UML-based Forward Database Engineering

Zvjezdan Spasic, Aleksandar Vukotic, Drazen Brdjanin, Danijela Banjac, Goran Banjac

M-lab Research Group @ Faculty of Electrical Engineering

University of Banja Luka, Bosnia & Herzegovina

Presentation Outline

- Research context and motivation
- From CDM to RDM
- From RDM to DB
- Implementation within the AMADEOS tool
- Conclusion and future work

Research Context & Motivation

Model-driven Software Engineering Laboratory

Faculty of Electrical Engineering • University of Banja Luka

http://m-lab.etf.unibl.org

M-lab long-term research project:

Online model-driven tool for automated DB design using standardized modeling notations

Main M-lab achievements:

AMADEOS

http://m-lab.etf.unibl.org:8080/amadeos

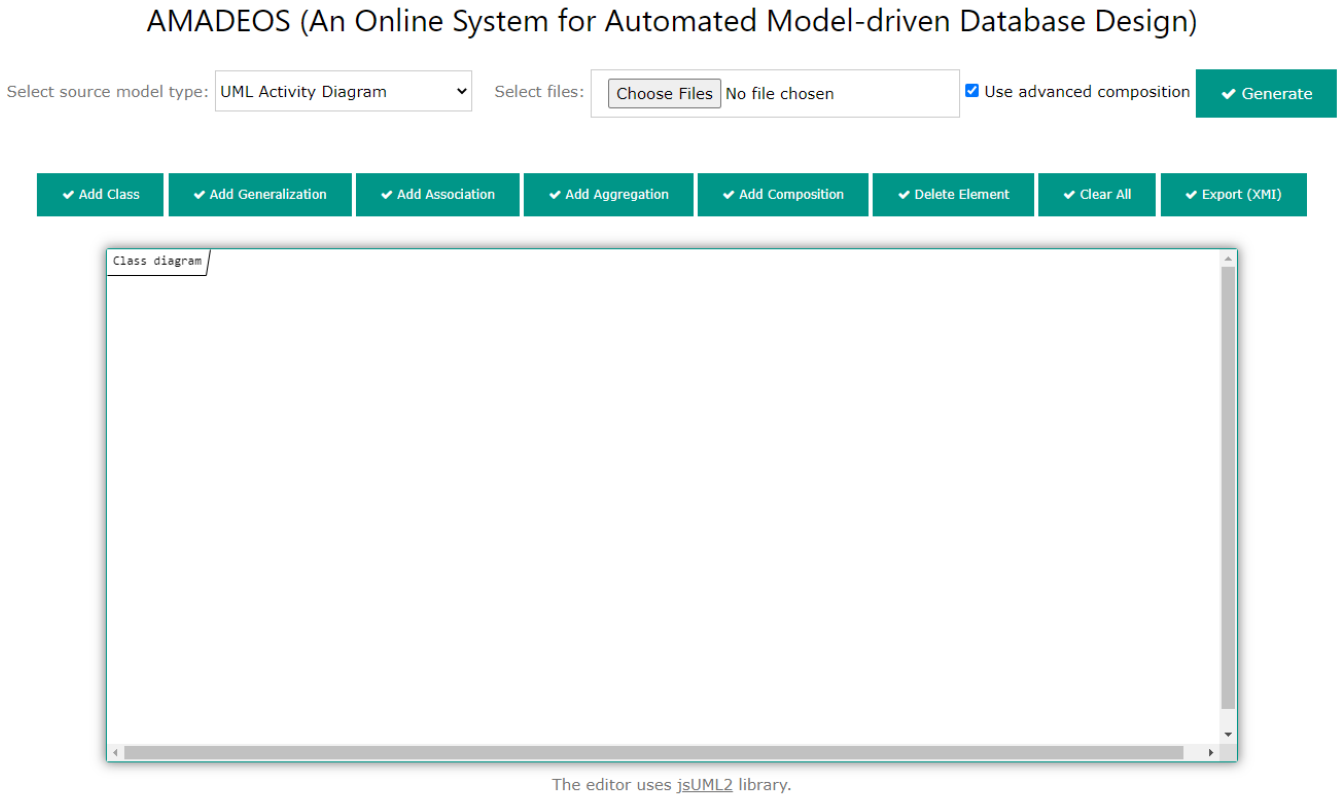
– The first online web-based tool for automatic CDM derivation from collections of differently represented and differently serialized BPMs

