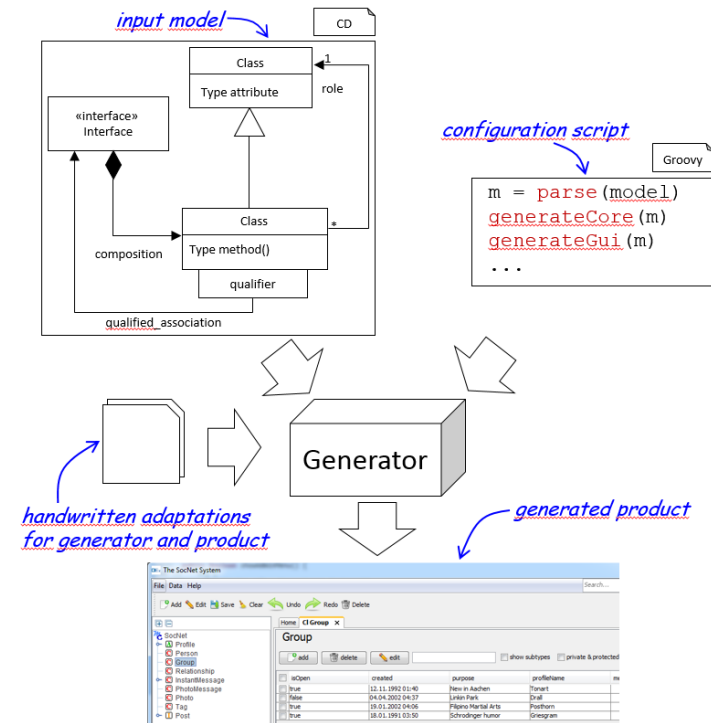


Mixed Generative and Handcoded Development of Adaptable data-centric Business Applications

27th October 2015

Pedram Mir Seyed Nazari, Alexander Roth,
and Bernhard Rumpe
Software Engineering
RWTH Aachen

