



Universiteit  
Leiden  
The Netherlands

## Interactive scalable condensation of reverse engineered UML class diagrams for software comprehension

Osman, M.H.B.

### Citation

Osman, M. H. B. (2015, March 10). *Interactive scalable condensation of reverse engineered UML class diagrams for software comprehension*. Retrieved from <https://hdl.handle.net/1887/32210>

Version: Not Applicable (or Unknown)

License: [Leiden University Non-exclusive license](#)

Downloaded from: <https://hdl.handle.net/1887/32210>

**Note:** To cite this publication please use the final published version (if applicable).

# STELLINGEN

behorende bij het proefschrift

## **Interactive Scalable Condensation of UML Reverse Engineered Class Diagrams for Software Comprehension**

*Mohd Hafeez Osman*

1. In open source software development: the focus of UML modeling shifts from structural aspects in the early phases of development to dynamic/behavioral aspects in the later stages of development. [Chapter 3]
2. In evolving software, it is not only the creation of new features that triggers the updating of documentation, but also the arrival of new groups of developers that triggers this. [Chapter 3]
3. Even though class diagrams are primarily structural, software developers tend to create a “story” from a class diagram that explains the role and responsibility of classes, by using the combination of the class elements names and the class relationships. [Chapter 5]
4. A combination of object-oriented design metrics and class names is the best set of features for training a classifier to decide on class inclusion/exclusion for class diagrams simplification. [Chapter 7 & 8]
5. The best classifier for deciding on class inclusion/exclusion for simplifying class diagrams is the Random Forests classification algorithm. This Random Forest algorithm performs robustly across all datasets, some of which are highly imbalanced. [Chapter 7 & 8]
6. While open source software repositories are commonly used for empirical studies in software engineering, this is not much the case for studies into the use of UML and software modeling. We assert that open source projects are also suitable for studying the usage of UML. [SE]
7. Software owners rarely realize that updating software documentation is a good investment. [SE]
8. Software design is an art of logic. [any]
9. Manually cleaning up of datasets has as a positive side-effect that it makes you better understand the essence of the data and the behavior you are trying to predict. [any]
10. Research is about the journey; a positive result is a bonus. [any]