11th Workshop on
„Software Engineering Education and Reverse Engineering“
23.8.2011, Ohrid, Macedonia

Domain-Driven Design and Development: Overview of Current Tools and a Sample Realization with Focus on Java

Vangel V. Ajanovski
Saints Cyril and Methodius University, Skopje

Domain-Driven Design

- Main premises
  - Primary focus should be on the problem domain and domain logic
  - Complex domain designs should be based on a model
  - Close collaboration between technical and domain experts to iteratively cut down to the heart of the issue
- Not a technology; not a methodology
  - It is a way of thinking and set of priorities
- Building domain knowledge
  - Learn the problem (*almost*) until having equal knowledge with the experts
  - Define Ubiquitous language

Naked Objects Pattern

- Related pattern that is often connected to implementation based on a DDD
- Architectural Pattern
  - All business logic should be encapsulated in the domain objects
  - User interface should be a direct representation of the domain objects
  - User interface should be created 100% automatically from the definition of the domain objects

Development Tools

- Mentioned on Wikipedia

  Actifsource, Apache Isis,
  Castle Windsor/MicroKernel, CodeFluent Entities
  DataObjects.Net, Domdrides
  ECO (Domain Driven Design), FLOW3,
  Habanero.NET, JavATE, JdonFramework,
  k-framework, ManyDesigns Portofino, Metawidget,
  Naked Objects MVC, NReco,
  OpenMDX, OpenXava, Plugger, Qi4j
  re-motion, Roma Meta Framework,
  Sculptor, Sculpture – Model Your Life, Strandz,
  Trueview for .NET, Tynamo

Interesting Java Frameworks

- Apache Isis (former Naked Objects for Java)

  „*Apache Isis is a full-stack open source application development framework, designed to let you rapidly develop domain-driven business enterprise applications*“

