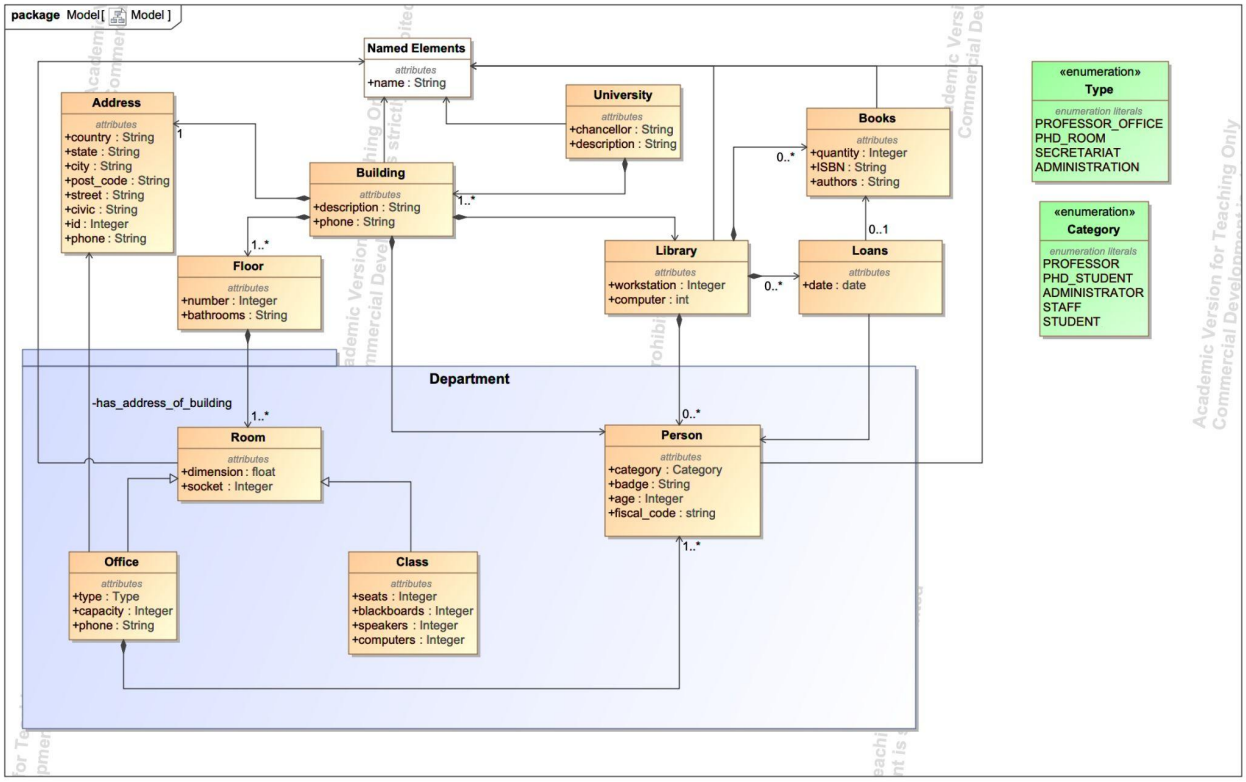
Model-Driven Engineering

Homeworks Report

Pietro Ciammaricone - 274427
Gianluca Rea - 278722

Domain

The domain which we have selected is the **University domain.** In the following, we added a diagram that represents our first version of the domain metaclasses with their relations. We have essentially a tree domain.



Unimodel Metamodel Definition

Homework 1

We defined a concept and an editor for each of the metaclasses presented previously.
We also defined two data types for necessary usage (float and data).
We defined multiple constraints in the homework.

