



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).

Samenvatting

Softwareontwerp is een uitdagende taak voor studenten. Door de toename van steeds complexer wordende systemen wordt ook het aanleren van softwareontwerp een uitdaging. Het is algemeen bekend dat studenten worstelen met het aanleren van de (abstracte) vaardigheden van softwareontwerp, daarom zijn nieuwe onderwijsinterventies nodig.

Dit proefschrift onderzoekt verschillende manieren waarop we onze huidige aanpak in het onderwijzen van softwareontwerp kunnen verbeteren en uitbreiden. Het hoofddoel van dit onderzoek is:

Hoofddoel *Hoe kunnen we de manier waarop we softwareontwerp aanleren verbeteren?*

We hebben verschillende aspecten van het hoofddoel bestudeerd: het meten van softwareontwerpvaardigheden; de begeleiding en de feedback op ontwerptaken; en de motivatie van studenten voor - en de betrokkenheid bij softwareontwerpactiviteiten.

We onderzochten interventies door gebruik te maken van tools die we ontwikkeld hebben voor het assen van ontwerpvaardigheden, modelleren van software en het begeleiden en beoordelen van het ontwerpproces. Centraal in ons onderzoek stonden softwareontwerpprincipes. Deze ontwerpprincipes vormen de basis van de complexe afweging van meerdere tegenstrijdige doelen die goede softwareontwerpers uiteindelijk zouden moeten kunnen maken.

Hieronder beschrijven we onze verschillende interventies en ontwikkelde tools.

We hebben een online meerkeuzetoets ontworpen, een instrument voor het meten van softwareontwerpvaardigheden en het vermogen tot abstract redeneren van studenten. Ons onderzoek toonde de relatie aan tussen abstract redeneren en het vermogen om

softwareontwerpproblemen op te lossen. Wij zien hiermee kansen voor onderzoek naar onderwijsinterventies voor specifieke (abstracte) redeneertaken en/of alternatieve werkvormen. Docenten kunnen onze test gebruiken voor het diagnosticeren van studenten en daarmee passende interventies kiezen.

Om studenten te motiveren voor het leren van softwareontwerp, en om ze betrokken te houden, hebben we onderzoek gedaan naar een educatieve puzzel-videogame. Door in-game te oefenen met het nemen van ontwerpbeslissingen leren de spelers hoe ze de balans kunnen vinden tussen de verschillende doelen van een softwareontwerp. Een belangrijk instrument in de game is het visuele feedbacksysteem dat studenten naar hun oplossing toe begeleidt. Met het ingebouwde feedbackmechanisme zien we de game als mogelijk middel voor zelfevaluatie van studenten. De interactieve game houdt ze betrokken.

Vanwege de behoefte aan een UML-tool die zowel als educatieve tool als onderzoekstool kan worden gebruikt, hebben we WebUML ontwikkeld. De web-based architectuur in combinatie met een logsysteem maakt het mogelijk om onderzoek met statistische analyses te doen bij grote groepen. We hebben data verzameld over de ontwerpactiviteiten van studenten tijdens het analyseren en ontwerpen van class diagrams. Onze tools creëren nieuwe manieren om het softwareontwerpproces te bestuderen. De analyse van het modelleergedrag van studenten helpt ons betere educatieve programma's en tools te ontwikkelen. Het herkennen van bepaalde strategieën van studenten bij het maken van ontwerpen kan ons helpen om betere feedback te geven tijdens een ontwerptaak in plaats van ná uitvoering van hun taak. Door de logs van de studenten die de WebUML-tool gebruiken bij het maken van ontwerpen te analyseren, ontdekten we typische strategieën die studenten gebruiken om class designs te maken: Depth Less, Depth First, Breadth First, en Ad Hoc. De strategieën verschillen in waar de student zich eerst op richt, op de high-level class structuur of op de individuele class details.

