# Exercise 4 - Model Quality

**Course**: DT4250 Model-driven Development of Information Systems
**Team**: Ivo Dlouhy, Noelia Maria Palma Perez, Simon Stastny

This document describes all evaluated attributes of the model. The last section contains summary of changes done on the model (compared to first exercise).

## Physical Quality of a Data Model

**Model repository with versioning.**
The tools used for modelling does not allow this functionality. We already added this functionality by using Git versioning system that allows us to see all versions (revisions) of a document, changes between revisions and revert back to old revision at any time. We are using online service GitHub for hosting the exercises. All of the team members have access to the versioning system and can upload their work.

**Possible to see differences between versions of models.**
See above. Git service allows user to see differences between the models in plaintext. If this is not sufficient, there can be several branches with different versions. Modelling tool support is necessary to mark the changes directly in model.

**Models available for browsing on web by all users.**
See above. All code of models is available for browsing online at GitHub**.** Modelling tool support is necessary to see the model online in graphical way.

**Model language.**
English language is used for modelling, since the course is in English and every team member speaks English. No other languages were required.

**Import/Export.**
The tool used for modelling allows us to easily export the model. One of the export possibilities is archive file - this is used for uploading the model to It's Learning. Same way the base files for other assignments are imported from archive file.

## Empirical Quality of a Data Model

**Angles between edges going out from the same node should not be too small.**
The relations isCoordinated and isAttended does not conform to this directive. The angles between the relation lines in diagram should be more open. We will fix it by moving the relation lines.

**Minimise the area occupied by the diagram.**
It is as minimised as possible. It is necessary to have a wide diagram to clearly understand all the functionalities.

**Balance the diagram with respect to the axis.**
The diagram is balanced enough.

**Minimise the number of bends along the edge.**
Most of the relation lines do not need to have bends - the diagram is simple enough, so that they can be straight.

**Minimise the number of crossings between edges.**
None of the relation edges needs to cross in the diagram.

**Place nodes with high degree in the centre of the drawing.**
The node with most degrees is Course Entity. It is placed well.

**Minimise the global length of edges.**
All lengths are already minimised enough, so that we can clearly see the nodes and edges with description.

**Minimise the length of the longest edge.**
The longest edges are between Root node and Person or Course nodes. The length of these edges is equivalent, because the Root entity is on the side, and the reader can clearly see, that all of the nodes have edges to it.

**Have symmetry of sons in hierarchies. In particular, relevant when you depict generalisation-hierarchies.**
It’s symmetrical.

**Have an uniform density of nodes in the drawing.**
As this drawing is not too crowded it was easy to achieve an uniform density of nodes.

**Have verticality of hierarchical structures.**
All nodes at the same level in the hierarchy are placed in right positions along the horizontal line. Good distribution.

**Entity classes should be named with nouns and noun phrases in singular form.**
Entity classes in model has these names: Person, Teacher, Student, Course, Answer and Assigment. The Assigment entity is not noun, since there is a typographical error. We will correct it to Assignment.

**Relationship classes should be named with nouns.**

The model does not use relationship classes.

**Attributes should be named using nouns or adjectives.**
All the attributes are nouns and one adjective.

## Syntactic Quality of a Data Model

**Syntax error prevention.**
Eclipse modelling tool uses interface for designing the model. User can design the model with a mouse and default values (EClass0 as a class name) are used. This provides basic prevention of an error, when using the graphical user interface - user is not allowed to set invalid values or use invalid commands.

**Syntax error detection.**
After the user finishes design, the model is compiled and source code is generated. This is a level of syntax error detection. If there is an syntax error in the model, user is alerted via Eclipse error console. Most of the errors are prevented by using the graphical user interface (see above).

## Semantic Quality of a Data Model

**Consistency checking.**
Eclipse modelling tool offers basic consistency checking. The user is not allowed to make basic semantic errors - for example create relation without specifying the entities. In a same way, user cannot create a relation between entities, that does not exists.

**Extended semantic checking.**
The used modelling tools does not use extended semantic checks, for example you are not forced to name and set cardinality to all the relations defined in the model. More advanced and complex tool is required to perform these checks.

**Constructivity.**
The tool supports decomposition using system of packages. When making simple model, all of the packages are predefined in the system.

## Pragmatic Quality of a Data Model

**Expressiveness.**
Naming of the entities in the project is expressive, names clearly states the functionality and reason the entity is implemented.

**Clarity.**

Model is simple enough to be clear to reader.

**Localisation.**
When naming relations in model, the predicates are used on places, where it is necessary. For example Person submitted Answer, but Answer is submittedBy Person.

## Deontic Quality of a Data Model

Since this is a university course project, it is difficult to evaluate these qualities.
Cost of design and implementation can be measured using hours spend working on a project, benefit can be the final grade.

## Summary of changes in model

- Fixed small angles between edges from the same node (empirical).
- Fixed edge straightness (empirical).
- Course Assignment relation is now bi-directional (semantic).
- Person Answer relation is now bi-directional (semantic).
- Person Student Teacher hierarchy was abandoned. Course has two bi-directional relations: attended and responsible (semantic).
- Root class contains Person class and Course class (syntactic, the Root class is required by the tool, so that we can generate the code properly).

## References

J. Krogstie: Model-Based Development and Evolution of Information Systems - A Quality Approach, 2012

J. Krogstie: Slides for the course TDT4250 Model-driven Development of Information Systems, Autumn 2012
Available on It’s Learning