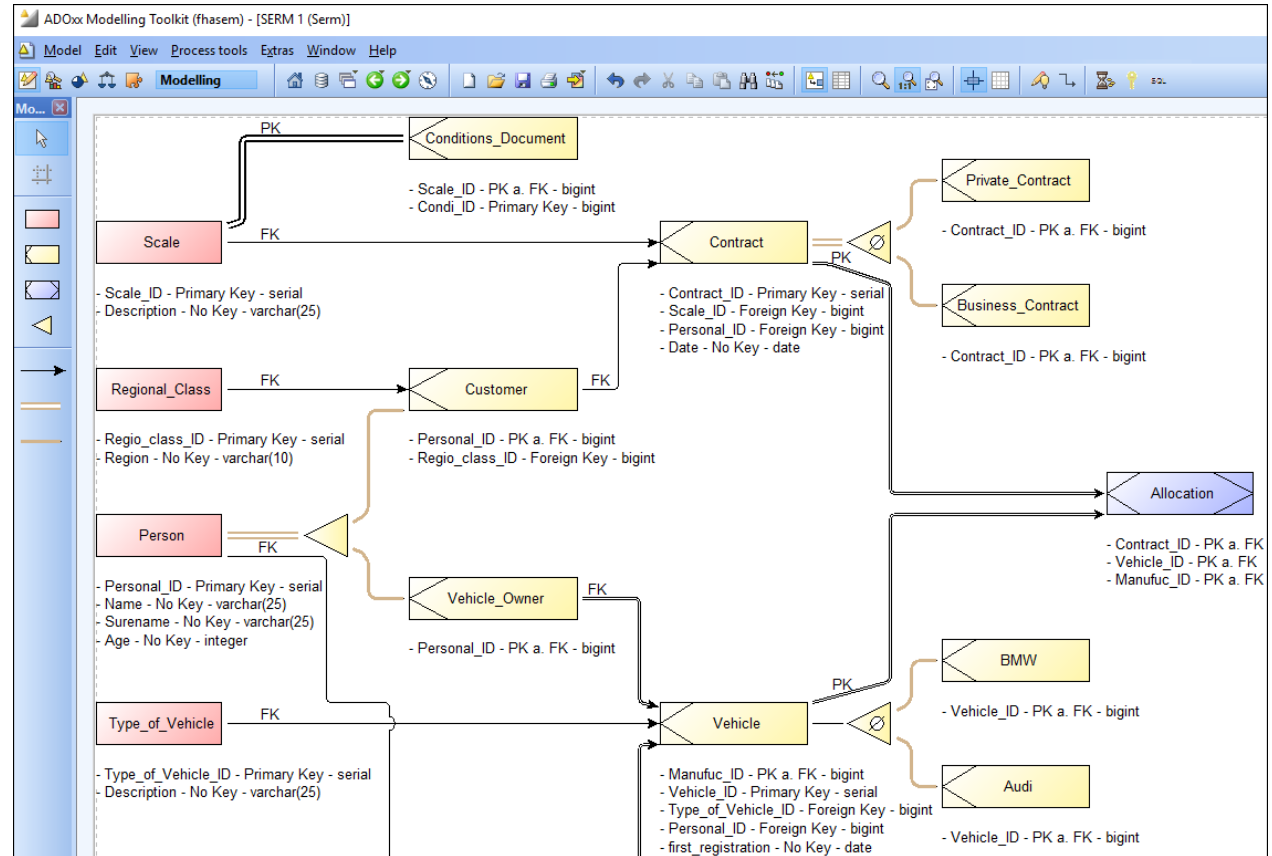
1. Cardinalities for data types → Constraints
2. Dependency level calculation → ADOscript
3. Attribute and key inheritance → Attributes and LEO Expressions
4. SQL-Code generation by level → XML and XQUERY



2.4 Implementation – SERM on ADOxx

Preliminary → Foundations → Formalization → Design & Implementation

- ❖ Finished product
- ❖ Sample model for evaluation



3 Experiences and Recommendations

Experiences of Students and Teachers

- ❖ Experience evaluation from students and teachers
- ❖ Students had prior knowledge of SERM approach and SQL
- ❖ No prior experience with meta modeling or ADOxx
- ❖ Learning results
 - Technologically:
XML, XQuery, ADOxx, ADOscript
 - Methodologically:
Meta modeling, tool implementation, enterprise modeling
- ❖ For teaching, a project-type course works best
- ❖ Frequent interactions and hands-on sessions, e.g. alternating with theory
- ❖ Challenge: provide tools and resources on all abstraction levels