REDBUL

http://m-lab.etf.unibl.org:8081

– The first online web-based system for reverse database engineering which represents the extracted DB schema by standard UML class diagram

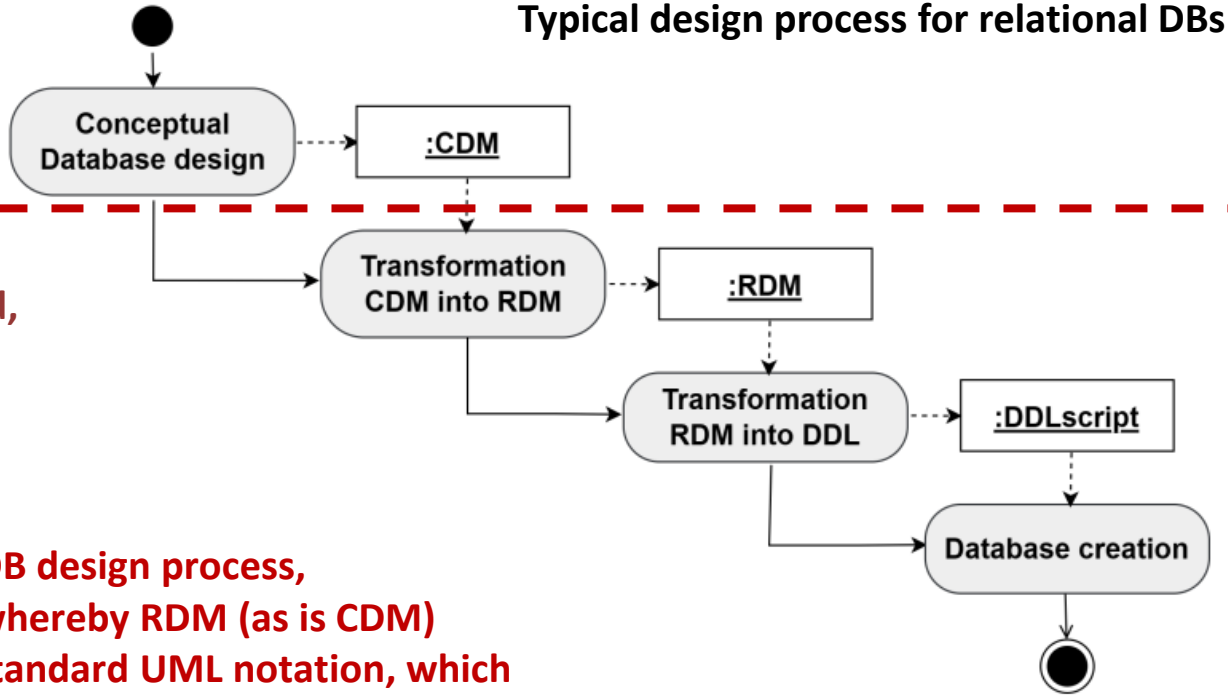
Research Context & Motivation



Research Context & Motivation

In AMADEOS, initial CDM can be derived automatically from a collection of BPMs

Typical design process for relational DBs



However, other phases of the DB design process were not supported, which means that a user was able only to derive a CDM

