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).

# Bibliography


[1] Ritu Agarwal and AP Sinha. Object-oriented modeling with UML: a study of developers' perceptions. *Communications of the ACM*, 46(9):248–256, 2003. (cited on page 79).

[2] Seiko Akayama, Marion Brandsteidl, Birgit Demuth, Kenji Hisazumi, Timothy C Lethbridge, Perdita Stevens, and Dave R. Stikkolorum. Tool use in software modelling education: state of the art and research directions. In *the Educators' Symposium co-located with ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013)*, 2013. (cited on pages 8, 63, 65, 104, 117 and 145).

[3] Vincent Aleven and Kenneth R Koedinger. Limitations of student control: Do students know when they need help? In *International Conference on Intelligent Tutoring Systems*, pages 292–303. Springer, 2000. (cited on page 170).

[4] Vincent Aleven, Elmar Stahl, Silke Schworm, Frank Fischer, and Raven Wallace. Help seeking and help design in interactive learning environments. *Review of educational research*, 73(3):277–320, 2003. (cited on page 170).

[5] Scott W. Ambler. Agile model driven development is good enough. *IEEE Software*, 20(5):71–72, 2003. (cited on pages 110 and 190).

[6] Helen Anckar. Providing automated feedback on software design for novice designers. BSc thesis, Goteborg University, 2015. (cited on page 153).

[7] John R Anderson, Frederick G Conrad, and Albert T Corbett. Skill acquisition and the lisp tutor. *Cognitive Science*, 13(4):467–505, 1989. (cited on page 170).

[8] Cynthia Andres and Kent Beck. Extreme programming explained: Embrace change. *Reading: Addison-Wesley Professional*, 2004. (cited on page 110).

[9] Nikolaos Avouris, Vassilis Komis, Georgios Fiotakis, Meletis Margaritis, and Eleni Voyiatzaki. Logging of fingertip actions is not enough for analysis of learning activities. In *12th Int. Conf. AI Edu. (AIED 05) Workshop 1: Usage analysis in learning systems*, pages 1–8, 2005. (cited on page 79).

[10] Nilufar Baghaei and Antonija Mitrovic. COLLECT-UML : Supporting Individual and Collaborative Learning of UML Class Diagrams in a Constraint-Based Intelligent Tutoring System. *n R. Khosla, R. Hewlett & L. Jain (Eds.) Proc. KES*, pages 458–464, 2005. (cited on page 169).

[11] Nilufar Baghaei, Antonija Mitrovic, and Warwick Irwin. Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. *International Journal of Computer-Supported Collaborative Learning*, 2(2-3):159–190, 2007. (cited on page 66).

[12] Lut Baten. The c-test revisited a freeware package for placing efl students of business english in the new bachelor-master structure. *BELL: Belgian Journal of English Language and Literatures*, 2:301–315, 2004. (cited on page 30).

[13] Mordechai Ben-Ari. Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1):45–74, 2001. (cited on page 108).

[14] Jens Bennedssen and Michael E. Caspersen. Abstraction ability as an indicator of success for learning computing science? In *Proceedings of the Fourth international Workshop on Computing Education Research*, ICER '08, pages 15–26, New York, NY, USA, 2008. ACM. (cited on page 27).

[15] Marc Berges, Andreas Mühling, and Peter Hubwieser. The gap between knowledge and ability. In *Proceedings of the 12th Koli Calling international conference on computing education research*, pages 126–134. ACM, 2012. (cited on page 44).

[16] Gustav Bergström. Using machine learning to evaluate the layout quality of uml class diagrams. M.sc. thesis, Chalmers and Göteborg University, Sweden, April 2021. supervisor: M.R.V. Chaudron. (cited on page 167).

[17] Weiyi Bian, Omar Alam, and Jörg Kienzle. Automated grading of class diagrams. In *Proceedings of the Educators' Symposium at the 22nd International Conference on Model Driven Engineering Languages and Systems*, 2019. (cited on page 153).

[18] Paulo Blikstein. Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proc. 1st int. conf. learning analytics and knowledge*, pages 110–116. ACM, 2011. (cited on page 79).

[19] Narasimha Bolloju and Felix S.K. Leung. Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM*, 49(7):108–112, 2006. (cited on page 110).

[20] Grady Booch. Object-oriented design. *ACM SIGAda Ada Letters*, 1(3):64–76, 1982. (cited on pages 4 and 7).

[21] Nadia Boudewijn. Automated grading of java assignments. Master's thesis, Utrecht University, The Netherlands, 2016. (cited on page 154).

[22] Martin M Broadwell. Teaching for learning (xvi). *The Gospel Guardian*, 20(41):1–3, 1969. (cited on page 214).