<http://www.se-rwth.de/>



Motivation

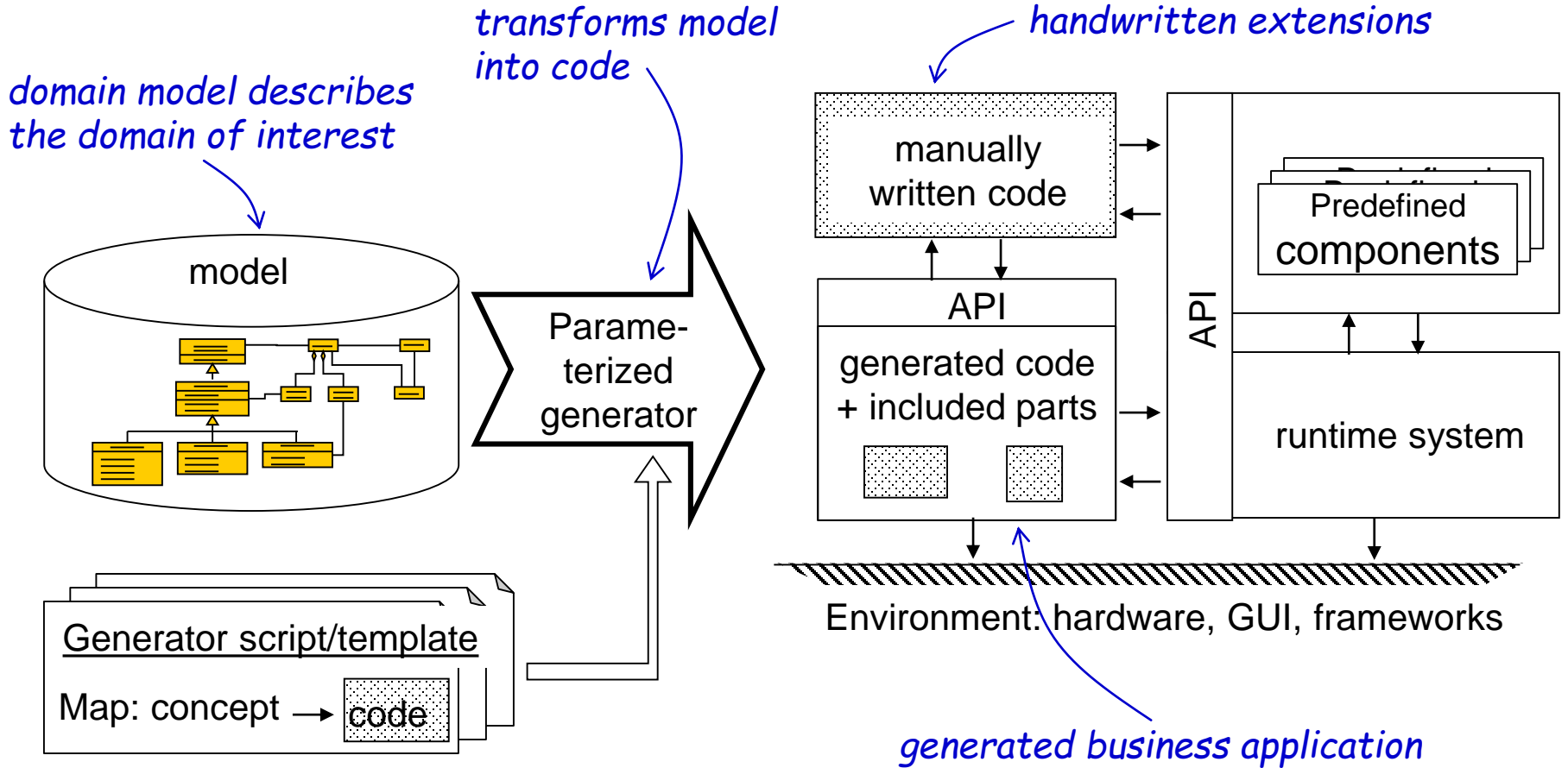
- A data-centric business application
 - is based on **structured, consistent information**
 - aims at **controlling, accessing and managing data**
 - E.g.: MS Access, human resource management, etc.
- Common business applications provide [MV09, SPHV10]
 - **CRUD** (create, read, update, delete) functionality for managed data
 - **search functionality**
 - **persistence functionality**

Motivation

- Goal: Generative development of data-centric business applications
 - generate as much code as possible
 - from an abstract description and
 - provide mechanisms for adaptations

