



Universiteit
Leiden
The Netherlands

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

Karasneh, B.H.A.

Citation

Karasneh, B. H. A. (2016, July 7). *An online corpus of UML Design Models : construction and empirical studies*. Retrieved from <https://hdl.handle.net/1887/41339>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/41339>

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

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/41339> holds various files of this Leiden University dissertation.

Author: Karasneh, B.H.A.

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

Issue Date: 2016-07-07

An Online Corpus of UML Design Models:
Construction and empirical studies

Bilal Karasneh

July 2016



- The author of this PhD thesis was partially financed by Erasmus Mundus program (JOSYLEEN)
- The author of this PhD thesis was employed at Leiden University

An Online Corpus of UML Design Models:
Construction and empirical studies

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op graag van Rector Magnificus Prof. Mr. C.J.J.M. Stolker
volgens besluit van het College voor Promoties
te verdedigen op donderdag 7 juli 2016
klokke 10:00 uur

door

Bilal Karasneh
geboren te Irbid, Jordan
in 1982

Promotiecommissie

Promotoren:

Prof. Dr. Joost N.Kok	Universiteit Leiden
Prof. Dr. Michel R. V. Chaudron	Chalmers Universitet and Göteborgs Universitet, Sweden

Commissieleden:

Prof. Dr. Ir. Thomas H. W. Bäck	Universiteit Leiden
Dr. Regina Hebig	Chalmers Universitet and Göteborgs Universitet, Sweden
Prof. Dr. Aske Plaat	Universiteit Leiden
Prof. Dr. Ir. Bedir Tekinerdogan	Wageningen University

*To the spirit of my father Hikmat Karasneh,
and to my mother Maryam Obeidat*

Contents

List of Figures	v
List of Tables	ix
1 Introduction	1
1.1 Problem Statement	2
1.2 Objective of the Study	2
1.3 Research Methodology	3
1.4 Contributions	3
1.5 Dissertation Outline	5
1.6 Publications	6
2 Background	9
2.1 Quality Models	9
2.1.1 Software Quality Models	10
2.1.2 Measuring Software Quality	11
2.2 Severity of Software Defect	13
2.3 Ontologies in SE	14
2.3.1 Ontology Editors	14
2.3.2 Web Ontology Language	15
2.4 Unified Modeling Language	16
2.4.1 Class Diagrams	17
2.4.2 Sequence Diagrams	17
2.4.3 Use Case Diagrams	17
2.4.4 Challenges of modeling and studying using UML	17
2.5 Repositories in SE	18

3	A Method for Automated Prediction of Defect Severity Using Ontologies	21
3.1	Approach	22
3.1.1	Developing the Ontology	23
3.1.2	The Method Flow	28
3.2	Case Studies	28
3.2.1	Data Collection	30
3.2.2	Data Analysis and Conversion	30
3.2.3	Data Classification	31
3.2.4	Results and Comparison	32
3.3	Validation	33
3.3.1	Approach - VCS	34
3.3.2	Results and Comparison	35
3.3.3	Validation of the Results	36
3.3.4	Comparison	37
3.4	Related Work	43
3.5	Threats to Validity	45
3.5.1	Conclusion Validity	45
3.5.2	Internal Validity	45
3.5.3	Construct Validity	45
3.6	Conclusions	45
3.7	Future Work	47
4	Establishing an Infrastructure for Empirical Research on UML Diagrams	49
4.1	Overview of UML Crawler	50
4.1.1	Methodology of making UML Crawler	51
4.1.2	Differences with other solutions	51
4.1.3	Crawler Requirement	52
4.1.4	Using Google Images	52
4.1.5	Implementation of UMLCrawler	53
4.1.6	Crawler Database	54
4.1.7	Limitations of UMLCrawler	55
4.2	UML Image Classifier	55
4.2.1	Classifier Approach	56
4.2.2	Experiment Description	59
4.2.3	Classification Results	60
4.2.4	Image Processing Time	61
4.3	Extracting UML Models From Images	61
4.3.1	Approach of Img2UML	61
4.3.2	Why UMLCrawler, UMLImgClassifier and Img2UML	73
4.3.3	Validation of Img2UML	73
4.3.4	UML Use case	74
4.4	Related Work	74