[23] PJ Burton and RE Bruhn. Using UML to facilitate the teaching of object-oriented systems analysis and design. *Journal of Computing Sciences in Colleges*, 19(3):278–290, 2004. (cited on page 41).

[24] Weiqin Chen, Roger Heggernes Pedersen, and Øystein Pettersen. CoLeMo: A collaborative learning environment for UML modelling. *Interactive Learning Environments*, 14(May 2014):233–249, 2006. (cited on pages 66 and 169).

[25] S R Chidamber and C F Kemerer. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6):476–493, 1994. (cited on pages 53 and 167).

[26] Jan Claes, Irene Vanderfeesten, Jakob Pinggera, HajoA. Reijers, Barbara Weber, and Geert Poels. A visual analysis of the process of process modeling. *Information Systems and e-Business Management*, 13(1):147–190, 2015. (cited on page 93).

[27] Jan Claes, Irene Vanderfeesten, Hajo a Reijers, Jakob Pinggera, Matthias Weidlich, Stefan Zugal, Dirk Fahland, Barbara Weber, and Jan Mendling. Tying Process Model Quality to the Modeling Process : The Impact of Structuring , Movement , and Speed Background on the Process of Process Modeling. *10th International Conference, BPM 2012, Tallinn, Estonia, September 3-6, 2012. Proceedings*, pages 1–16, 2012. (cited on page 79).

[28] J. Cohen. *Statistical power analysis for the behavioral sciences*. Erlbaum, 1988. (cited on page 32).

[29] Eric Crahen, Carl Alphonce, and Phil Ventura. QuickUML: a beginner's UML tool. In *OOPSLA '02 Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, OOPSLA '02, pages 62–63, NY, USA, 2002. ACM New York. (cited on page 66).

[30] Holger Danielsiek, Jan Vahrenhold, Peter Hubwieser, Johannes Krugel, Johannes Magenheim, Laura Ohrndorf, Daniel Ossenschmidt, and Niclas Schaper. Undergraduate teaching assistants in computer science: Teaching-related beliefs, tasks, and competences. In *IEEE Global Engineering Education Conference, EDUCON*, pages 718–725, 2017. (cited on pages 128 and 148).

[31] Oswald de Bruin. The art of software design, creating an educational game teaching software design. Master's thesis, Leiden University, 2012. (cited on page 34).

[32] Birgit Demuth and Dave R. Stikkolorum, editors. *Proceedings of the MODELS Educators Symposium co-located with the ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), Valencia, Spain, September 29, 2014*, volume 1346 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014. No citations.

[33] John Dewey. *Experience and education*. Simon and Schuster, 2007. (cited on page 44).

[34] Serkan Dinçer and Ahmet Doğanay. The effects of multiple-pedagogical agents on learners' academic success, motivation, and cognitive load. *Computers and Education*, 111:74–100, 2017. (cited on page 172).

[35] Brian Dobing and Jeffrey Parsons. Dimensions of uml diagram use: a survey of practitioners. *Journal of Database Management (JDM)*, 19(1):1–18, 2008. (cited on page 6).

[36] João M Fernandes and Sónia M Sousa. Playscrum-a card game to learn the scrum agile method. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on*, pages 52–59. IEEE, 2010. (cited on page 191).

[37] Kiko Fernandez-Reyes, Dave Clarke, and Janina Hornbach. The impact of opt-in gamification on students' grades in a software design course. *21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS-Companion 2018*, 3:90–97, 2018. (cited on page 44).

[38] L Dee Fink. *Creating significant learning experiences: An integrated approach to designing college courses*. John Wiley & Sons, 2013. (cited on page 10).

[39] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415, 2014. (cited on page 146).

[40] Fabrice Gibert-Darras, Elisabeth Delozanne, Françoise Le Calvez, Agathe Merceron, Jean-Marc Labat, and Fabrice Vandebrouck. Towards a design pattern language to track students' problem-solving abilities. In *12th Int. Conf. AI Edu. (AIED 05) Workshop 1: Usage analysis in learning systems*, page 33, 2005. (cited on page 79).

[41] Jos Groenewegen, Stijn Hoppenbrouwers, and Erik Proper. Playing archimate models. In *Enterprise, Business-Process and Information Systems Modeling*, volume 50 of *Lecture Notes in Business Information Processing*, pages 182–194. Springer Berlin Heidelberg, 2010. (cited on page 43).

[42] Agneta Gulz and Magnus Haake. Design of animated pedagogical agents - A look at their look. *International Journal of Human Computer Studies*, 64(4):322–339, 2006. (cited on page 172).

[43] Magnus Haake and Agneta Gulz. A look at the roles of look & roles in embodied pedagogical agents - A user preference perspectivenet. *International Journal of Artificial Intelligence in Education*, 19(1):39–71, 2009. (cited on page 172).

