An online corpus of UML Design Models : construction and empirical studies

Karasneh, B.H.A.

Chapter 9

Conclusion and Future Work

In this chapter, we describe the conclusion of this research and outline the future work.

9.1 Conclusion

Our work can be positioned in the area of quality assurance for software designs. Our first effort explored the use of ontologies in assessing the severity of defects. Here we found that our automated prediction method MAPDESO performs well compared with the manual (original) classifications of the defects obtained from the conducted case studies. In addition, MAPDESO performs better than the classifiers in WEKA. We notice that our method is very practical because it is based on IEEE standard [25] for the defects' attributes and its values.

Subsequently, we explored the creation and use of a corpus of UML models (esp. class diagrams). In this thesis, we describe the design of a database which can be accessed via a web-based system. For collecting our corpus, we developed new techniques for classifying class diagrams and extracting class models from images.

We describe some basics characteristics of our corpus. We find that class diagrams are in general not so large, and it can be part of larger systems, but then the model of such a large system is split up across multiple diagrams. We investigate relations between models design metrics, and we find there are many interesting relations between diagrams design metrics, which affects the quality of these diagrams such as the relation between diagrams size and maximum coupling.

We went on to illustrate the usefulness of the corpus through additional studies. We showed how the use of a corpus can be an aid in teaching students how to design class diagrams. We used the corpus to study whether students are reliable assessors of UML class diagram designs from peer-students. The finding is that students evaluation are higher than experts. However, from the quantitative analysis, we found that the students are not eligible to evaluate other students' class diagrams, we perceive from the qualitative analysis that their feedback is similar to experts feedback.Therefore, we conclude that feedback from students is valuable and can help them for improving their design.

We showed how the corpus can be used to find projects, and various data related to the projects such as source code, documentation, test cases that may or may not be available for empirical studies.

We studied the relation between the quality of a UML class diagrams design by looking at the anti-patterns and compare this to the quality of the associated source code – also by looking at the anti-patterns in that. This study showed that anti-patterns can be detected in the design, and these anti-patterns transfer to the source code. In addition, we observe that classes in the design that have anti-patterns have more changes and bugs in the implementation. Therefore, anti-patterns in the design have a big impact on the software implementation and software maintenance.

9.2 Future Work

We believe the corpus of UML models that we have collected will be a valuable source for empirical studies in the future. We have only started to explore its characteristics and use. We are working on expanding the corpus throw extracting class diagrams from documents in Word or Pdf format. More meta-data and related data: linking diagrams to source code is desirable. For this, we are working on more advanced crawling of open source repositories. Another refinement would be to automatically separate reverse engineered diagrams from forward designed diagrams. The UML Repository can be extended to include other types of UML diagrams. Therefore, we are going to include other types of diagrams (sequence diagrams and use case and use case).

We believe that applying machine learning for detecting patterns in the corpus is very useful for learning layout of diagrams, learning roles of classes that proposed by Wirfs-Brock [127]. She suggests roles such as: information holder, controller, decider, and user-interface. From this and using machine learning techniques we try to investigate in which combinations do such roles appear in design? Moreover is there a 'grammar' that can generate well designs or recognize violations of good design combinations?

In addition, learning good and bad naming practices is useful and affects quality attributes of designs such as understandability and layout.

For software quality assurance, using UML repository as a benchmark: i.e. find out which ranges of metrics (threshold) are considered reasonable as a function of the size of a diagrams.

We believe that the repository should be supported with a UML CASE tool, and we are establish a link with Web-UML editor with the repository. At the moment, models in the repository can be opened and edited online via Web-UML. At the end, the data that is collected from Web-UML editor such as class diagrams, user tracking, and feedbacks are going to be stored in the repository. This will enrich the data in the repository and enrich experiment and research that can be conducted.