4.5	Conclusion and Future Work	76
5	Models-db.com: An Online Repository for UML Models	77
5.1	Related Work	78
5.2	Usefulness of the Repository	79
5.3	Data Collection Approach	80
5.3.1	Difficulties	80
5.3.2	Collecting Approach	81
5.4	Repository Overview	82
5.5	Repository Schema	83
5.6	Repository Services	87
5.6.1	Searching for Models	87
5.6.2	Performing Experiments and Online Questions	90
5.6.3	Similarity Between Diagrams	91
5.6.4	Visualize Search Result	91
5.6.5	Uploading Models	94
5.7	Conclusion and Future Work	94
6	UML Repository As Benchmark for Quality Analysis	97
6.1	Common Characteristics of Class Diagrams	98
6.1.1	The Size of Class Diagrams	98
6.1.2	Maximum Coupling	98
6.1.3	The Relation between Class Size and Max. Coupling	99
6.1.4	Discussion	100
6.2	Studying the Relation between Design Quality and Source Code	101
6.2.1	Relation between Software Design and Source Code	102
6.2.2	Effects of Anti-patterns on Software Quality	104
6.2.3	Effects of Anti-patterns in design on Software Changes and Faults	114
6.3	Threat to Validity	117
6.3.1	Construct Validity	117
6.3.2	Internal Validity	117
6.3.3	External Validity	118
6.4	Conclusion and Future Work	118
7	Quality Assessment of UML Class Diagrams	121
7.1	Related Work	123
7.2	Experiment Design	124
7.2.1	Approach	124
7.2.2	Participant	124
7.2.3	Evaluation Form	125
7.2.4	Modeling Assignment	126
7.3	Comparing Model Evaluation	126

7.3.1	Experts Evaluation and Students <i>self</i> Evaluation	126
7.3.2	Experts Evaluation and Students <i>peer</i> Evaluation	126
7.4	Results and Analysis	126
7.4.1	Quantitative Analysis	126
7.4.2	Qualitative Analysis	130
7.5	Discussion	132
7.6	Threats to Validity	133
7.6.1	Internal Validity	133
7.6.2	External Validity	133
7.7	Conclusion and Future Work	133
8	Using Examples for Teaching Software Design	135
8.1	Related Work	137
8.2	Research Questions and Hypotheses	138
8.3	Experiment Design	138
8.3.1	Method	139
8.3.2	Operation	139
8.3.3	Evaluation	140
8.3.4	Participant	140
8.3.5	Data Collection	140
8.4	Results	141
8.4.1	Experts Evaluation	141
8.4.2	Models Improvements	144
8.4.3	Post-assignment Questionnaire	144
8.5	Discussion	149
8.6	Threats to Validity	153
8.6.1	Internal Validity	153
8.6.2	External Validity	153
8.7	Conclusion and Future Work	154
9	Conclusion and Future Work	155
9.1	Conclusion	155
9.2	Future Work	156
	Bibliography	159
	Summary	169
	Samenvatting	173
	About the Author	177

List of Figures

2.1	Framework of ISO 25010 quality models	11
2.2	Framework of (Lange and Chaudron) for quality of UML models	12
2.3	The Taxonomy of UML Diagram Types	16
3.1	The created classes and properties for the ontology	24
3.2	Class Defect, its subclasses	25
3.3	Activity diagram for the prediction of defects' severity levels	29
3.4	Percentages of the 80 defects classified into the same severity levels (SLs), lower SLs and higher SLs by the ontology compared with the original classifications from CS1 and CS2	33
3.5	Percentages of the 50 defects classified into the same severity levels (SLs), lower SLs and higher SLs by the ontology compared with the original classification from VCS	36
3.6	The results for precision per severity level for the six classifiers and for MAPDESO	40
3.7	The results for recall per severity level for the six classifiers and for MAPDESO	41
3.8	The results for F-measure per severity level for the six classifiers and for MAPDESO	42
3.9	Summary of the comparison between the six classifiers and MAPDESO	42
4.1	Overall Classification Process	56
4.2	Image processing	57
4.3	Three different left-leaning diagonal lines and how they look like in pixels	63
4.4	Flowchart of the horizontal lines detection algorithm	64
4.5	Flowchart of the dashed horizontal lines detection algorithm	65
4.6	UML Class Diagrams Example	66
4.7	UML Class Diagrams before the recognition	69