[44] Patricia L. Hardré. Instructional design as a professional development tool-of-choice for graduate teaching assistants. *Innovative Higher Education*, 30(3):163–175, 2005. (cited on page 128).

[45] Patricia L. Hardré and Alicia O. Burris. What contributes to teaching assistant development: Differential responses to key design features. *Instructional Science*, 40(1):93–118, 2012. (cited on page 128).

[46] R Harrison, S J Counsell, and R V Nithi. An investigation into the applicability and validity of object-oriented design metrics. *Empirical Software Engineering*, 3(3):255–273, 1998. (cited on pages 60 and 167).

[47] Robert W Hasker. UMLGrader: an automated class diagram grader. *Journal of Computing Sciences in Colleges*, pages 47–54, 2011. (cited on page 153).

[48] Robert W Hasker and Mike Rowe. UMLint: Identifying defects in uml diagrams. In *American Society for Engineering Education*. American Society for Engineering Education, 2011. (cited on page 153).

[49] J. Hattie and H. Timperley. The Power of Feedback. *Review of Educational Research*, 77(1):81–112, mar 2007. (cited on pages 59, 146, 171, 176 and 181).

[50] Orit Hazzan and Jeff Kramer. Assessing abstraction skills. *Communications of the ACM*, 59(12):43–45, 2016. (cited on pages 12 and 108).

[51] Peter B. Henderson. Mathematical reasoning in software engineering education. *Commun. ACM*, 46(9):45–50, September 2003. (cited on page 27).

[52] Peter B. Henderson. Math counts: Mathematical reasoning in computing education. *ACM Inroads*, 1(3):22–23, September 2011. (cited on page 27).

[53] Peter B. Henderson. Mathematical reasoning in computing education ii. *ACM Inroads*, 2(1):23–24, February 2011. (cited on page 27).

[54] Alan Hevner and Samir Chatterjee. *Design Science Research in Information Systems*, pages 9–22. Springer US, Boston, MA, 2010. (cited on page 173).

[55] Shuhei Hiya, Kenji Hisazumi, Akira Fukuda, and Tsuneo Nakanishi. clooca: Web based tool for domain specific modeling. In *Demos/Posters/StudentResearch@ MoDELS*, pages 31–35, 2013. (cited on page 67).

[56] Ting-Chia Hsu, Shao-Chen Chang, and Yu-Ting Hung. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126:296–310, 2018. (cited on page 12).

[57] Jeff Irvine. A framework for comparing theories related to motivation in education. *Research in Higher Education Journal*, 35, 2018. (cited on page 10).

[58] Ivar Jacobson. *Object-oriented software engineering: a use case driven approach*. Pearson Education India, 1993. (cited on page 4).

[59] W Lewis Johnson and Jeff W Rickel. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11:47–78, 2000. (cited on page 66).

[60] Rodi Jolak, Boban Vesin, Marcus Isaksson, and Michel R.V. Chaudron. Towards a new generation of software design environments: Supporting the use of informal and formal notations with octouml. In *Second International Workshop on Human Factors in Modeling (HuFaMo 2016). CEUR-WS*, pages 3–10, 2016. (cited on page 67).

[61] Goda Jusaite, Pim Sanders, Damani Lawson, Koen van Polanen, Hani Al-Ers, and Dave R. Stikkolorum. Improving the quality of online collaborative learning for software engineering students. In *International Academic Conference on Teaching, Learning and E-learning*, pages 8–15, 2020. No citations.

[62] Hermann Kaindl. Difficulties in the transition from OO analysis to design. *IEEE Software*, 16(5):94–102, 1999. (cited on pages 12 and 190).

[63] Slava Kalyuga. Enhancing instructional efficiency of interactive e-learning environments: A cognitive load perspective. *Educational Psychology Review*, 19(3):387–399, 2007. (cited on page 171).

[64] Bilal Karasneh, Dave R. Stikkolorum, Enrique Larios, and Michel R.V. Chaudron. Quality assessment of UML class diagrams. In *Proc. Educators' Symp at MoDELS*, 2015. No citations.

[65] Barbara a. Kitchenham, Shari L Pfleeger, Lesley M Pickard, Peter W Jones, David C Hoaglin, Khaled El Emam, and Jarrett Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–734, 2002. (cited on page 14).

[66] Andrew J Ko, Brad A Myers, Michael J Coblenz, and Htet Htet Aung. An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *Software Engineering, IEEE Transactions on*, 32(12):971–987, 2006. (cited on page 79).

[67] David A Kolb. *Experiential learning: Experience as the source of learning and development*. FT Press, 2014. (cited on pages 78 and 92).

[68] Michael Kölling. Using bluej to introduce programming. In *Reflections on the Teaching of Programming*, pages 98–115. Springer, 2008. (cited on page 44).