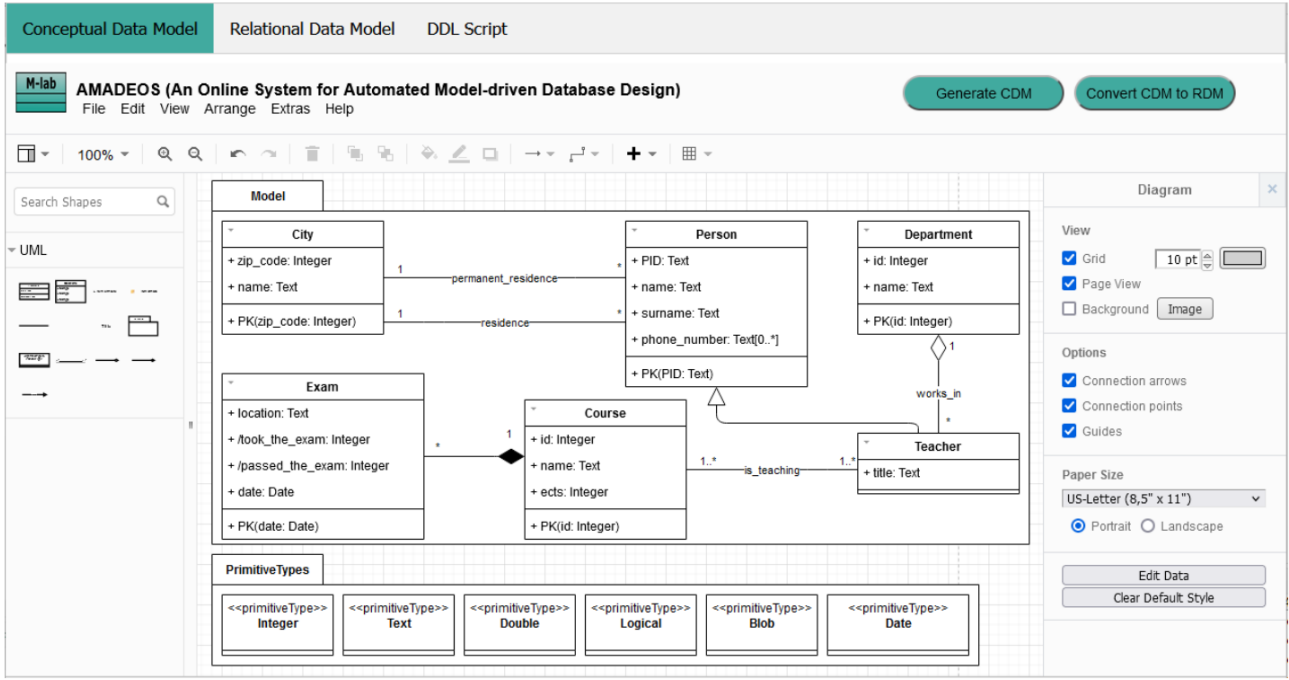
Research objectives

- **Complete the coverage of the DB design process, from CDM to the physical DB, whereby RDM (as is CDM) should be represented by the standard UML notation, which will eliminate the portability issues**

From CDM to RDB

Support for CDM design

- **Automatically generate** an initial CDM (from a collection of BPMs) which can be further improved
- **Import** an existing CDM from a file
- **Create** a new CDM from scratch



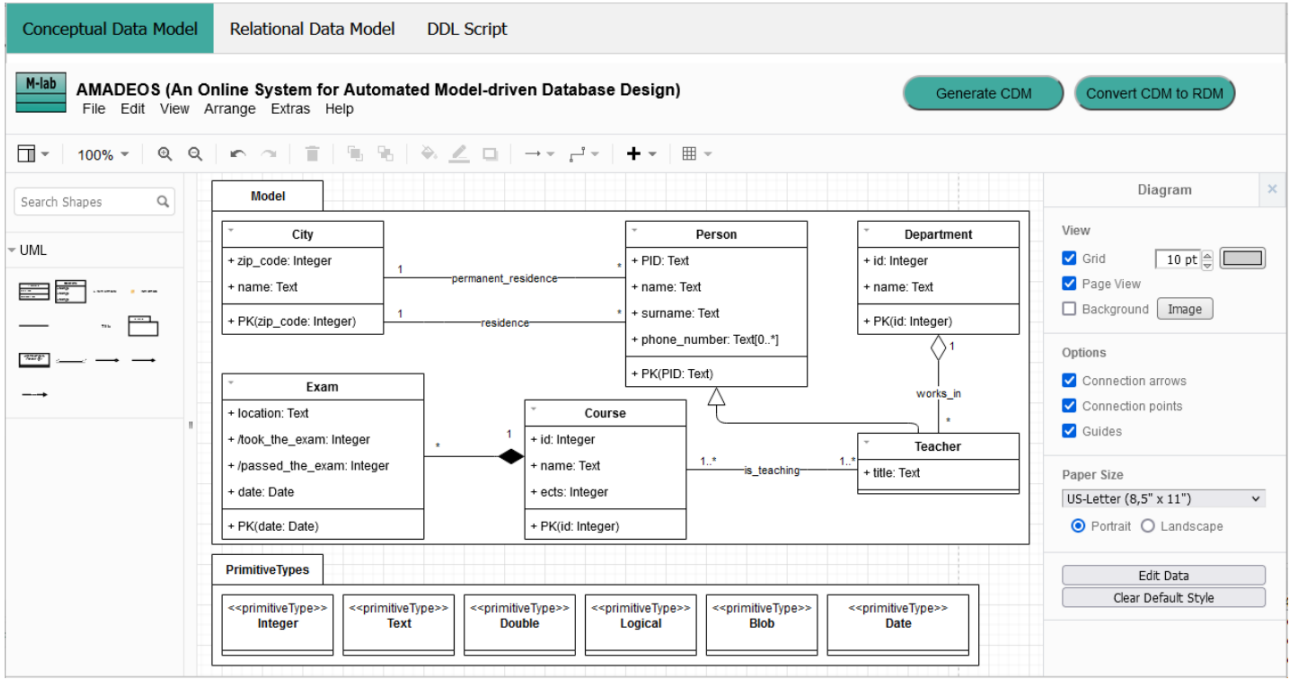
AMADEOS has been significantly improved in UI and UX

