



Universiteit
Leiden

The Netherlands

Studies into interactive didactic approaches for learning software design using UML

Stikkolorum, D.R.

Citation

Stikkolorum, D. R. (2022, December 14). *Studies into interactive didactic approaches for learning software design using UML*. Retrieved from <https://hdl.handle.net/1887/3497615>

Version: Publisher's Version

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

Downloaded from: <https://hdl.handle.net/1887/3497615>

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

Studies into Interactive Didactic Approaches for Learning Software Design Using UML

Dave R. Stikkorum

December 2022

Cover design assistance: Robbert-Jan van Ommeren
This cover has been designed using assets from Freepik.com

Copyright ©2022 by Dave Stikkolorum

Studies into Interactive Didactic Approaches for Learning Software Design Using UML

Proefschrift

ter verkrijging van

de graad van doctor aan de Universiteit Leiden,

op gezag van rector magnificus prof.dr.ir. H. Bijl,

volgens besluit van het college voor promoties

te verdedigen op woensdag 14 december 2022

klokke 15:00 uur

door

Dave Roy Stikkolorum

geboren te Den Haag, Nederland

in 1976

Promotores:

Prof. dr. M.R.V. Chaudron

Prof. dr. A. Plaat

Co-promotor:

Dr. P.W.H. van der Putten

Promotiecommissie:

Prof. dr. H.C.M. Kleijn

Prof. dr. T.H.W. Bäck

Dr. G.J. Ramackers

Prof. dr. G. Robles

Dr. E. Kang

Prof. dr. ir. J.M.W. Visser

University Rey Juan Carlos
Carnegie Mellon University

Contents

I	Introduction, Context and Problem Statement	1
1	Introduction – Challenges in Software Design Education	3
1.1	Problem Context – Education in Software Design	3
1.2	Background - UML, Design Principles, Tools, Agile Development and Theories of Education and Learning	4
1.3	Problem Statement	12
1.4	Research Objective and Research Questions	13
1.5	Research Approach	14
1.6	Contribution and Dissertation Structure	16
1.7	Publications	18
II	Software Design Comprehension	23
2	An Instrument for Measuring Design Skills	25
2.1	Introduction	26
2.2	Related Work	27

2.3	Method	27
2.4	Results	31
2.5	Discussion	34
2.6	Conclusions and Future Work	34
III	Tooling	37
3	Exploring the Application of Game-Based Learning in Software Design Education	39
3.1	Introduction	40
3.2	Related Work	42
3.3	Method	44
3.4	Game Design	45
3.5	The Game	55
3.6	Evaluation	57
3.7	Discussion	58
3.8	Conclusion	60
4	WebUML - a UML Class Diagram Editor for Research and Online Education	63
4.1	Introduction	63
4.2	Related Work	65
4.3	Overview of WebUML	69
4.4	WebUML in research activities	71
4.5	Discussion	72
4.6	Conclusion and Future Work	73

IV	Student Guidance and Feedback	75
5	Strategies of Students Performing Design Tasks	77
5.1	Introduction	78
5.2	Related Work	79
5.3	Method	80
5.4	Results	82
5.5	Discussion	86
5.6	Threats to Validity	89
5.7	Conclusions and Future Work	89
6	Uncovering Common Difficulties of Students Learning Software Design	91
6.1	Introduction	92
6.2	Related Work	93
6.3	Method	94
6.4	Results	99
6.5	Discussion	104
6.6	Threats to Validity	105
6.7	Conclusion and Future Work	106
7	Teaching of Agile UML Modelling: Recommendations from Students' Reflections	107
7.1	Introduction	108
7.2	Related Work	109
7.3	Teaching Method	111
7.4	Research Method	112

7.5	Results and Discussion - Points of Concern for Education	113
7.6	Additional Discussion - Implications for Education	121
7.7	Threats to Validity	122
7.8	Conclusion and Future Work	123
8	Evaluating Didactic Approaches used by Teaching Assistants for Software Analysis and Design using UML	125
8.1	Introduction	126
8.2	Related Work	127
8.3	Research Method	129
8.4	Results	133
8.5	Discussion	144
8.6	Recommendations for Future Deployment of TAs in SAD Courses . . .	147
8.7	Threats to Validity	148
8.8	Conclusion	149
9	Towards Automated Grading of UML Class Diagrams with Machine Learning	151
9.1	Introduction	152
9.2	Related Work	153
9.3	Method	154
9.4	Results	157
9.5	Discussion	161
9.6	Conclusion	163
10	An Online Educational Agent: Automated Feedback for Designing UML Class Diagrams	165

10.1 Introduction	166
10.2 Related Work	168
10.3 Research Approach	173
10.4 Results	179
10.5 Discussion - Strengths, Possibilities, Limitations and Future Directions	181
10.6 Conclusion	182
 V Student Engagement	 185
 11 A Workshop for Agile Modelling	 187
11.1 Introduction	188
11.2 Related Work	190
11.3 Approach	191
11.4 Results	196
11.5 Discussion	200
11.6 Limitations and Threats to Validity	201
11.7 Conclusion and Future Work	202
 VI Reflection	 203
 12 Conclusion and Future Work	 205
12.1 Reading Guide	205
12.2 Research Objectives	206
12.3 RQ1 – Can We Assess How Good Students Are at Designing Software? .	208

12.4 RQ2 – What guidance do students need to improve their understanding of designing software?	211
12.5 RQ3 – How can we increase the motivation and engagement of students for learning software design?	217
12.6 Supportive Tooling	220
12.7 Future Work	222
Bibliography	223
Acknowledgements	241
Summary	243
Samenvatting	247
About the Author	251
Appendices	
Appendix A Overview Recommendations for Teaching Software Modelling	254
Appendix B Coffee Machine Assignment	264
Appendix C Tank Assignment	265
Appendix D Grading Rubric	266
Appendix E Class Diagram Evaluation Criteria	267