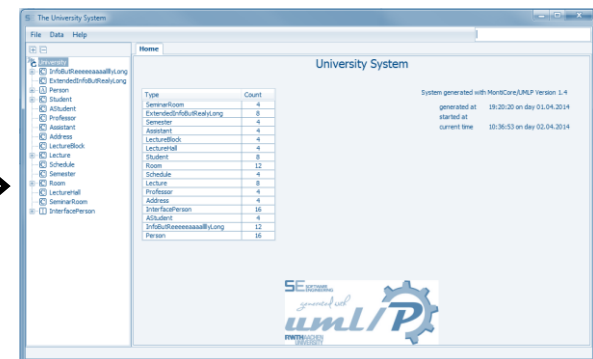
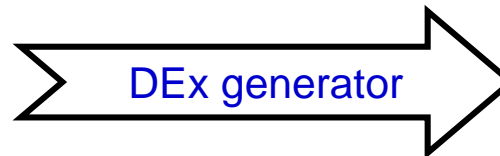
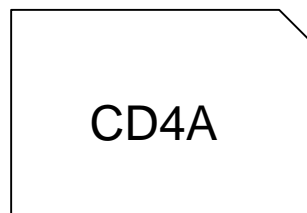
- A solution should regard
 - consistency of generated data structure at all time
 - light-weight approach (consider the different levels of users)
 - rapid and high customizability (regeneration without loss)

Generative Development Overview



Data Explorer (DEx)

- DEx is a generator for (parts of) **business applications**
- Input: **Class Diagram**
- Generated result:
 - Running **application** for managing data modeled in the class diagram
 - A **graphical user interface**
 - Support for **storing** data in the cloud