(implementation of the client side is based on JavaScript and mxGraph library)

From CDM to RDB

CDM representation

- AMADEOS represents CDM by the **standard UML class diagram**
- All **typical CDM concepts** are supported: strong entity types, relationship types, aggregations, compositions, generalizations/specializations, multi-valued attributes
- Each **PK** is represented by the same-named operation in the class that corresponds to the given strong entity type, whereby the operation parameters correspond to the PK attributes, by the name, order, and type

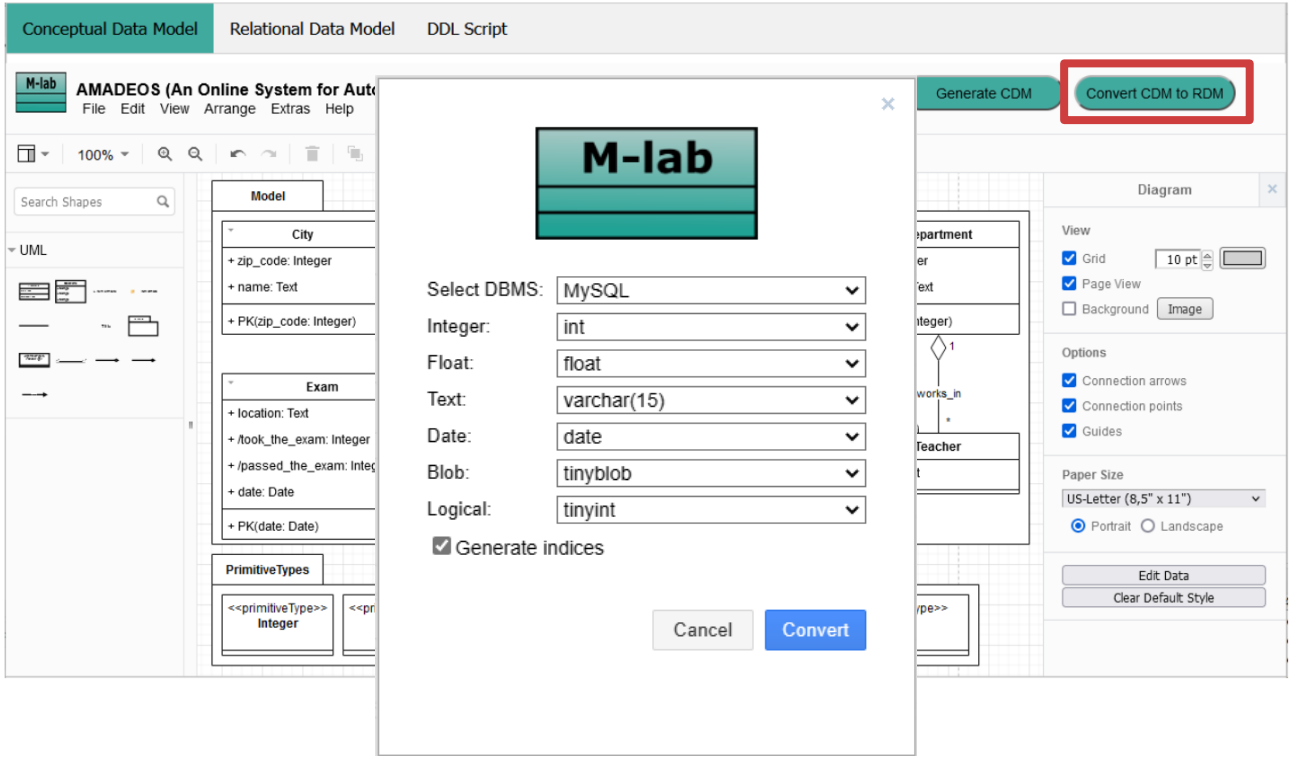


AMADEOS has been significantly improved in UI and UX
(implementation of the client side is based on JavaScript and mxGraph library)