[69] Jeff Kramer. Is abstraction the key to computing? *Communications of the ACM*, 50(4):36–42, April 2007. (cited on pages 4, 12, 26, 40, 92, 108 and 166).

[70] David R. Krathwohl. A revision of bloom's taxonomy: An overview. *Theory Into Practice*, 41(4):212–218, 2002. (cited on pages 10 and 44).

[71] A. Krivitsky. A multi-team, full-cycle, product-oriented scrum simulation with lego bricks – the small & medium business edition v2.0., jun October 2011. (cited on page 190).

[72] Han Lai and Wenjuan Xin. An experimental research of the pair programming in java programming course. *Proceeding of the International Conference on eEducation Entertainment and eManagement*, pages 257–260, 2011. (cited on pages 108, 110 and 117).

[73] Craig Larman. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development*. Pearson Education India, 2012. (cited on page 7).

[74] Michael J. Lee and Andrew J. Ko. Personifying programming tool feedback improves novice programmers' learning. In *Proceedings of the seventh international workshop on Computing education research - ICER '11*, page 109, New York, New York, USA, 2011. ACM Press. (cited on page 43).

[75] Michael J. Lee and Andrew J. Ko. A demonstration of gidget, a debugging game for computing education. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, pages 211–212, 2014. (cited on pages 43 and 59).

[76] Lego4scrum. `https://www.lego4scrum.com`. [Online; accessed 20-March-2016]. (cited on pages 189, 190 and 219).

[77] T.C. Lethbridge. What knowledge is important to a software professional? *Computer*, 33(5):44–50, 2000. (cited on page 27).

[78] Timothy C Lethbridge. Teaching modeling using Umple: Principles for the development of an effective tool. *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*, pages 23–28, 2014. (cited on page 67).

[79] F. Leung and N. Bolloju. Analyzing the Quality of Domain Models Developed by Novice Systems Analysts. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 00(C):188b–188b, 2005. (cited on pages 108, 110 and 127).

[80] Felix Leung and Narasimha Bolloju. Analyzing the quality of domain models developed by novice systems analysts. In *System Sciences, 2005. HICSS'05. Proc. of the 38th Annual Hawaii Int. Conf. on*, pages 188b–188b. IEEE, 2005. (cited on pages 40, 92, 93 and 166).

[81] C. Lewis. Using the "thinking-aloud" method in cognitive interface design. Technical report, IBM T.J. Watson Research Center, 1982. (cited on pages 15 and 45).

[82] Clayton Lewis and John Rieman. Task-centered user interface design. *A Practical Introductio*, 1993. (cited on page 94).

[83] Grischa Liebel, Rogardt Heldal, and Jan Philipp Steghofer. Impact of the use of industrial modelling tools on modelling education. *Proceedings - 2016 IEEE 29th Conference on Software Engineering Education and Training, CSEEandT 2016*, pages 18–27, 2016. (cited on pages 8, 118 and 145).

[84] Zhiyi Ma, Chih-yi Yeh, Huihong He, and Hongjie Chen. A Web Based UML Modeling Tool with Touch Screens. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, pages 835–838, 2014. (cited on page 67).

[85] Ntima Mabanza and Lizette de Wet. Determining the usability effect of pedagogical interface agents on adult computer literacy training. In *E-Learning Paradigms and Applications*, pages 145–183. Springer, 2014. (cited on page 71).

[86] Philip Machanick. A social construction approach to computer science education. *Computer Science Education*, 17(1):1–20, 2007. (cited on page 11).

[87] Maíra R Marques, Alcides Quispe, and Sergio F Ochoa. A systematic mapping study on practical approaches to teaching software engineering. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–8. IEEE, 2014. (cited on pages 11 and 40).

[88] Robert C Martin. Design Principles and Design Patterns. *Object Mentor*, pages 1–34, 2000. (cited on pages 7, 29 and 44).

[89] Ian McChesney. Three Years of Student Pair Programming – Action Research Insights and Outcomes. *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16)*, pages 84–89, 2016. (cited on pages 108, 110 and 117).

[90] Sabine Moisan and Jean Paul Rigault. Teaching object-oriented modeling and UML to various audiences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6002 LNCS:40–54, 2010. (cited on page 108).

[91] Anders I. Mørch, Silje Jondahl, and Jan A. Dolonen. Supporting conceptual awareness with pedagogical agents. *Information Systems Frontiers*, 7(1):39–53, 2005. (cited on page 172).

[92] Robert Moser. A fantasy adventure game as a learning environment. why learning to program is so difficult and what can be done about it. *ITiCSE 97 Proceedings of the 2nd conference on Integrating technology into computer science education*, 29(3):114–116, 1997. (cited on page 52).

[93] Madlen Müller-Wuttke and Nicholas H Müller. Cognitive load levels while learning with or without a pedagogical agent. In *International Conference on Human-Computer Interaction*, pages 266–276. Springer, 2019. (cited on page 173).

