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

- Generator script/template
  - Map: concept ➔ code
- Parameterized generator
  - transforms model into code
- Domain model describes the domain of interest
- Manually written code
  - API
  - Generated code + included parts
  - Predefined components
- Handwritten extensions
- Runtime system
- Environment: hardware, GUI, frameworks
- Generated business application
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

Relationship
- boolean isPending
- Date requested
- Date accepted

Profile
- abstract
- String profileName
- /int numOfPosts
- /int friends

Person
- String firstName
- String secondName
- Date dateOfBirth
- int zip
- String city
- String country
- tagged

Group
- boolean isOpen
- Date created
- String purpose
- /int members

Tag
- boolean confirmed

Photo
- double height
- double width

InstantMessage
- Date timestamp
- String content

PhotoMessage
- 1.. *

Relationship
- boolean isPending
- Date requested
- Date accepted
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

- **Basic functionality**
- **Class and interface**
- **Managed instances**
Mapping CD4A to UI Elements

<table>
<thead>
<tr>
<th>Relationship</th>
<th>invited</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isPending</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Date requested</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Date accepted</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>invited</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Profile

String profileName
/int numOfPosts
/int friends
Validating Input

A valid String is required

association with cardinality [1]
Role-based Access Control

CRUD operations for every type

Roles and users managed with Apache Shiro
Generated Products

- the DEx product architecture consists of three layers

### GUI
- generated
- RTE
- standard components

### Application Core
- generated
- RTE
- standard components

### Persistence
- generated
- RTE
- standard components

- 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

![Diagram showing the generated product architecture with HC internal components and generated standard components.](image-url)
Extending the generated Product

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

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

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