From CDM to RDB

CDM → RDB

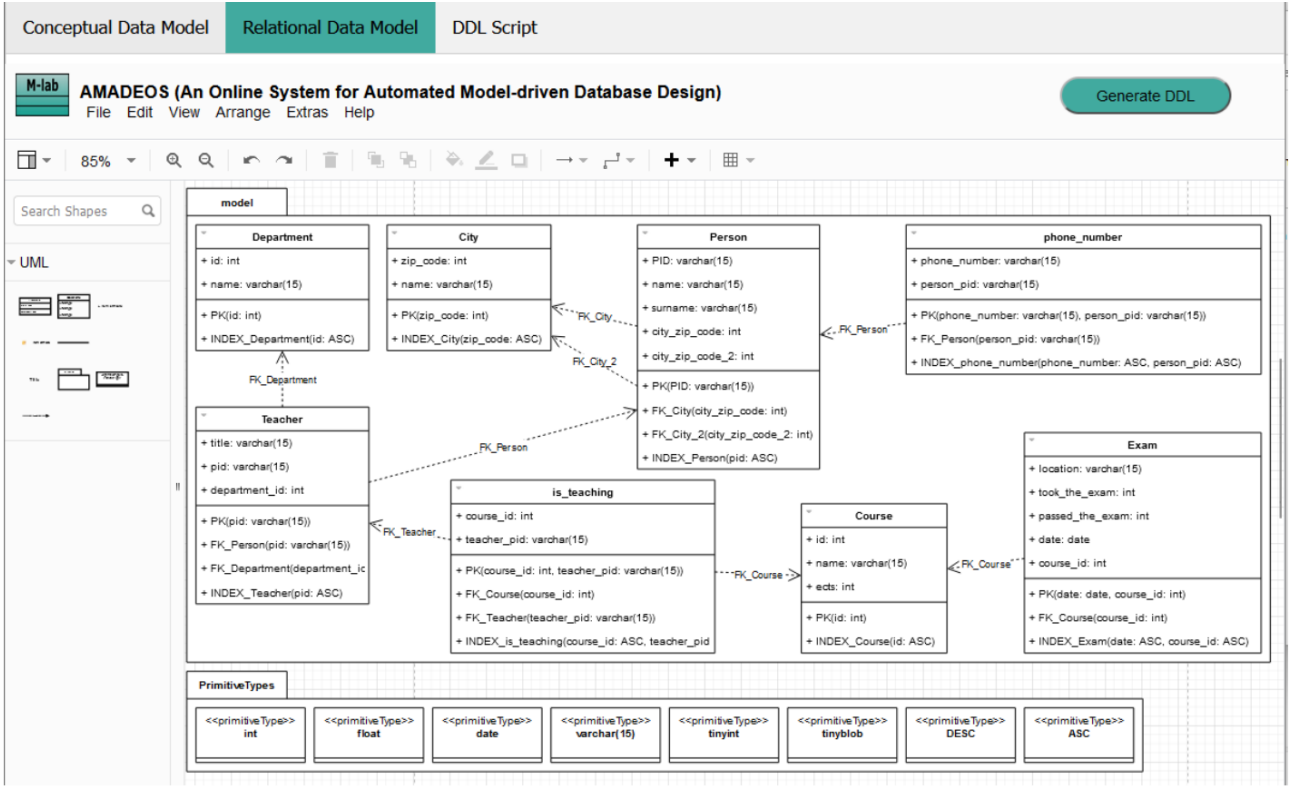
- Once a CDM is designed, a user proceeds to the next step of the forward engineering process – **the transformation of the CDM into the corresponding RDM**
- Supported **DBMS**s: MySQL, PostgreSQL, Microsoft SQL Server, Oracle, and IBM DB2
- After the configuration (datatype mapping and indices), AMADEOS **automatically generates the corresponding RDM**
- The transformation process is driven by a **set of rules**



