MDE 20/21 Homeworks

The following homeworks consists of replicating the technologies presented during the MDE 2021 course. In such a context, the students should first identify a domain of interest (i.e., web cms, database, airport system, voting system, eCommerce, etc).
Once the domain of interest has been defined and approved by the team (Alfonso, Francesco, or Juri), all the homeworks will focus on the selected domain.

Homework scores:

6/30 HW1
6/30 HW2
10/30 HW3
10/30 HW4

HOMEWORK 1

Summary

This homework is about the use of the MPS platforms for modeling the defined domain.

Task 1

Define a metamodel in MPS for describing the selected domain (as illustrated during the course).
The metamodels should contain at least 5 metaclasses (10 if the group consists of **2 people**)

For the metamodel definition, we will evaluate the following things:

- Metamodel completeness;
- Homogeneity of the modeled concepts;
- Coverage of technological stuff (containment and bi-directional references, inheritance, enumerations, custom data types).

Task 2

Build an editor for the metamodel you defined. It will be assessed concerning the following criteria

- conciseness
- syntactical elegance
- coherence

Instantiate the metamodel by two concrete model instances.

Task 3 (optional)

Introduce at least 3 metamodel constraints and provide the corresponding notification in the editor in case of violations. (e.g., in the database domain if a VARCHAR has a length greater than 255, the length is visualized in red and notified.)

HOMEWORK 2

Summary

This homework is about the use of the EMF platforms for modeling the domain you defined. In particular, it requires the realization of a number of tasks by using the following one of the tools/notations:

- EMF
- Emfatic
- OCLInEcore

Task 1

Define a metamodel in Emfatic or EMF for describing your domain (as illustrated during the course).
The metamodel should contain at least the concepts identified during the lecture, e.g.,

- at least 5 metaclasses (10 if the group consists of **2 people**)
- Coverage of the following concepts:
  - inheritance,
  - relation (containment, non containment, opposite (optional)),
  - enumeration,
  - attribute;
- Each metaclass has at least one attribute or reference;
- There must be no lonely metaclasses;

For the metamodel definition, we will evaluate the following things:

- Metamodel completeness;
- Homogeneity of the modeled concepts;
- Coverage of technological stuff (containment and bi-directional references, inheritance, enumerations, custom data types);

Task 2

Instantiate the metamodel by two concrete instances:
Please, check that any concepts (represented as metaclasses) at the metamodel level can be instantiated in your models.

Task 3

Define metamodel constraints (and “critiques” in the case of EVL), operation, and derived fields in OCL or EVL/EOL.

- 3 constraints/(critique and constraints) (that is, constraints to validate the model).
- 2 operations
- 2 derived fields (only for OCL)

HOMEWORK 3

Summary

This homework is about the use of model transformations. It requires the realization of several tasks by using the following languages/frameworks:

- Epsilon (ETL and EGL)
- ATL and Acceleo

Task 1

Apply different refactory operations on the metamodel defined in the HW2 to create an updated version. Those refinements should include:

- Deletion and addition of concepts,
- Concept renaming,
- Structural refinements, for instance:
  - Adding/removing hierarchies,
  - adding/removing structural features (i.e., attributes and references)
  - Change attribute

In our envision, the refining steps should ease you in implementing task 3. At this stage, we will evaluate the complexity of the refining steps.

Task 2

Define a Model-to-Model transformation (M2M) that allows the migration of models to conform to the initial metamodel version to the one refined in task 1. For this task, feel free to choose your preferred Model-to-Model transformation (M2M) language (i.e., ATL or ETL).

Task 3

Define an M2T transformation that generates informative HTML pages for the selected domain. By resembling the demo implemented during the lectures, the M2T should generate at least an index page that summarizes the model data linked to informative details pages for the modeled concepts (e.g., terminals, sections, facilities, etc.).
For this task, feel free to choose your preferred Model-to-Text transformation (M2T) language (i.e., Acceleo or EGL).

HOMEWORK 4

Summary

This homework is about the development of the textual and graphical editors for modeling the selected domain. It requires the realization of several tasks by using the following frameworks:

- xText
- Sirius

This homework aims to understand your knowledge about the listed technologies. Because of the complexity of such technologies, **we are asking for straightforward editor implementations** that partially cover the concepts defined at the metamodel level.

Task 1

Define a complete concrete syntax for the metamodel defined in H2 by using xText. The concrete syntax should cover all the concepts defined at the metamodel level.

Suggestion: Feel free to reuse and refine the default concrete syntax provided by xText wizard.

Task 2

Define a **straightforward** graphical editor to visualize and edit your domain models by using Sirius.
In this task, we ask for a dummy graphical editor that partially covers your domain metamodel concepts. In other words, the graphical editor should provide the representations for few metamodel concepts and the relations. Try to include at least the following Sirius features:

- Node representations,
- Edge representations (both relation-based and element-based), and
- Palette operators to create such nodes and edges.