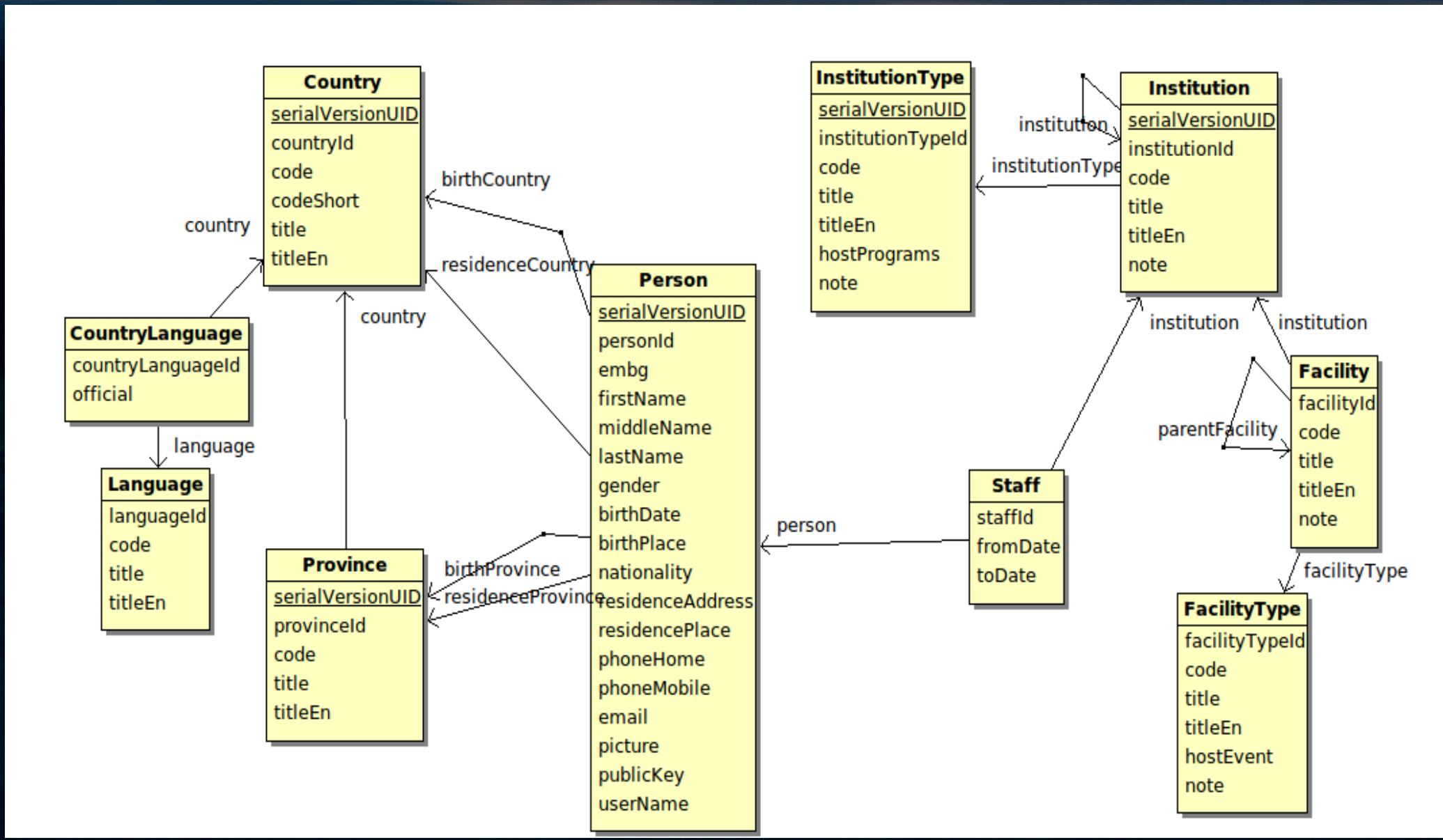
- Tynamo

  „*Tynamo is model-driven, full-stack web framework based on Tapestry 5. Tapestry-model allows rapid CRUD application development by automatically creating list, display and edit pages for POJOs*.“

- OpenXava

  „*OpenXava is an AJAX Java Framework for Rapid Development of Enterprise Applications. In OpenXava you only have to write the domain classes in plain Java to get an AJAX application ready for production.*“

Sample Model



Code generation

- BOUML tool was used for UML modeling and code generation
    - There are other tools that enable this with more success
  - Alternative: traditional DB design and reverse engineer database schema to Java classes (Hibernate tools)
- @nnotations were added for entities and relations
  - It is possible for auto export with annotations from BOUML but one has to customize the code a bit
  - Other tools have this feature builtin based on stereotypes

Final generated class

```
@Entity
@Table(name = "country", schema = "public")
public class Country {

   private long countryId;
   private String code;
   private String codeShort;
   private String title;
   private String titleEn;
```

Constructors

```
public Country() {
}

public Country(long countryId) {
   this.countryId = countryId;
}

public Country(long countryId, String code,
String codeShort, String title,
String titleEn) {
   this.countryId = countryId;
   this.code = code;
   this.codeShort = codeShort;
   this.title = title;
   this.titleEn = titleEn;
}
```

Getters / Setters

```
@Id
@GeneratedValue(strategy =
   GenerationType.IDENTITY)
@Column(name = "country_id", unique = true)
public long getCountryId() {
   return this.countryId;
}

public void setCountryId(long countryId) {
   this.countryId = countryId;
}

@Column(name = "code", length = 16)
public String getCode() {
   return this.code;
}
```

Relations

```
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "institution_id")
public Institution getInstitution() {
    return this.institution;
}

public void setInstitution(
                    Institution institution) {
    this.institution = institution;
}
```

GUI

- Based on the code of the domain objects
  - the tool generates It's own internal code for actions
- There are two main options:
  - Use a pre-compile time generator to create the source code of the GUI
  - Enrich the already compiled objects with internal code for rich interaction

Live Presentation

Further Reading

- Learning materials
  - Eric Evans – „*Domain-Driven Design: Tackling complexity in the Heart of the Software*“ – Addison Wesley, 2003 – ISBN 0-321-12521-5
  - Case Charlton – *Domain-Driven Design – Step by Step Guide* – http://dddstepbystep.com – 2009
- Platforms
  - http://incubator.apache.org/isis
  - http://www.openxava.org
  - http://www.tynamo.org

Q

and answers