From CDM to RDB

UML representation of RDM

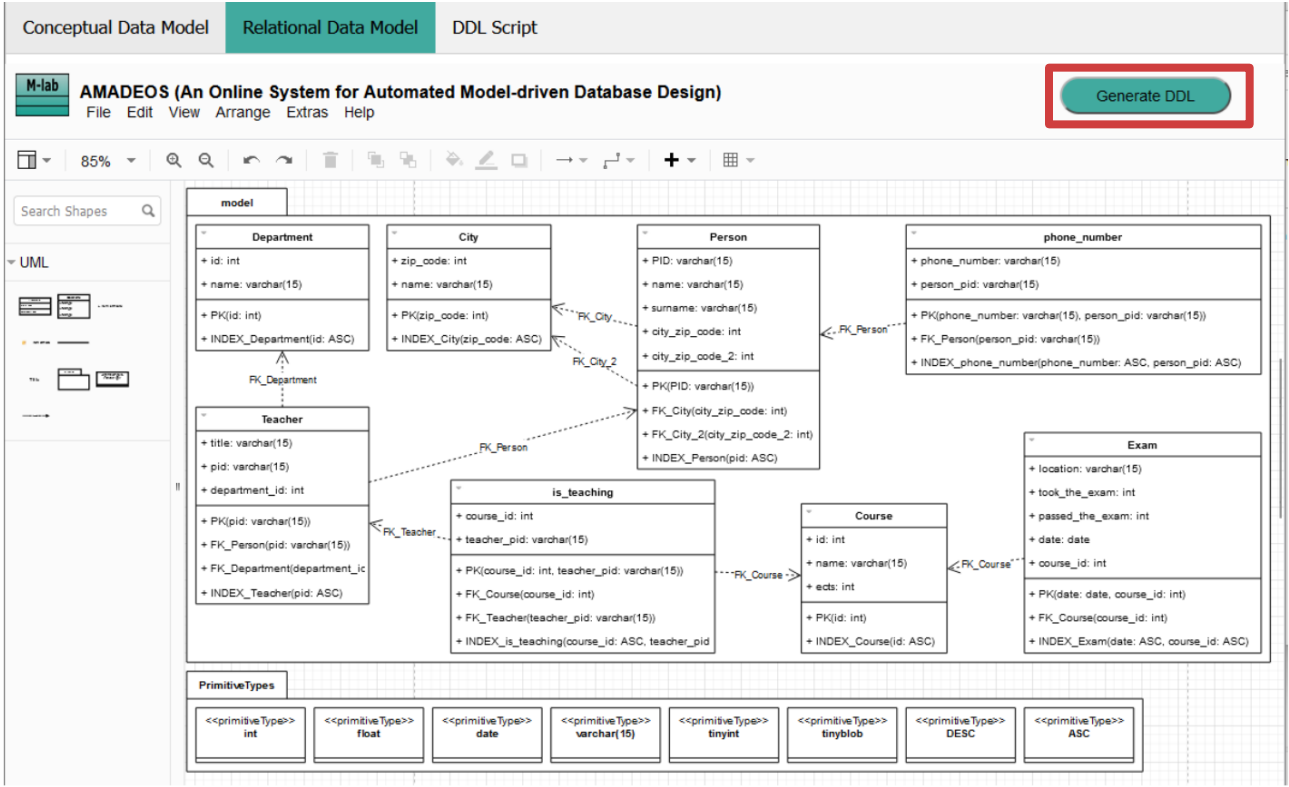
- AMADEOS represents RDM by the **standard UML class diagram**
- **RS** is represented by the corresponding same-named class, whereby each schema attribute is represented by the same-named class property
- **PK** of an RS is represented by the PK operation
- **FK** is also represented by the appropriate operation + the dependency between the mutually related RSs
- Each table **index** is also represented by an operation



From CDM to RDB

RDM → DDL script

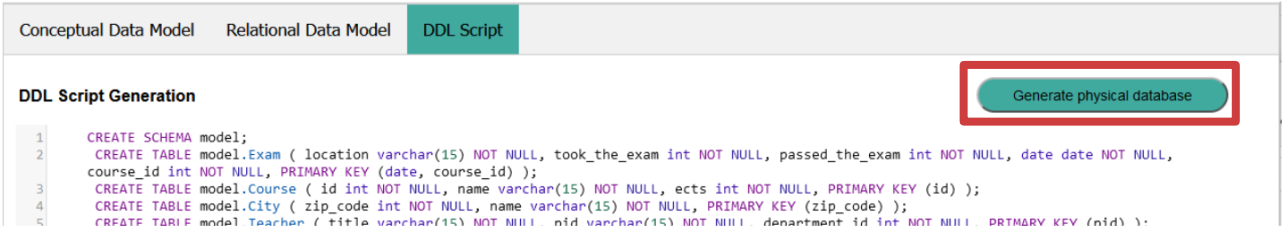
- Once an RDM is designed, a user proceeds to the next step of the forward engineering process – **the transformation of the RDM into the corresponding DDL script**
- AMADEOS provides a mechanism for transforming RDM to DDL script, where the mechanism itself is implemented using **Acceleo**
- The process is **fully automatic**