4.8	UML Class Diagrams after the recognition	69
4.9	UML Sequence Diagram before the recognition	70
4.10	UML Sequence Diagram after the recognition	70
4.11	UML Use case before the recognition	72
4.12	UML Use case after the recognition	72
5.1	Distribution of experiment diagrams size	83
5.2	Distribution of project diagrams size	84
5.3	Distribution of student diagrams size	84
5.4	Distribution of web diagrams size	85
5.5	Database Schema	86
5.6	Result page of searching for the class name "reservation"	89
5.7	Question(s) page	90
5.8	Similarity between two class diagrams	92
5.9	Similarity between two class diagrams	93
5.10	Relation between operation and coupling	93
5.11	Bubble chart for the relation between classes and coupling	94
5.12	Upload Project form	95
6.1	Size of class diagrams in the repository	99
6.2	Distribution of class diagrams size in the repository	99
6.3	Maximum coupling for diagrams in the UML Repository	100
6.4	Relation between Diagrams size and Max. coupling	101
6.5	Occurrences of the complex class anti-pattern in class diagrams and source code	111
6.6	Occurrences of the large class anti-pattern in class diagrams and source code	112
6.7	Occurrences of the Lazy class anti-patterns in class diagrams and source code	112
7.1	Experts and students evaluation (self-evaluation) for Understandability	127
7.2	Experts and students evaluation (self-evaluation) for Layout	127
7.3	Experts and students evaluation (peer-evaluation) for Understandability	128
7.4	Experts and students evaluation (peer-evaluation) for Layout	128
8.1	Experimental Approach	139
8.2	Evaluation of RG models and CG models	142
8.3	Evaluation of CG models created during the experiment and after their improvement using the repository	143
8.4	Using examples of class diagrams helps to create better design	145
8.5	Usefulness of having multiple examples for the same application domain in the repository	145

8.6	Rate of the relevant class diagrams found in the repository to the design assignment	146
8.7	Rate of the quality of class diagrams found in the repository	146
8.8	Using UML Repository is more helpful than searching for examples on the internet	147
8.9	Usefulness of searching based on class-, attribute- and operation names for finding relevant models	147
8.10	Time spent by students using the repository	149
8.11	Time spent by students using the repository	149
8.12	Time spent by students using the repository	150

List of Tables

3.1	Example of defects report of the project CS1 converted to IEEE standard [25]	26
3.2	Classification rules of detecting severity of defects	27
3.3	Number of fixed defects according to the severity levels from project CS1	31
3.4	Number of fixed defects according to the severity levels from project CS2	31
3.5	The relation between the severity levels from the IEEE Standard [25], CS1 and CS2	32
3.6	Summary of the results from the comparison using a confusion matrix (CS1 and CS2)	32
3.7	Number of fixed defects according to the severity levels from the project in VCS	34
3.8	The relation between the severity levels from the IEEE Standard [25] and VCS	35
3.9	Summary of the results from the comparison using a confusion matrix (VCS)	35
3.10	The results from classifying the test data (VCS data) by the six chosen classifiers and by MAPDESO	39
3.11	Summary of the comparison between the six classifiers and MAPDESO	40
4.1	Table "Img", contains information about images that are downloaded by the crawler	54
4.2	Table Blacklist, contains blacklist URLs	54
4.3	Extracted Features	58
4.4	Confusion Matrix	59
4.5	Result of InfoGain	60
4.6	Sensitivity and Specificity Scores for all Features	60
4.7	Confusion Matrix – (LR) classification	61

5.1	Summary of Models in the Repository	83
6.1	Descriptive statistics of the size of class diagrams in the repository	98
6.2	Descriptive statistics of Max. coupling in class diagrams in the repository	100
6.3	Descriptive statistics of Max. coupling in class diagrams in the repository	102
6.4	Means of classes Changes in ArgoUML and Wro4j	103
6.5	Means of classes faults in ArgoUML and Wro4j	103
6.6	Correlation between Changes, LOC and AvgCyc	104
6.7	Correlation between Changes, LOC and AvgCyc	104
6.8	Studied Software Systems	106
6.9	Summary of number of Classes in class diagrams versus in source code	108
6.10	Proportion of classes that exist in both class diagrams and source code	109
6.11	Anti-patterns detection in both class diagrams and source code	110
6.12	Proportion of classes in class diagrams that transfer same anti-patterns to the source code	110
6.13	Summary of classes used in the experiment	115
6.14	Means of classes changes in ArgoUML and Wro4j	115
6.15	Means of classes faults in ArgoUML and Wro4j	116
6.16	Correlation between Changes, LOC and AvgCyc	116
6.17	Correlation between faults, LOC and AvgCyc	117
7.1	Results of Multivariate General Linear Model	129
7.2	Description of Experts and Students Evaluation	129
7.3	Correlation of Experts and Students peer-Evaluation	130
7.4	Features that Experts and Students Focus on When They Evaluate Un- derstandability	131
7.5	Features that experts and students focus on when they evaluate Layout	131
7.6	Features that experts and students focus on when they evaluate Com- pleteness	132
8.1	Results of Students t-test one tail	142
8.2	Results of students t-test and Mann Whitney test (One tail) of the CG .	143