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

050132 Advanced Software Architecture (UE), Winter Semester 2015/16
Task 2

Task 2: Model-Driven Development

Task-ID: EMF

General Remarks

- The **deadline** for this work is the 08.11.2015, which can also be found in Moodle (https://moodle.univie.ac.at/course/view.php?id=46197). The deadline is firm! No deadline extensions are given.
- You will have to present your solution during the presentation in class: 17.11.2015 (Group 3), 18.11.2015 (Group 2), 19.11.2015 (Group 4), 20.11.2015 (Group 1)– so be prepared and bring with you the requisite materials (laptop, your presentation, running example, …)
- Your delivered solution will be graded with 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 query the Moodle forum (https://moodle.univie.ac.at/mod/forum/view.php?id=1120095), contact the tutors (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.
  - In case of any unsatisfied guidelines, three (3) points will be deducted from the assessment of the task.
- Artifacts to upload into Moodle are (see guidelines for more information):
  One (1) ZIP archive containing:
  - One (1) Eclipse project containing the Ecore model(s)
  - The Eclipse project generated from your Ecore model(s)
  - One (1) full instance of the model(s)
  - 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 …).

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

050132 Advanced Software Architecture (UE), Winter Semester 2015/16
Task 2

Task 2 Assignment

a) Implement the model of the Wikipedia system developed in Task 1 in a single Eclipse project that contains one or more Ecore model(s) which represent the situation as described in the Domain and Model Description of Task 1 (repeated below for convenience). If you choose to work upon your own model from Task 1 and determine that it is insufficient for the purposes of the exercise, you are of course free to adapt/correct it for this assignment, but please document the fact as well as the extent of the changes.

b) After you have implemented a), implement four (4) constraints in OCL to ensure the following requirements:

1. At all times, there must be at least one sysop and one administrator accounts active before users can be registered.
2. A newly registered user is autoconfirmed one week after his registration and after at least 20 edits.
3. Only autoconfirmed users (and above) are allowed to create new content items (articles, project pages, talk pages), and only administrators (and above) are allowed to delete them.
4. Suggest and implement one more constraint based on your comprehension of the Wikipedia model. Explain why you chose this particular constraint.

c) Generate the model code, edit code, and editor code, for the model(s) you have created.

d) Create one (1) instance of the model(s) you created in a), containing at least one (1) instance of the model elements present in the respective model(s).

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

050132 Advanced Software Architecture (UE), Winter Semester 2015/16
Task 2

The Domain and Model Description

a) Wikipedia is a free, online encyclopedia project that consists of its content and its community (users).

b) Content is divided into three broad categories: encyclopedia entries (articles), media (e.g. https://en.wikipedia.org/wiki/File:Great_Wave_off_Kanagawa2.jpg), and internal Wikipedia project pages (e.g. https://en.wikipedia.org/wiki/Wikipedia:Community_portal, https://en.wikipedia.org/wiki/Wikipedia:MOS, https://en.wikipedia.org/wiki/Wikipedia:AUSTRIA, or https://en.wikipedia.org/wiki/Wikipedia:Help_desk). Each of these has a relevant discussion page, and a version history page. Content belonging to internal Wikipedia material or projects forms special groups and must be clearly marked as such.

b) There are two kinds of users, registered and unregistered. Users can opt for/shift between two modes, reader and editor. Reader mode allows a user to search and browse articles and Wikipedia project pages, view media, read article talk pages, etc. Reader mode allows browsing the entire public content, including article history, but cannot alter it. Editor mode allows a user to edit articles, including moving them, and participate in discussions in article talk pages.

c) Some editor actions require the user to be registered and to possess specific editor privileges. Privileges are cumulative based on a user hierarchy, comprising simple registered users, autoconfirmed users, administrators, and sysops. A simple registered user has the same rights as an unregistered user, i.e. basic editing rights on already existing pages. An autoconfirmed user can create and move pages, and upload or move a media file. An administrator can delete pages and block users. A sysop can make a user into an administrator, revoke adminship, and block administrators.

d) Each registered user has his/her own unique username and a user page (profile) and user talk page (e.g. https://en.wikipedia.org/wiki/User:User). Unregistered users are identified by their IP address.

e) Registered users can declare membership to specific Wikipedia projects or interest groups.

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 (1) Ecore model
- I generated the model code, edit code, and editor code for all models
- I implemented four (4) OCL constraints
- I created a complete instance of my model(s)

Tips and references

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

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

050132 Advanced Software Architecture (UE), Winter Semester 2015/16
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.

- Documentation about OCL can be found at: https://wiki.eclipse.org/OCL
- You can use the OCLinEcore editor to write the OCL constraints. (Right click on your model ► Open with ► OCLinEcore). Documentation for OCLinEcore can be found here: https://wiki.eclipse.org/OCL/OCLinEcore
- 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
- Create your instances with your generated editor. You will find the created data in your workspace within the folder runtime-EclipseApplication.
- Document your thought and solution process. Feel free to make your own assumptions, but mention them in your document. If you use solutions found on the web, please record them as such, with the appropriate link.