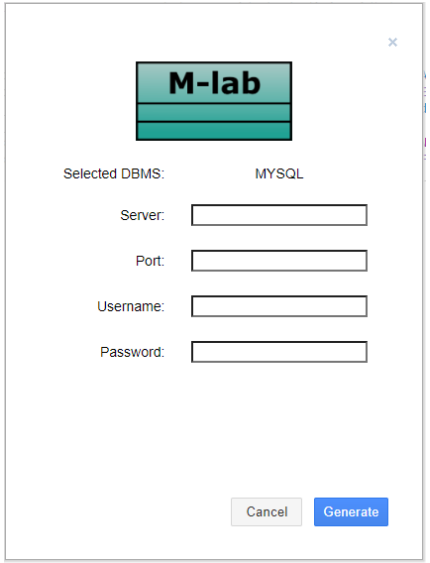


From RDB to DB

Generation of DB schema

- The **generated DDL code** is shown in the SQL editor of the DDL manipulation screen
- At this point, a user can perform **fine-tuning** of the DDL script before actual execution in the target DBMS
- Once a DDL script is prepared, it should be executed in the target DBMS
- Before executing the script, a connection to the target DBMS must be established (user must provide proper connection parameters)

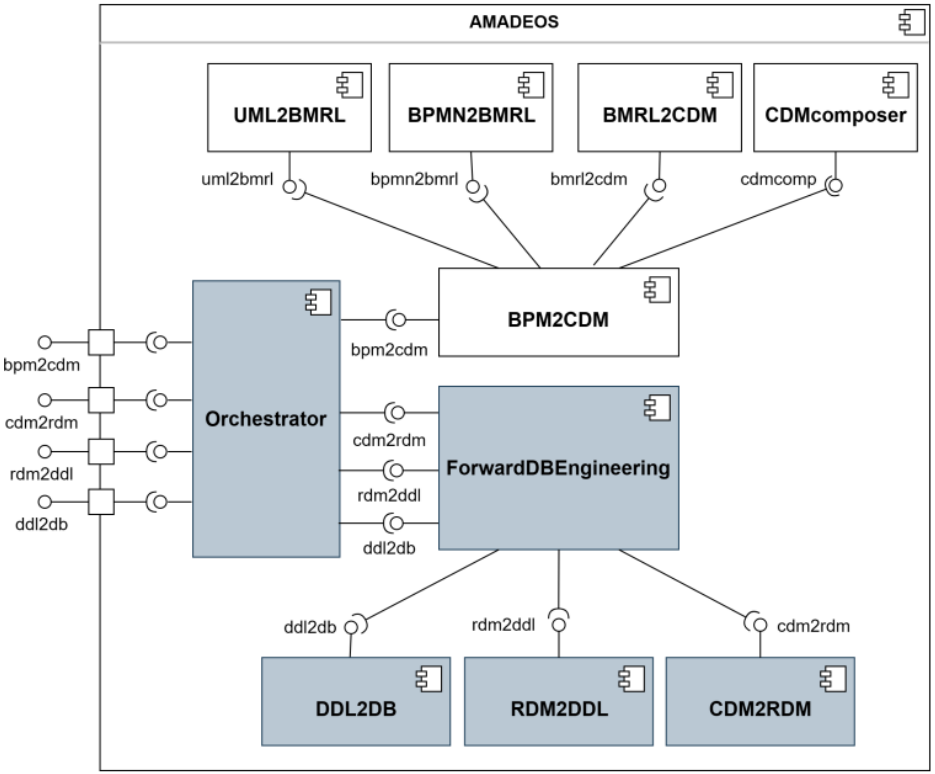




Implementation within the AMADEOS tool

Architectural changes

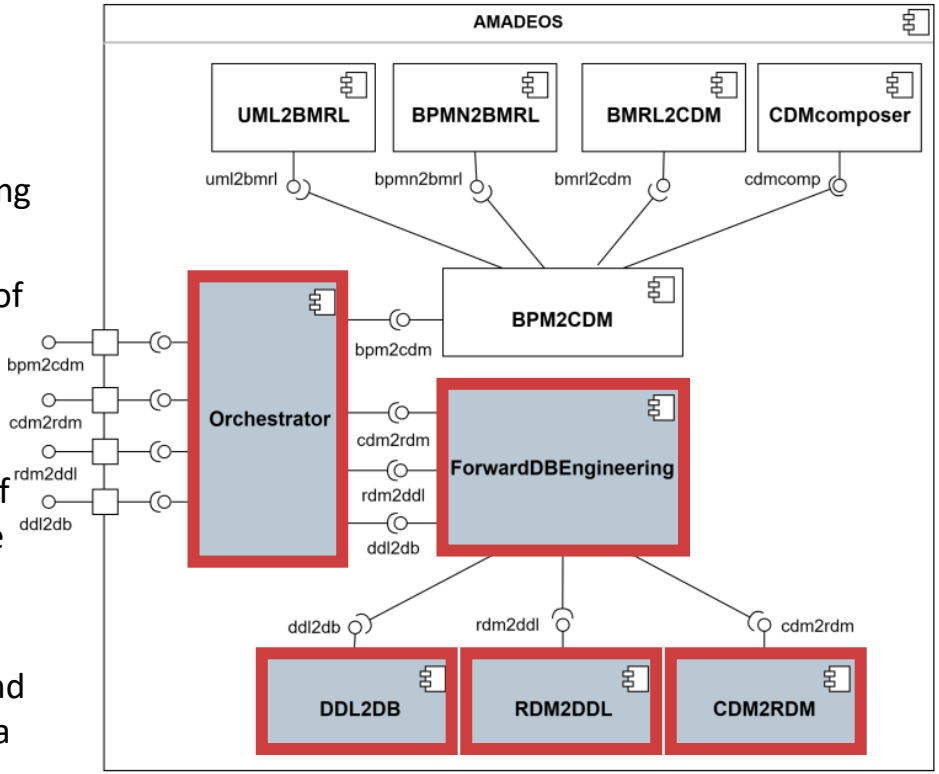
- Besides the **pre-existing automated BPM-driven CDM synthesis**, AMADEOS now enables all subsequent stages, including the **generation of the platform-specific RDM** based on CDM, **automated generation of the DDL script**, and, at the end of the workflow, **connection to the target DBMS and DDL script execution** that results with the corresponding physical DB schema
- All these new features and improvements caused architectural changes
- The pre-existing non-changed AMADEOS components are depicted as **white rectangles** and they implement the automatic CDM derivation from a collection of BPMs
- The new or changed components are depicted as **gray rectangles**



Implementation within the AMADEOS tool

Server-side architecture

- The **ForwardDBEngineering** service unites all design phases – it receives requests from the **Orchestrator** service and delegates these requests to the corresponding service that implements each phase