Een andere onderwijsinterventie die we onderzochten was het doen van peer-reflecties door studenten tijdens een analyse- en ontwerpopdracht. De peer-reflecties legden de problemen waarmee de studenten worden geconfronteerd bloot. We identificeerden veelvoorkomende problemen en presenteerden 10 concrete aandachtspunten voor het aanleren van softwareontwerp. Uit de vastgelegde peer-reflecties concluderen we dat peer-reflectie studenten helpt om hun moeilijkheden bij het modelleren van softwareontwerpen uit te drukken. De studenten bespraken actief de moeilijkheden die ze ondervonden tijdens hun taak. Ze bespraken zowel de voortgang als de kwaliteit-aspecten van hun modellen. Met de gepresenteerde aandachtspunten breiden we de kennis uit over veelvoorkomende problemen bij het aanleren van softwareontwerp door studenten.

Voor een andere benadering om problemen van studenten tijdens het maken van een softwareontwerp bloot te leggen, observeerden we de interactie tussen onderwijsassistenten (OAs) en studenten tijdens een analyse- en ontwerpcurcus. We hebben de gemeenschappelijke problemen waarbij studenten de hulp van OAs hebben gevraagd gecategoriseerd. Verder hebben we een overzicht van de typische interventies die OAs gebruiken bij het ondersteunen van studenten tijdens het uitvoeren van hun opdrachten.

We onderzochten de toepassing van geautomatiseerde feedback en beoordeling van UML class diagrams om studenten in staat te stellen vaker te oefenen en na te denken over hun eigen voortgang door het gebruik van zelfevaluatie. Naast de evaluatiefunctie, maakt de feedback het mogelijk studenten te begeleiden naar hun oplossing. De feedback gaat over: het gebruik van de juiste semantiek, zoals de juiste vertaling van een relatie tussen concepten en het voldoen aan requirements, zoals het identificeren van relevante concepten binnen een gegeven context.

We hebben de data van softwareontwerpen van bachelorstudenten verzameld. Op deze data hebben we machine learning toegepast met als doel deze opdrachten automatisch te kunnen beoordelen. Op basis van onze resultaten concluderen we dat het gebruik van machine learning voor een classificatie die dezelfde precisie gebruikt als een 10-punts beoordelingsschaal³ niet nauwkeurig is. Het huidige voorspellingsmodel geeft gebruikers echter wel een ruwe indicatie van de kwaliteit van hun modellen.

We hebben onze pedagogische feedbackagent gepresenteerd als uitbreiding op WebUML. Deze agent geeft feedback aan studenten terwijl ze leren om UML class designs te maken. De studenten waarderen de begeleidingsrol van de agent: de manier waarop wordt bevestigd dat stappen op de juiste manier zijn uitgevoerd en de suggesties die worden gegeven voor vervolgstappen richting de oplossing.

Naast onze tooling hebben we een workshop ontwikkeld waarin UML-modelleren integreert is in het agile softwareontwikkelingsproces. Een geïntegreerde benadering van softwareontwikkeling helpt studenten de rol van (UML)-modelleren in een softwareontwikkelingscyclus te begrijpen. Daarnaast hebben we tijdens groepsgesprekken over de modelleertaak opvallende problemen van studenten verzameld. Studenten zijn positief over de workshop. De studenten zijn van mening dat ze hun modelleervaardigheden hebben verbeterd en dat de workshop inzicht heeft gegeven in de relatie tussen modelleren en het softwareontwikkelingsproces.

Met de bovengenoemde onderzoeken hebben we het repertoire aan onderwijsinterventies en tools voor het aanleren van softwareontwerp uitgebreid. Met de juiste feedback begeleiden deze interventies en tools de studenten tijdens hun ontwerptaken. We

³Dit is de conventionele schaal voor het beoordelen van toetsen in Nederland.

nodigen andere docenten uit om onze aanpak te gebruiken en voort te bouwen op onze inzichten en hiermee toekomstige interactieve leeromgevingen te verrijken.