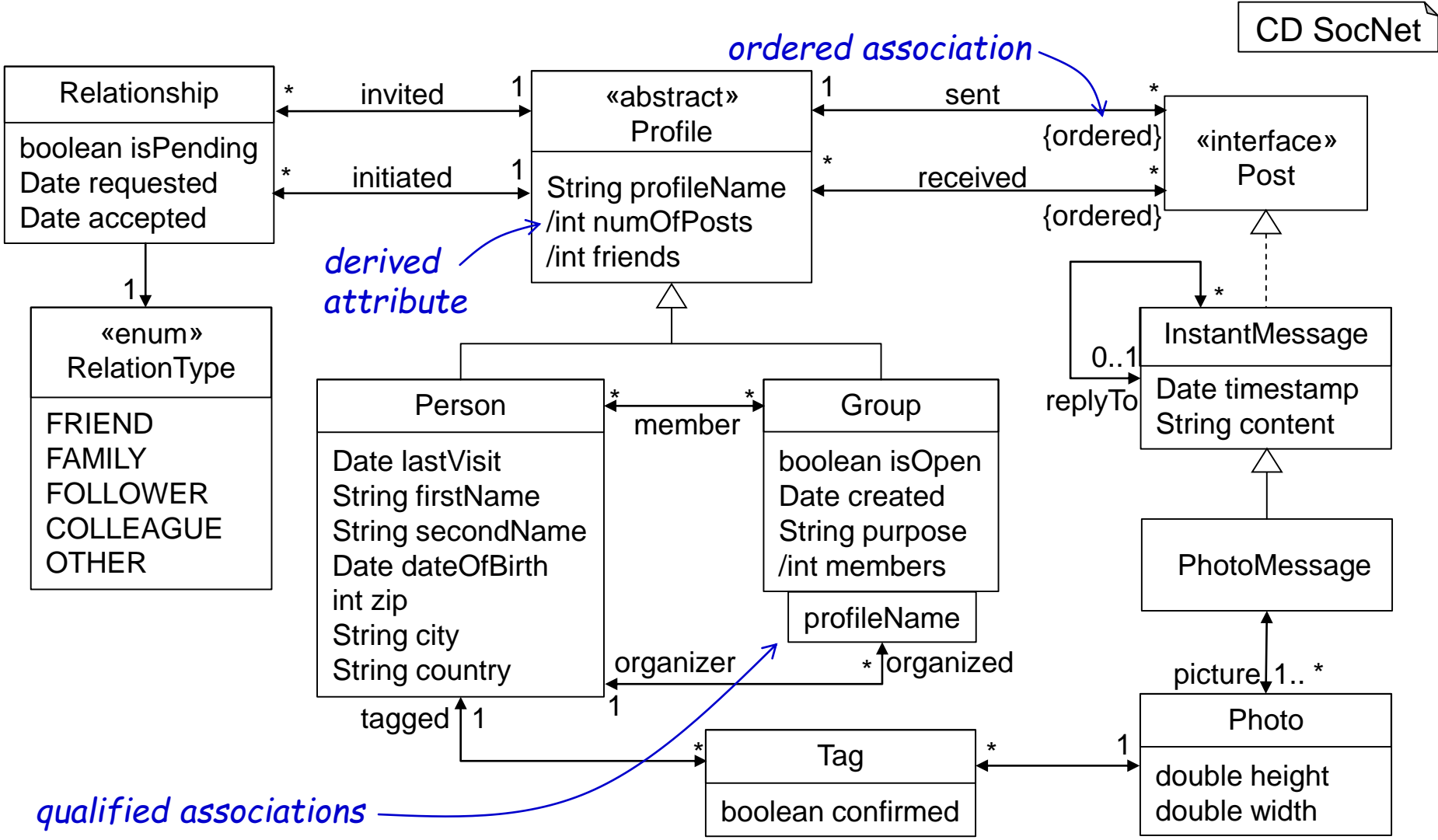


The **model**:
A textual class diagram

DEx product:
Java Swing Application

A Social Network

CD SocNet



Modeling data-centric Business Applications

- Structural description language represent the **domain model**
 - E.g.: book management, human resource management
- **CD4A** is
 - a **textual language** to describe **class diagrams for analysis**
 - a simplification of UML/P CD [Sch12]

```
classdiagram SocNet {  
  
    abstract class Profile {...}  
  
    class Person extends Profile{...}  
  
    association ... ;  
}
```