We grouped them by metaclasses:

- **Address**
  - **post_code** length must be 5 characters long
  - **post_code** must be formed only by digit using the regex [0-9]+
  - **phone** length must be 10 characters long
  - **phone** must be formed only by digit using the regex [0-9]+
  - **city, state, country** must be formed only by letters and special characters using the regex [a-zA-Z']+
  - **id** value must be >= 1
- **Library**
  - **workstations** must be >= 1
  - **computers** must be >= 0
- **Book**
  - **quantity** must be >= 0
  - **ISBN** length must be 13 characters long
  - **ISBN** must be formed only by digit using the regex [0-9]+
- **Floor**
  - **bathrooms** must be >= 1
- **Room**
  - **sockets** must be >= 0
- **Class**
  - **seats** must be > 0
  - **blackboards** must be >= 0
  - **speakers** must be >= 0
- **Office**
  - **phone** length must be 10 characters long
  - **phone** must be formed only by digit using the regex [0-9]+
  - **capacity** must be >= 1
- **Person**
  - **age** must be >= 16 && must be <= 120
  - **fiscal_code** length must be 16 characters long

We defined two model instances as an example:

- Univaq (Università degli studi dell'Aquila)
- UPO (Università del Piemonte Orientale)

Entrambe sono formate da dati specifici casuali.

Homework 2

We defined our metamodel using EMF. We defined the two model instances as the same the previous homework:

- Univaq (Università degli studi dell'Aquila)
- UPO (Università del Piemonte Orientale)

Both the model contains one or more instance of each concept we defined at the metaclass level. This means that any concept at the metamodel level can be instantiated in our models.

Then, we used OCL to add constraints, operations and derived fields :

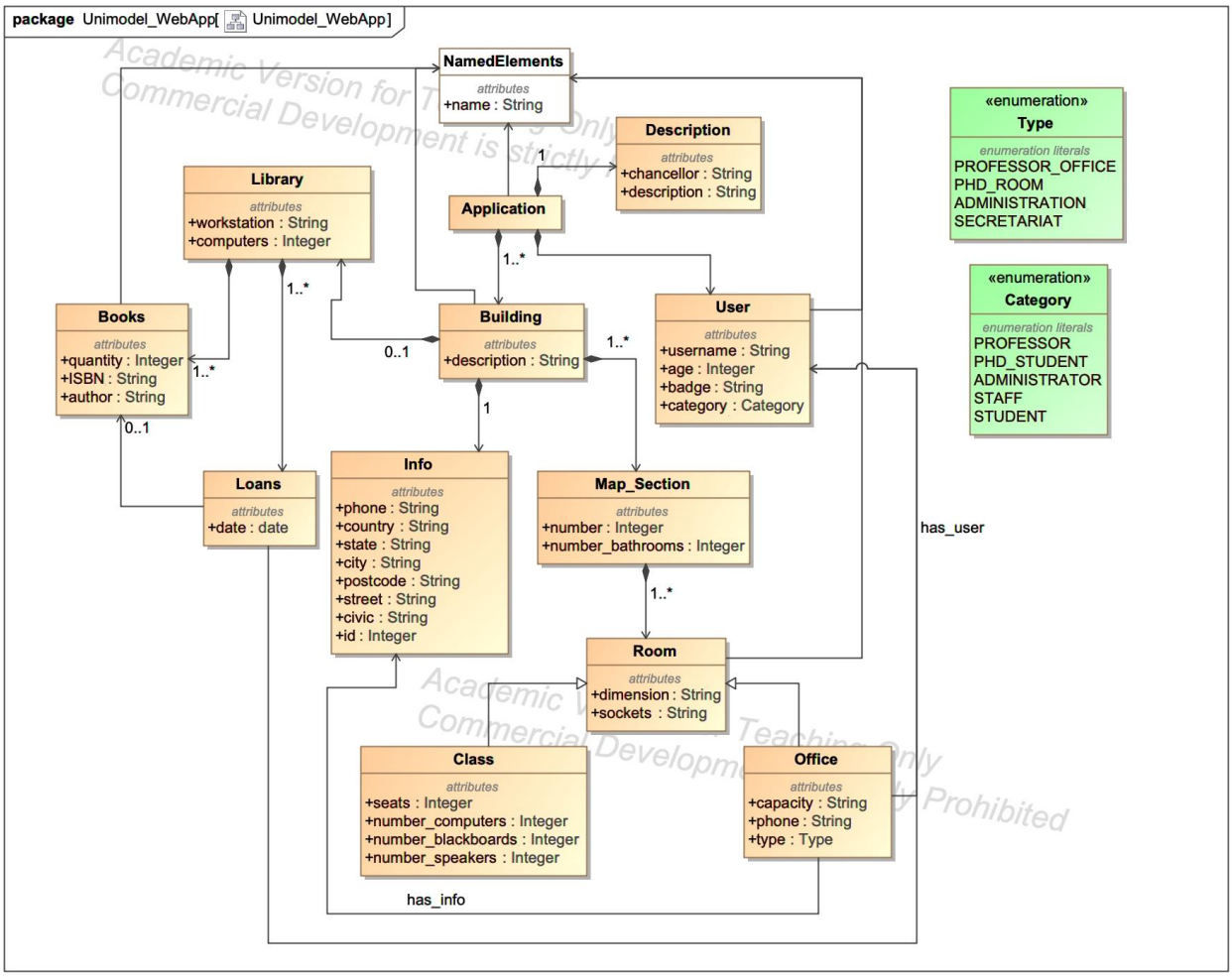
1. **Constraints**
   - **1.1. Address**
     - 1.1.1. postcode_length: the postcode length must be equal to 5
     - 1.1.2. id_number: id must be >= 1
     - 1.1.3. phone_length: the phone length must be equal to 10
   - **1.2. Library**
     - 1.2.1. workstation_positive: the number of workstations must be >= 1
     - 1.2.2. computers_positive: the number of computers must be >= 0
   - **1.3. Book**
     - 1.3.1. quantity_positive: the number of books must be >= 0
     - 1.3.2. ISBN_length: the ISBN length must be equal to 13
     - 1.3.3. SufficientCopies: the number of copies on loan must be less or equal to the number of available books (quantity)
   - **1.4. Floor**
     - 1.4.1. bathroom_positive: the number of bathrooms must be >= 1
   - **1.5. Room**
     - 1.5.1. sockets_positive: the number of sockets must be >= 0
   - **1.6. Office**
     - 1.6.1. capacity_positive: the capacity of the office must be >= 1
     - 1.6.2. person_in_office_constraint: the number of persons in the office must be less or equal to the number of available spaces (capacity)
   - **1.7. Class**
     - 1.7.1. seats_positive: the number of seats must be > 0
     - 1.7.2. blackboards_positive: the number of blackboards must be >= 0
     - 1.7.3. speakers_positive: the number of speakers must be >= 0
   - **1.8. Person**
     - 1.8.1. age_upper_bound: the age of a person must be >= 16
     - 1.8.2. age_lower_bound: the age of a person must be <= 120
     - 1.8.3. fiscal_code_length: the fiscal code length must be equal to 16