[94] Gunter Mussbacher, Daniel Amyot, Ruth Breu, Jean-Michel Bruel, Betty HC Cheng, Philippe Collet, Benoit Combemale, Robert B France, Rogardt Heldal, James Hill, et al. The relevance of model-driven engineering thirty years from now. In *Model-Driven Engineering Languages and Systems*, pages 183–200. Springer, 2014. (cited on pages 3, 13 and 188).

[95] Valbona Muzaka. The niche of graduate teaching assistants (GTAs): Perceptions and reflections. *Teaching in Higher Education*, 14(1):1–12, 2009. (cited on page 126).

[96] Sharon Nelson-Le Gall. Help-seeking: An understudied problem-solving skill in children. *Developmental review*, 1(3):224–246, 1981. (cited on page 170).

[97] Hafeez Osman, Arjan van Zadelhoff, Dave R. Stikkolorum, and Michel R.V. Chaudron. UML class diagram simplification: what is in the developer's mind? In *Proceedings of the Second Edition of the International Workshop on Experiences and Empirical Studies in Software Modelling*, page 5. ACM, 2012. No citations.

[98] Maria Paasivaara, Ville Heikkilä, Casper Lassenius, and Towo Toivola. Teaching students scrum using lego blocks. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 382–391. ACM, 2014. (cited on page 190).

[99] Chris Park. The graduate teaching assistant (GTA): lessons from North American experience. *Teaching in Higher Education*, 9(3):349–361, 2004. (cited on page 127).

[100] Óscar Pastor, Sergio España, and Jose Ignacio Panach. Learning Pros and Cons of Model-Driven Development in a Practical Teaching Experience. In *International Conference on Conceptual Modeling*, pages 218–227. Springer, 2016. (cited on page 110).

[101] Elizabeth Ann Patitsas. A case study of the development of CS teaching assistants and their experiences with team teaching. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research - Koli Calling '13*, pages 115–124, 2013. (cited on page 128).

[102] Dewayne E Perry, Adam a Porter, and Lawrence G Votta. Empirical Studies of Software Engineering : A Roadmap. *Proceedings of the conference on The future of Software engineering ICSE 00*, pages 345–355, 2000. (cited on page 14).

[103] Jakob Pinggera, Pnina Soffer, Stefan Zugal, Barbara Weber, Matthias Weidlich, Dirk Fahland, Hajo a. Reijers, and Jan Mendling. Modeling styles in business process modeling. *Lecture Notes in Business Information Processing*, 113 LNBIP:151–166, 2012. (cited on page 79).

[104] M Prensky. *Digital game-based learning*. McGraw-Hill & Paragon House, New York, 2001. (cited on page 42).

[105] Marc Prensky. Digital Game-Based Learning. *Computers in Entertainment (CIE)*, 1(1):1–4, 2003. (cited on page 42).

[106] Andrew K Przybylski, C Scott Rigby, and Richard M Ryan. A motivational model of video game engagement. *Review of General Psychology*, 14(2):154–166, 2010. (cited on pages 52, 59 and 218).

[107] Truong Ho Quang. LogViz - visual log analyzer. `https://gitlab.com/truonghoquang/LogVisualizer`, 2015. [Online; accessed 8-March-2015]. (cited on page 71).

[108] Ebrahim Rahimi and Dave R. Stikkolorum, editors. *CSERC '19: Proceedings of the 8th Computer Science Education Research Conference*, New York, NY, USA, 2019. Association for Computing Machinery. No citations.

[109] Ervin Ramollari and Dimitris Dranidis. StudentUML: An educational tool supporting object-oriented analysis and design. *... of the 11th Panhellenic Conference on ...*, pages 363–373, 2007. (cited on page 66).

[110] John Raven. The raven's progressive matrices: change and stability over culture and time. *Cognitive psychology*, 41(1):1–48, 2000. (cited on page 31).

[111] Maryam Razavian, Antony Tang, Rafael Capilla, and Patricia Lago. In two minds: how reflections influence software design thinking. *Journal of Software: Evolution and Process*, 28(6):394–426, 2016. (cited on page 110).

[112] Arthur J. Riel. *Object-Oriented Design Heuristics*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1996. (cited on page 29).

[113] Patricia Roberts. Abstract thinking: a predictor of modelling ability? In *Educators Symposium of the ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems*, pages 753–754. Springer, 2009. (cited on pages 27, 40, 92 and 166).

[114] Anthony Robins, Janet Rountree, and Nathan Rountree. Learning and teaching programming: A review and discussion. *Computer science education*, 13(2):137–172, 2003. (cited on page 44).

[115] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William E. Lorensen, et al. *Object-oriented modeling and design*, volume 199. Prentice-hall Englewood Cliffs, NJ, 1991. (cited on page 4).