Example of generated SocNet application

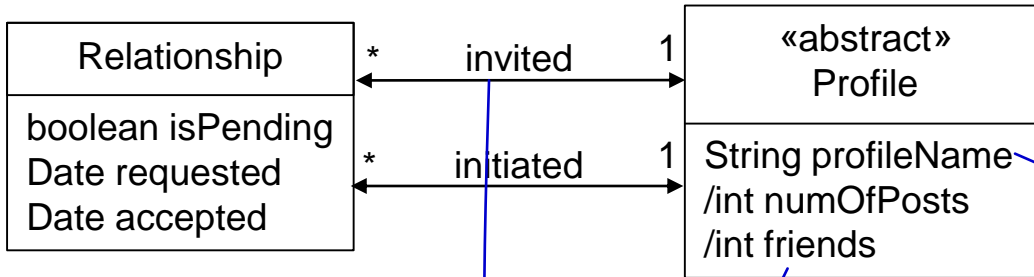
The screenshot shows a software application window titled "The SocNet System". The interface includes a menu bar (File, Data, Help), a search box, and a toolbar with icons for Add, Edit, Save, Clear, Undo, Redo, and Delete. A blue arrow points to the toolbar with the handwritten text "baisc functionality".

On the left side, there is a tree view showing a class hierarchy under "SocNet": Profile, Person, Group (highlighted), Relationship, InstantMessage, PhotoMessage, Photo, Tag, and Post. A blue arrow points to this tree with the handwritten text "class and interface".

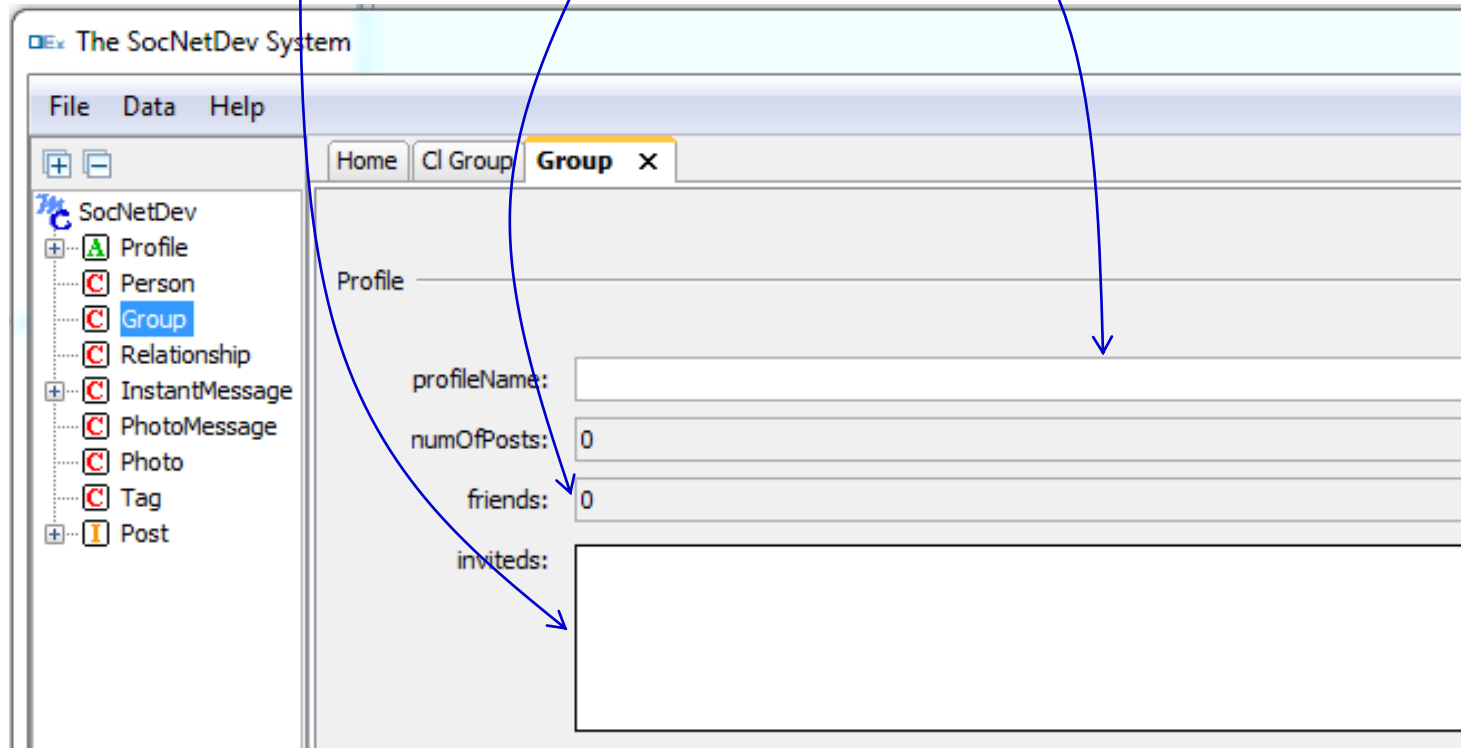
The main area displays a "Group" view with a table of data. A blue arrow points to the table with the handwritten text "managed instances".