2. **Operations**
   - 2.1. isBookAvailable(): This operation checks if the number of loans of a book is less than the quantity of the book
   - 2.2. getPersonInOffice(): This operation calculates the number of people in an office

3. **Derived Fields**
   - 3.1. Book.loans: This field is auto-filled by a list of all the loans of the specific book
   - 3.2. Person.Book: This field is auto-filled by a list of all the books that are loaned to the Person
   - 3.3. Person.Loan: This field is auto-filled by a list of all the loans that are active to the Person

Homework 3

In this homework, we refined the initial metamodel updating it to the following UML Class Diagram.



Metamodel Unimodel_Webapp Definition

For this homework, we changed a few things from the original metamodel to transform the metamodel that describes a university to a web application that describes the university web app.

The attributes present in the original university class is moved to a proper class called description, meanwhile, the University class is renamed to Application. The address is renamed Info. The Floor is renamed to Map_Section. The Person is redefined as a User. The User also is contained in the application instead of the Library.

The Library instead of representing the actual building represent the online library on the website.

Two types of models transformation were executed. The first is the Model to Model transformation using ATL present in the project Unimodel2WebApp.The second transformation is a Model to Text using Acceleo.

Homework 4

In the first half of this homework, we have created an XText language for our metamodel defined in the second homework. We generated the project using the university metamodel as an input because it contains all the concepts we wanted to use for the grammar definition. The generation is been done automatically but we extended it by tweaking a few of the rules.

In the second half of this homework, we have created a straightforward graphical editor to visualize and edit the domain created with Sirius. We represented only a part of the domain. The part represents University - Building - Library - Loans - Book - Person. The diagram contains Nodes for the concepts and edges representing the references we used in our metamodel.