Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Engineering (UE), Winter Semester 2014/15
Task 2

Task 2
Model Driven Development

Task-ID: EMF

General Remarks

- The **deadline** for this work is 16.11.2014 which can also be found in Moodle (https://moodle.univie.ac.at/course/view.php?id=34717).
  The deadline is firm! No deadline extensions are given.
- You will have to present your solution – so be prepared
  (laptop, pdf of your presentation, running example, …)
- You can get up to 10 points, which have to be defended during presentation.
- This is an **individual work**. Consequently, group work is not allowed in this assignment.
- If you have problems, do not hesitate to contact the tutor (ase.tutor@swa.univie.ac.at) or us (ase@swa.univie.ac.at) via e-mail.

Submission Guidelines

- The guidelines proposed on https://swa.univie.ac.at/General_guidelines must **always** be satisfied.
- If guidelines are ignored,
  - an assessment of the tasks **cannot be guaranteed** because they are processed electronically.
  - one (1) point will be deducted from the assessment of the task for each unsatisfied guideline.
- Artifacts to upload into Moodle are (see guidelines for more information):
  One (1) ZIP archive containing:
  - 1 Eclipse project containing the ecore model(s)
  - The Eclipse projects generated from your ecore model(s).
  - 1 instance of the streaming service model!
  - Document about your assumptions (PDF)

Task Definition

Right at the Start

Please, read the guidelines at https://swa.univie.ac.at/General_guidelines, especially the part about naming conventions for your uploads! If you do not follow these guidelines, we will not be able to grade your work, because we will not be able to find your files any more (yes, it is that simple …).

1) Create a single Eclipse project that contains one or more Ecore models which represent the situation described below (same as Task 1).
2) Implement 3 constraints with OCL to ensure the following requirements:
   a. Each account may have up to 5 profiles
   b. Only customers from the US are allowed to rent DVDs.
   c. Customers are only allowed to create reviews for DVDs they have rented before.
3) Generate the model code, edit code, and editor code for the models you created.
4) Create one instance of the models you created in 1). This instance needs to contain at least one instance of each of the model elements in the respective model.

Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Engineering (UE), Winter Semester 2014/15
Task 2

The Domain and Model Description

The target domain is the same as described in Task 1. Create one or more meta-models which model and comply with the requirements described below. Think about which parts might be reusable in other contexts – these are good candidates for splitting off into separate models. If you choose to split off some parts into separate models you need to integrate them correctly (do not duplicate the elements from the other models – import and reference them in your streaming-service model).
Note that meta-models represent a high level of abstraction. Therefore, some information in the requirements description does not have to be implemented within the meta-model.

Netflix is an internet-based service that provides digital media against a monthly subscription fee. The service is available in the Americas and certain countries in Europe.

a) **Content Distribution**: The service is provided via on-demand online streaming, but users in the United States also have the option of renting a DVD or Blu-ray disk, sent by mail. The libraries of the two services have different content.

b) **Streaming Library**: The available media comprises movies, television series and music, licensed from a number of distributors, as well as a number of in-house products.

c) **Subscription Management**: Subscriber accounts pay a monthly fee after a free trial period. Each account can be associated with up to five profiles.

d) **Social Content:** Each account and profile can be shared by different individual users, who can create additional content such as favourite media queues, friend lists, reviews and recommendations within each profile. Each account is enabled for use in specific devices.

e) **Streaming Service**: The streaming service is available for a wide range of devices and platforms and offers a broad variety of available encodings and bitrates. The offered quality must be able to adapt itself to the subscriber's broadband connection.

Checklist

Before uploading your solution, ensure the following checkpoints can be marked positive:

- I followed the guidelines and conventions from https://swa.univie.ac.at/General_guidelines
- I created at least one Ecore model
- I generated the model code, edit code, and editor code for all my models.
- I created an instance of the my models and do not forget to include it in my ZIP file.
- I implemented 3 OCL constraints for my models

Tipps:

1) You might need to install additional Modeling components for this assignment. You can do this in Eclipse Modeling at Help ► Install modeling components .
2) A tutorial for working with Ecore can be found at: http://www.vogella.com/articles/EclipseEMF/article.html

050132 Advanced Software Engineering (UE), Winter Semester 2014/15

Task 2

The tutorial covers most parts of this assignment. However, it was written for an older version of Eclipse and some inconsistencies might be possible.

3) Documentation about OCL can be found at: http://help.eclipse.org/luna/index.jsp?topic=/org.eclipse.ocl.doc/help/CompleteOCL.html
4) You can use the OCLinEcore editor to write the 3 OCL constraints. (Right click on your model ► Open with ► OCLinEcore). Documentation for OCLinEcore can be found here: http://help.eclipse.org/luna/topic/org.eclipse.ocl.doc/help/OCLinEcore.html?cp=38_1_2
5) If you run into problems when comparing EMF model objects: Generated EMF models do not override equals and hashCode methods. So a comparison is always reference based.
   If you have no clue what that means, this might help you:
   http://www.javaworld.com/community/node/1006
6) Create your instances with your generated editor. You will find the created data in your workspace within the folder runtime-EclipseApplication.

Feel free to make your own assumptions. Mention them in your document.
If you have questions about the requirements, feel free to discuss them with your colleagues and us using the Moodle forum.