[116] Bernhard Rumpe. Agile Modeling with the UML 1 Portfolio of Software Engineering Techniques. *Radical Innovations of Software and Systems Engineering in the Future. 9th International Workshop*, pages 297–309, 2004. (cited on pages 110 and 190).

[117] Bernhard Rumpe. *Agile Modellierung mit UML: Codegenerierung, Testfälle, Refactoring*. Springer-Verlag, 2012. (cited on page 190).

[118] Adrian Rusu, Robert Russell, Remo Cocco, and Spence DiNicolantonio. Introducing object oriented design patterns through a puzzle-based serious computer game. In *2011 Frontiers in Education Conference (FIE)*, pages F1H–1. IEEE, 2011. (cited on page 43).

[119] Andreas Schroeder, Annabelle Klarl, Philip Mayer, and Christian Kroiss. Teaching agile software development through lab courses. *IEEE Global Engineering Education Conference, EDUCON*, pages 1177–1186, 2012. (cited on page 190).

[120] Noah L Schroeder. The influence of a pedagogical agent on learners' cognitive load. *Journal of Educational Technology & Society*, 20(4):138–147, 2017. (cited on pages 172 and 173).

[121] Swapneel Sheth, Jonathan Bell, and Gail Kaiser. A competitive-collaborative approach for introducing software engineering in a cs2 class. In *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 41–50. IEEE, 2013. (cited on page 42).

[122] Shin-shing Shin. A Study on the Difficulties of Learning Phase Transition in Object-Oriented Analysis and Design From the Viewpoint of Semantic Distance. *IEEE*, 58(2):1–7, 2014. (cited on pages 12, 108, 121 and 190).

[123] Ricardo Azambuja Silveira, Eduardo Rodrigues Gomes, and Rosa Maria Viccari. Intelligent learning objects: An agent approach to create reusable intelligent learning environments with learning objects. *Advances in Artificial Intelligence - Iberamia-Sbia 2006, Proceedings*, pages pp. 17–26, 2006. (cited on page 71).

[124] J. Soler, I. Boada, F. Prados, J. Poch, and R. Fabregat. A web-based e-learning tool for UML class diagrams. *2010 IEEE Education Engineering Conference, EDUCON 2010*, pages 973–979, 2010. (cited on page 67).

[125] Minseok Song and W.M.P. van der Aalst. Supporting process mining by showing events at a glance. *In Proceedings of 17th Annual Workshop on Information Technologies and Systems*, pages 139–147, 2007. (cited on page 93).

[126] Mauricio Ronny De Almeida Souza, Lucas Furtini Veado, Renata Teles Moreira, Eduardo Magno Lages Figueiredo, and Heitor Augustus Xavier Costa. Games for learning: bridging game-related education methods to software engineering knowledge areas. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, pages 170–179. IEEE, 2017. (cited on pages 42 and 44).

[127] Shashank Srikant and Varun Aggarwal. A system to grade computer programming skills using machine learning. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 1887–1896, 2014. (cited on page 153).

[128] Thomas Staubitz, Hauke Klement, Jan Renz, Ralf Teusner, and Christoph Meinel. Towards practical programming exercises and automated assessment in massive open online courses. In *Teaching, Assessment, and Learning for Engineering (TALE), 2015 IEEE International Conference on*, pages 23–30. IEEE, 2015. (cited on page 44).

[129] Christoph Johann Stettina and Werner Heijstek. Necessary and neglected?: An empirical study of internal documentation in agile software development teams. In *Proceedings of the 29th ACM International Conference on Design of Communication*, SIGDOC '11, pages 159–166, New York, NY, USA, 2011. ACM. (cited on page 188).

[130] Dave R. Stikkolorum. Online Experiment with WebUML : recording difficulties and strategies of students. Technical report, Leiden University, LIACS, 07 2015. (cited on page 94).

[131] Dave R. Stikkolorum and Michel R.V. Chaudron. A workshop for integrating UML modelling and agile development in the classroom. In *Proceedings of the*

*Computer Science Education Research Conference 2016*, pages 4–11. ACM, 2016. (cited on pages 109, 110 and 116).

[132] Dave R. Stikkolorum and Michel R.V. Chaudron. Teaching of agile uml modelling: Recommendations from students' reflections. In *Proceedings of the 20th Ibero-American Conference on Software Engineering, Buenos Aires, Argentina*, 2017. (cited on pages 127, 145 and 147).

[133] Dave R. Stikkolorum, Michel R.V. Chaudron, and Oswald de Bruin. The art of software design, a video game for learning software design principles. In *Gamification Contest MODELS'12 Innsbruck*, 2012. (cited on page 34).