<input type="checkbox"/>	isOpen	created	purpose	profileName	members
<input type="checkbox"/>	true	12.11.1992 01:40	New in Aachen	Tonart	
<input type="checkbox"/>	false	04.04.2002 04:37	Linkin Park	Drall	
<input type="checkbox"/>	true	19.01.2002 04:06	Filipino Martial Arts	Posthorn	
<input type="checkbox"/>	true	18.01.1991 03:50	Schrodinger humor	Griesgram	

Mapping CD4A to UI Elements



CD SocNet..



Validating Input

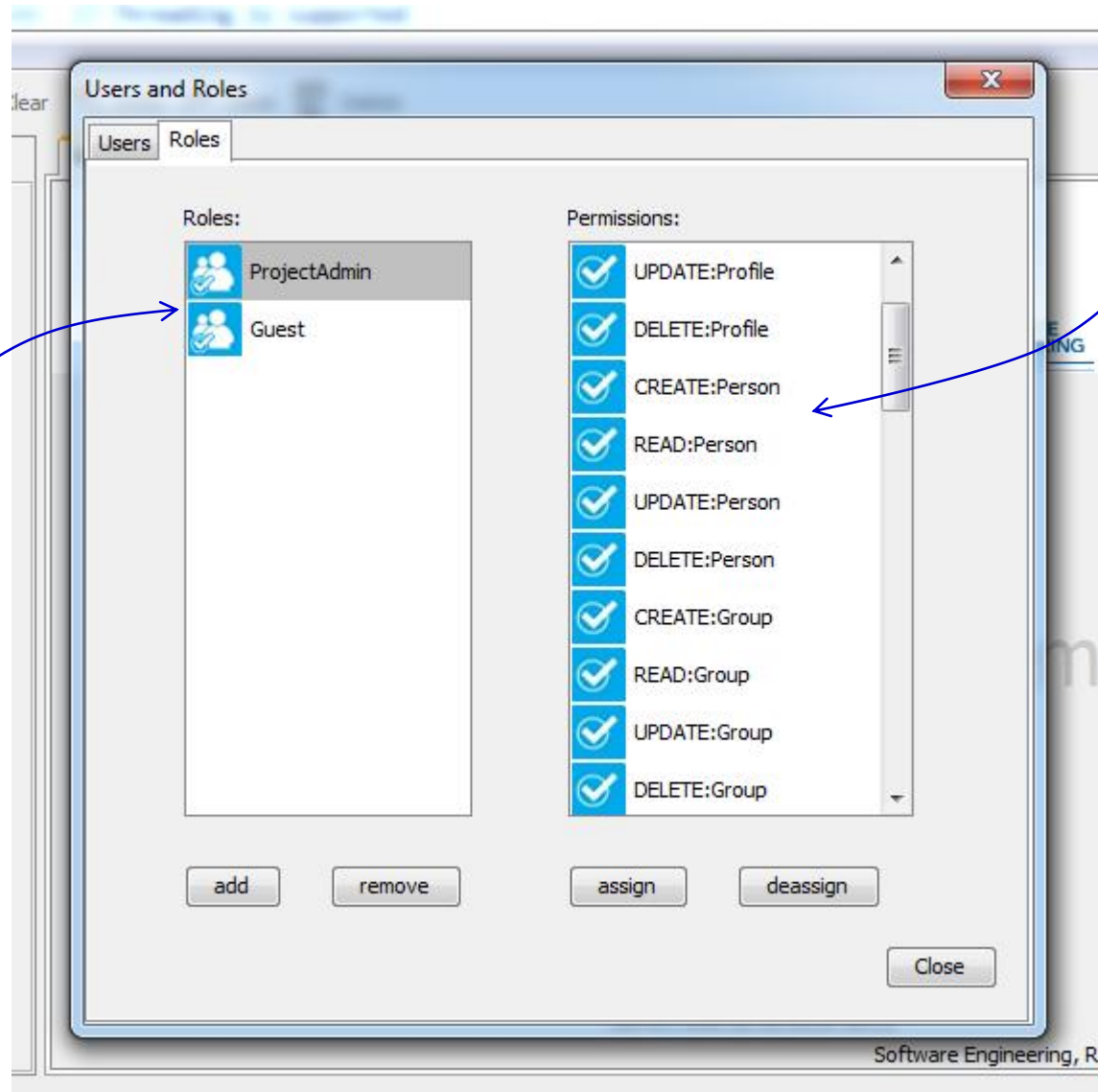
The screenshot shows a software development environment window titled "The SocNetDev System". The interface includes a menu bar (File, Data, Settings, Help), a toolbar with icons for Add, Edit, Save, Clear, Undo, Redo, and Delete, and a search bar. A left-hand sidebar displays a project tree for "SocNetDev" with sub-items: Profile, Person, Group, Relationship, InstantMessage, PhotoMessage (highlighted), Photo, Tag, Aa, Bb, and Post. The main workspace shows a form for "InstantMessage" with the following fields and controls:

- timestamp:** A date-time picker set to "22.10.15 10:08".
- content:** A text input field with a red warning icon and a blue annotation: *a valid String is required*.
- received:** A large empty text area with "add...", "show...", and "remove..." buttons to its right.
- sent:** A large empty text area with a pink background, "add...", "show...", and "remove..." buttons to its right.
- replyTo:** A text input field with "add...", "show...", and "remove..." buttons to its right.
- InstantMessage:** A text input field with an "add..." button to its right.

Two blue annotations with arrows are present:

- A blue arrow points from the text *association with cardinality [1]* to the "InstantMessage" field.
- A blue arrow points from the text *a valid String is required* to the "content" field.