- The **CDM2RDM** service implements the transformation of CDM into RDM – it receives a CDM and configuration parameters for the transformation and returns the generated RDM
- The **RDM2DDL** service implements the transformation of RDM into DDL code – it receives an RDM and returns the generated DDL script
- The **DDL2DB** service implements the physical DB generation phase – it receives connection parameters and DDL script, establishes a DBMS connection, and creates a physical DB



Conslusion and future work

- In this paper, we presented AMADEOS with the new functionalities making the AMADEOS a **complete tool for BPM-driven DB design**
- On top of pre-existing automated BPM-driven CDM synthesis, AMADEOS now enables:
  - Generation of the **platform-specific RDM based on CDM**
  - Automated generation of the **DDL script**
  - **Connection** to the target DBMS and **DDL script execution** that results with the corresponding physical DB schema
- Thus, AMADEOS became the **first online BPM-driven tool for DB design**, as well as the **first online tool for DB design using the standard UML notation**
- Our future work will include:
  - Further improvement of the UI and UX
  - Adding still unsupported CDM concepts (e.g. *n-ary* associations)
  - Adding still unsupported RDM concepts (e.g. triggers, views)
  - Improvement of automated CDM derivation from BPMs

Thank you!

Zvjezdan Spasic, Aleksandar Vukotic, Drazen Brdjanin,
Danijela Banjac, Goran Banjac
M-lab Research Group @ Faculty of Electrical Engineering
University of Banja Luka, Bosnia & Herzegovina