[134] Dave R. Stikkolorum, F. Gomes de Oliveira Neto, and Michel R.V. Chaudron. Evaluating didactic approaches used by teaching assistants for software analysis and design using uml. In *Proceedings of the 3rd European Conference of Software Engineering Education*, ECSEE'18, pages 122–131, New York, NY, USA, 2018. ACM. No citations.

[135] Dave R. Stikkolorum, Birgit Demuth, Vadim Zaytsev, Frédéric Boulanger, and Jeff Gray. The MOOC hype: Can we ignore it? In *Reflections on the current use of massive open online courses in software modeling education. MODELS Educators Symposium, EduSymp*, 2014. No citations.

[136] Dave R. Stikkolorum, Truong Ho-Quang, and Michel R.V. Chaudron. Revealing students' UML class diagram modelling strategies with WebUML and LogViz. In *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on*, pages 275–279. IEEE, 2015. (cited on pages 71, 97, 153, 155 and 189).

[137] Dave R. Stikkolorum, Truong Ho-Quang, Bilal Karashneh, and Michel R.V. Chaudron. Uncovering students' common difficulties and strategies during a class diagram design process: an online experiment. In *Educators Symposium 2015, co-located with the ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems*, 2015. (cited on pages 71, 108, 109, 174 and 189).

[138] Dave R. Stikkolorum and Ebrahim Rahimi, editors. *CSERC '20: Proceedings of the 9th Computer Science Education Research Conference*, New York, NY, USA, 2020. Association for Computing Machinery. No citations.

[139] Dave R. Stikkolorum and Ebrahim Rahimi, editors. *CSERC '21: Proceedings of the 10th Computer Science Education Research Conference*, New York, NY, USA, 2021. Association for Computing Machinery. No citations.

[140] Dave R. Stikkolorum and Ebrahim Rahimi, editors. *CSERC '22: Proceedings of the 11th Computer Science Education Research Conference*, New York, NY, USA, 2022. Association for Computing Machinery. No citations.

[141] Dave R. Stikkolorum, Claire Stevenson, and Michel R.V. Chaudron. Assessing software design skills and their relation with reasoning skills. In *EduSymp 2013. CEUR, vol. 1134, paper 5*, 2013. (cited on pages 40, 92, 108 and 166).

[142] Dave R. Stikkolorum, Claire E. Stevenson, and Michel R.V. Chaudron. Technical report 2013-02. https://research.drstikko.nl/files/technical_report_2013-02.pdf, 2013. (cited on page 31).

[143] Dave R. Stikkolorum, Peter van der Putten, Caroline Sperandio, and Michel R.V. Chaudron. Towards automated grading of uml class diagrams with machine learning. In *BNAIC/BENELEARN*, 2019. No citations.

[144] Harald Störrle. On the impact of layout quality to understanding UML diagrams: Diagram type and expertise. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, pages 49–56, 2012. (cited on page 94).

[145] Harald Störrle. On the impact of layout quality to understanding UML diagrams: Size matters. In *Model-Driven Engineering Languages and Systems*, pages 518–534. Springer, 2014. (cited on page 94).

[146] Michael Striewe and Michael Goedicke. Automated checks on uml diagrams. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ITiCSE '11, pages 38–42, New York, NY, USA, 2011. ACM. (cited on page 153).

[147] Michael Striewe and Michael Goedicke. Automated assessment of uml activity diagrams. In *ITiCSE*, page 336, 2014. (cited on page 153).

[148] Antony Tang. Software Designers , Are You Biased ? *Program*, pages 1–8, 2011. (cited on page 110).

[149] Jennifer Tenzer and Perdita Stevens. GUIDE: Games with UML for interactive design exploration. *Knowledge-Based Systems*, 20(7):652–670, 2007. (cited on page 44).

[150] Damiano Torre, Yvan Labiche, Marcela Genero, and Maged Elaasar. A systematic identification of consistency rules for uml diagrams. *Journal of Systems and Software*, 144:121–142, 2018. (cited on page 167).

[151] Kalliopi Tourtoglou and Maria Virvou. User modelling in a collaborative learning environment for uml. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 1257–1258. IEEE, 2008. (cited on page 169).

[152] Ian Utting, Stephen Cooper, Michael Kölling, John Maloney, and Mitchel Resnick. Alice, greenfoot, and scratch–a discussion. *ACM Transactions on Computing Education (TOCE)*, 10(4):17, 2010. (cited on page 43).

[153] Marcella Veldthuis, Matthijs Koning, and Dave R. Stikkolorum. A quest to engage computer science students:using dungeons & dragons for developing soft skills. In *Proceedings of the Computer Science Education Research Conference 2021*. ACM, 2021. No citations.

[154] JDHM Vermunt. *Leerstijlen en sturen van leerprocessen in het hoger onderwijs - Naar procesgerichte instructie in zelfstandig denken*. Amsterdam: Swets & Zeitlinger, 1992. (cited on pages 78 and 92).