Role-based Access Control

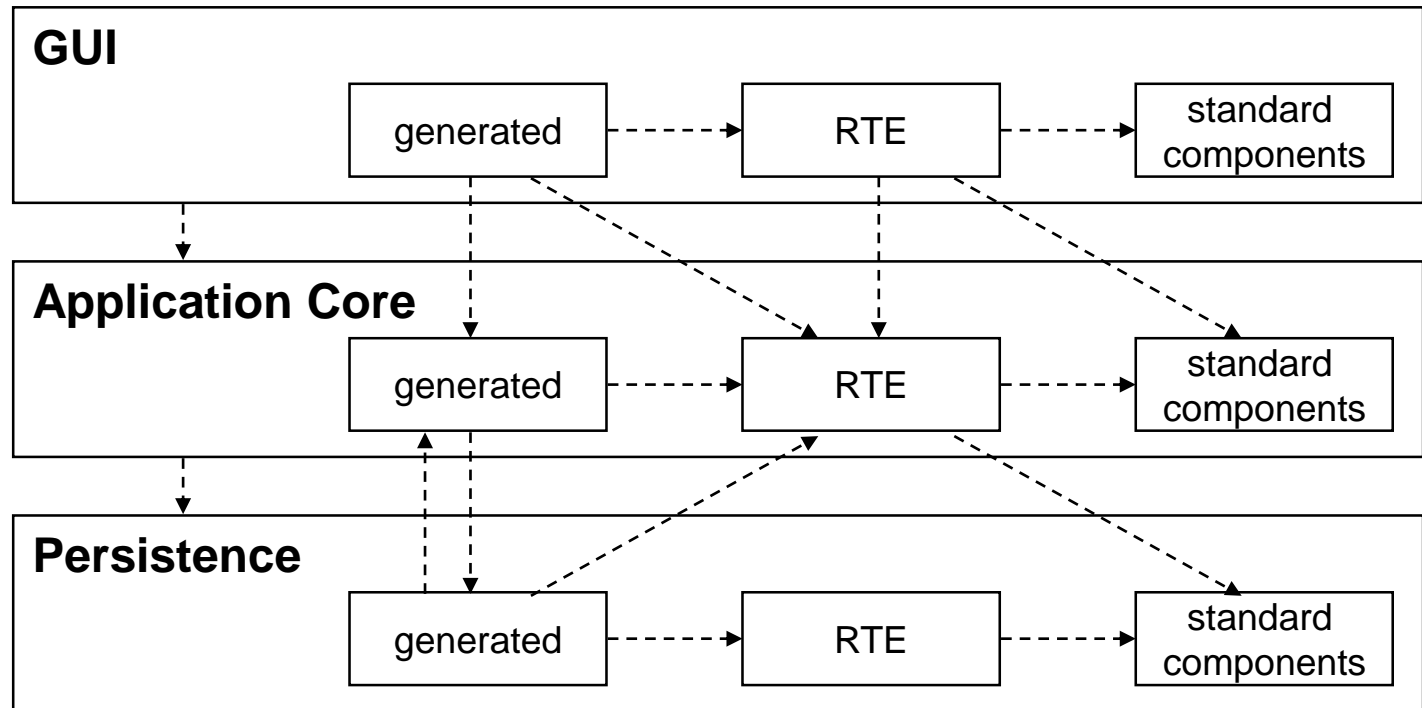


Roles and users managed with Apache Shiro

CRUD operations for every type

Generated Products

- the DEx product architecture consists of three layers



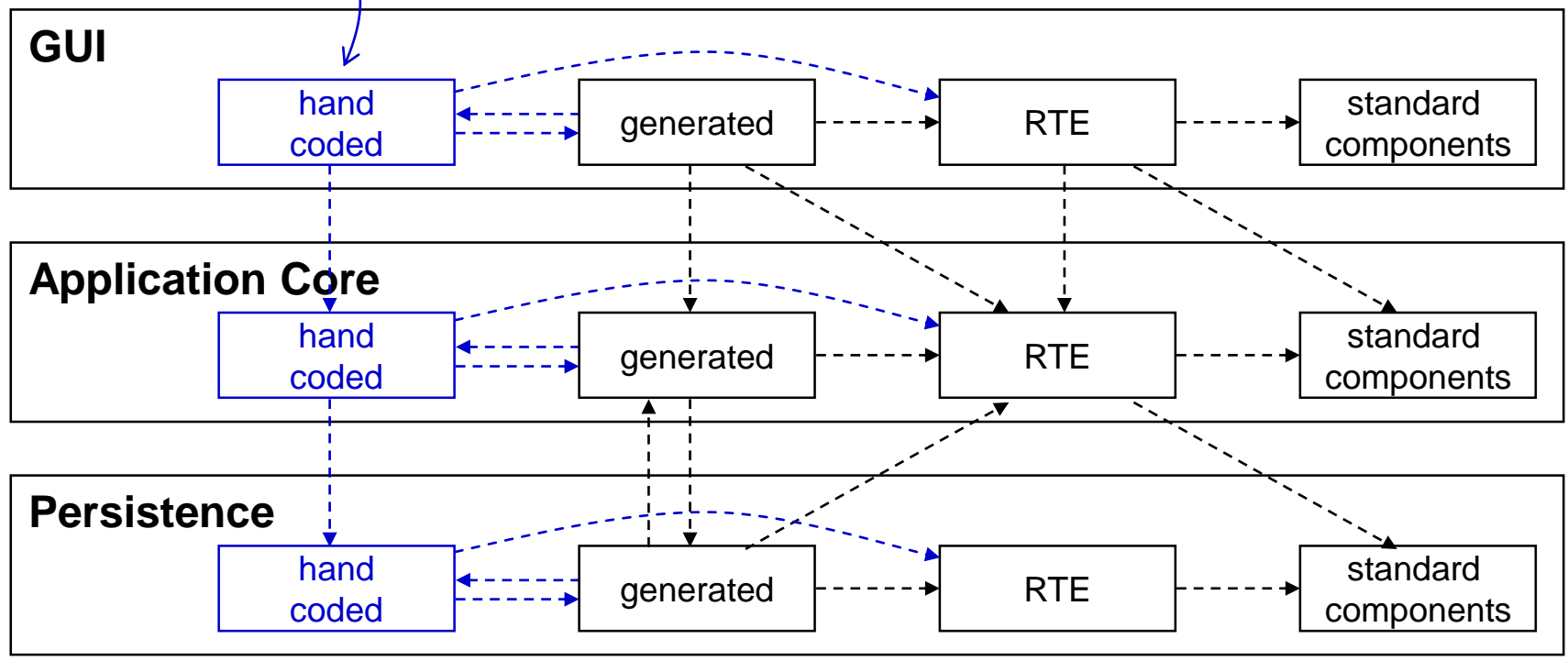
- Key: -----> reflect possible static knowledge (import)
 - Specific classes may import more general classes

Goals of Handcoding

- The goal of handcoding
 - **extend** domain model (data structure)
 - **add functionality** to generated code
 - **customize** generated code
- **DEx supports extensions** for
 - domain model by attributes and methods
 - domain model by signatures
 - generated GUI code

Overview of the Generated Product with HC

- Generated DEx product has this architecture