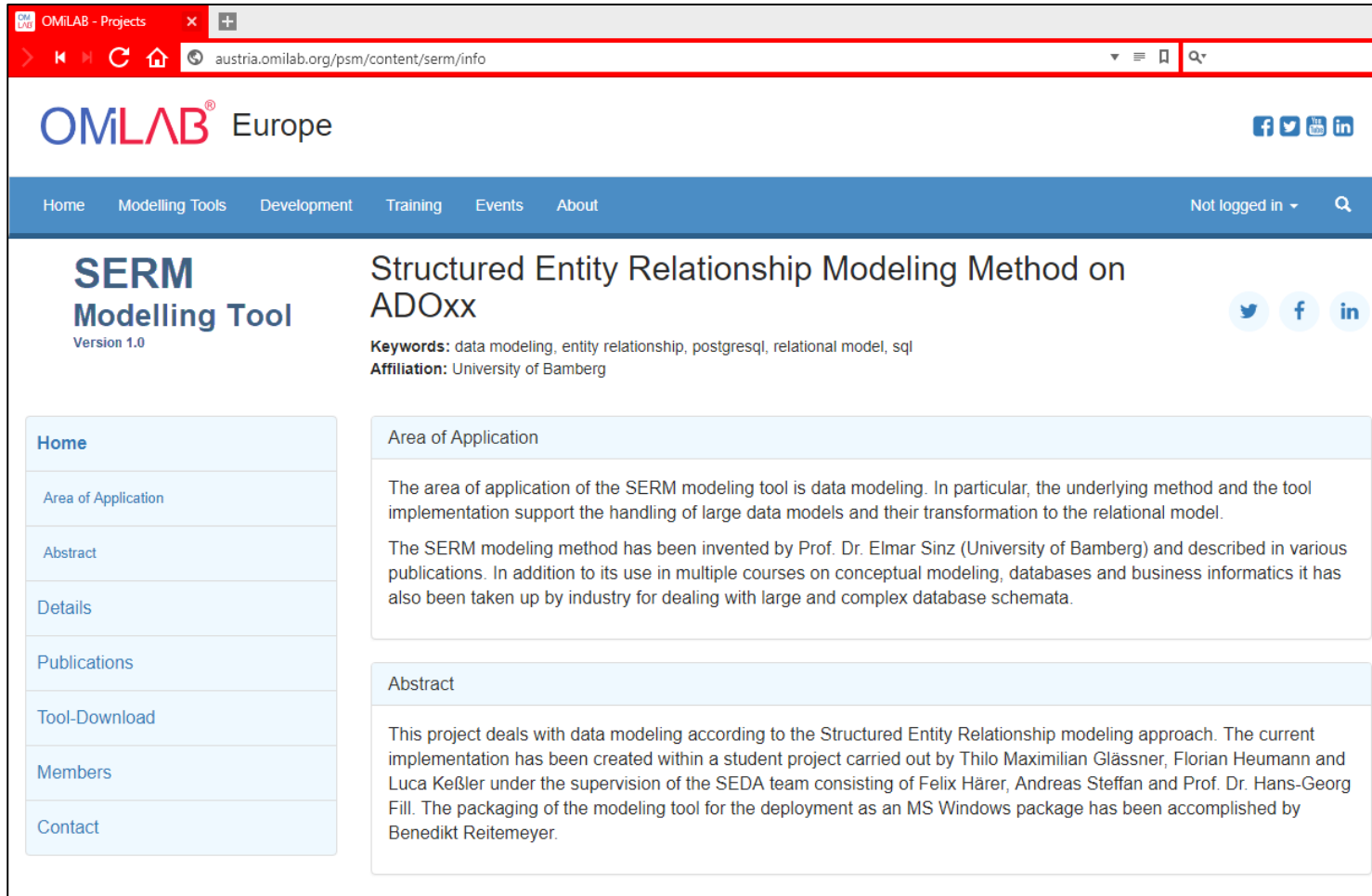
3 Experiences and Recommendations

Recommendations for Future Projects

- ❖ Definition of project goals with regard to modeling language
- ❖ Existing meta models and other material with consideration of sufficiency
- ❖ Conscious design decisions, for implementation or re-implementation
 - Inter-linkages in domain, processing, handling
- ❖ Illustration of examples and best practices
- ❖ Enable collaboration of participants by providing resources and tools

OMiLAB – SERM on ADOxx

Openly accessible at <http://austria.omilab.org/psm/content/serm>



The screenshot shows a web browser window displaying the SERM on ADOxx page. The browser's address bar shows the URL austria.omilab.org/psm/content/serm/info. The website header features the OMiLAB Europe logo and navigation links for Home, Modelling Tools, Development, Training, Events, and About. The main content area is titled "SERM Modelling Tool" (Version 1.0) and "Structured Entity Relationship Modeling Method on ADOxx". It includes a list of keywords (data modeling, entity relationship, postgresql, relational model, sql) and an affiliation (University of Bamberg). A sidebar on the left contains a menu with links to Home, Area of Application, Abstract, Details, Publications, Tool-Download, Members, and Contact. The main content area is divided into sections for "Area of Application" and "Abstract".

SERM Modelling Tool
Version 1.0

Structured Entity Relationship Modeling Method on ADOxx

Keywords: data modeling, entity relationship, postgresql, relational model, sql
Affiliation: University of Bamberg

Area of Application

The area of application of the SERM modeling tool is data modeling. In particular, the underlying method and the tool implementation support the handling of large data models and their transformation to the relational model.

The SERM modeling method has been invented by Prof. Dr. Elmar Sinz (University of Bamberg) and described in various publications. In addition to its use in multiple courses on conceptual modeling, databases and business informatics it has also been taken up by industry for dealing with large and complex database schemata.

Abstract

This project deals with data modeling according to the Structured Entity Relationship modeling approach. The current implementation has been created within a student project carried out by Thilo Maximilian Glässner, Florian Heumann and Luca Keßler under the supervision of the SEDA team consisting of Felix Härer, Andreas Steffan and Prof. Dr. Hans-Georg Fill. The packaging of the modeling tool for the deployment as an MS Windows package has been accomplished by Benedikt Reitemeyer.