[155] Boban Vesin, Aleksandra Klašnja-Milićević, Katerina Mangaroska, Mirjana Ivanović, Rodi Jolak, Dave Stikkolorum, and Michel Chaudron. Web-based educational ecosystem for automatization of teaching process and assessment of students. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pages 1–9, 2018. (cited on page 223).

[156] Arto Vihavainen, Matti Luukkainen, and Jaakko Kurhila. Multi-faceted support for mooc in programming. In *Proceedings of the 13th annual conference on Information technology education*, pages 171–176. ACM, 2012. (cited on page 44).

[157] Visual paradigm. https://www.visual-paradigm.com. [Online; accessed 20-November-2022]. (cited on page 192).

[158] Christiane Gresse von Wangenheim, Rafael Savi, and Adriano Ferreti Borgatto. Scrumia—an educational game for teaching scrum in computing courses. *Journal of Systems and Software*, 86(10):2675–2687, 2013. (cited on page 191).

[159] Lorin W Anderson, David R Krathwohl, Peter W Airasian, Kathleen A Cruikshank, Richard E Mayer, Paul R Pintrich, James Raths, and Merlin C Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*. Allyn & Bacon, 2000. (cited on page 30).

[160] Wim Westera, Rob Nadolski, and Hans Hummel. Serious Gaming Analytics : What Students Log Files Tell Us about Gaming and Learning. *International Journal of Serious Games*, 1(2):35–50, 2014. (cited on page 79).

[161] Grant P Wiggins, Grant Wiggins, and Jay McTighe. *Understanding by design*. Ascd, 2005. (cited on page 10).

[162] Yuefeng Zhang. Agile Development in Practice. *IEEE Software - IEEE Computer Society*, 2011. (cited on pages 110 and 190).

[163] Chen Zhuoyi, Li Na, and Zhang Hongjie. Exploration of teaching model of the database course based on constructivism learning theory. In *Consumer Electronics, Comm. and Networks (CECNet), 2nd Int. Conf. on*, pages 1808–1811. IEEE, 2012. (cited on page 110).

# Acknowledgements

Over the past years many people and organisations supported my research. I would like to take the opportunity to thank them. Without their support I could not have finished this journey.

*To the following organisations*: Leiden University (LIACS and Dual PhD Centre), The Hague University of Applied Sciences. Thank you for the opportunity and for trusting me to complete this research.

*To the following collaborators*: I would like to thank the students and lecturers from Chalmers University of Technology and Göteborg University in Sweden and Utrecht University in The Netherlands for their participation in the study of Chapter 2. I would like to thank the lecturers and students of the Software Engineering program of Makerere University in Uganda for their collaboration in the experiment of Chapter 6. I would like to thank the students and lecturers from The Hague University of Applied Sciences in The Netherlands for their participation and helpful discussions in the research of Chapter 7. I would like to thank the students of Chalmers University of Technology and Göteborg University, for evaluating the feedback agent of Chapter 9. I would like to thank all the students from Chalmers University of Technology and Göteborg University that participated in this experiment for their enthusiasm and constructive contribution to Chapter 11. I would like to thank the following students (they were students at the time of conducting the research): Claire Stevenson for collaborating with me in the research of Chapter 2, Oswald de Bruin for the collaboration in Chapter 3, Caroline Sperandio for the collaboration in Chapter 9 and Helen Anckar for the collaboration in Chapter 10.

*To my critical friends:* Grischa Liebel, Jan-Philipp Steghöfer, Marcella Veldthuis, Felienne Hermans, Tony Andrioli, Tim Cocx and Vreda Pieterse. Thank you for your

constructive critics on my research approaches and the writing of this dissertation.

*To my colleagues from LIACS:* Werner Heijstek, Ana Fernandez, Hafeez Osman, Bilal Karasneh, Christoph Stettina, Ali Mirsoleimani, Ben Ruijl, Ariadi Nugroho and Bob Zadok. Thank you for all the nice talks and discussions, good advises and sharing of insights of your own research.

*To my colleagues from Chalmers University of Technology and Göteborg University:* Håkan Burden, Rogardt Heldal, Truong Ho Quang, Rodi Jolak, Pariya Kashfi, Regina Hebig, Hiva Alahyari, Boban Vesin and Francisco Gomes. Thank you for sharing your insights, the treats during *fika* and the nice after work moments.

*To my colleagues from The Hague University:* I would like to thank you all (the group is too large to name everyone individually) for supporting me all these years in combining my teaching and managing role with my PhD research.

*To my family:* I would like to thank and express my love to all my family members and close friends, but in particular my wife *Maria José* and my kids *Jay* and *Julian* for their belief and patience. I could not have done this without you.