 - HC internal architecture is not in generators control:
 - But it is recommended to maintain this architecture



Extending the generated Product

- Required steps
 - 1. Add handwritten Java interface **XSIG**
 - 2. Add handwritten Java class **XEIMP**
 - 3. Implement new methods in **XEIMP**

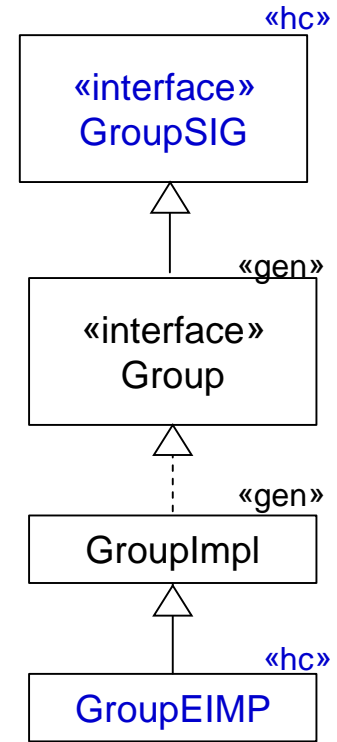
*handwritten
Java interface*

Java
«gen»

```
public interface Group extends dex.socnet.GroupSIG {  
    //...  
}
```

```
public abstract class GroupImpl implements dex.socnet.Group  
{  
    //...  
}
```

Product-CD



Conclusion

- Brief introduction into **data-centric business applications**
- **CD4A** to create domain models
- **Generative development of data-centric business applications**
 - Data Explorer (DEx) Generator
 - handwritten extensions