

## PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a preprint version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/75596>

Please be advised that this information was generated on 2019-06-27 and may be subject to change.

# Een Module over Model Checking voor het VWO

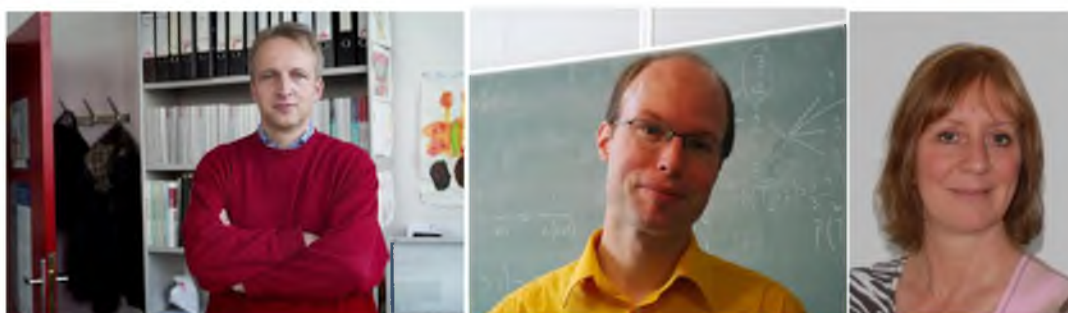
Frits Vaandrager<sup>1</sup>, David N. Jansen<sup>1</sup> en Els Koopmans<sup>2</sup>

<sup>1</sup>Radboud Universiteit Nijmegen

<sup>2</sup>Olympus College, Arnhem

**Samenvatting.** We beschrijven achtergrond, leerdoelen en opzet van een lesmodule over model checking voor klas 5 vwo, die dit jaar voor de derde keer verzorgd wordt in een samenwerking tussen de Radboud Universiteit Nijmegen en het Olympus College te Arnhem. Tevens doen we verslag van onze ervaringen bij het geven van deze module.

Trefwoorden: lesmodule over model checking, samenwerking VO-WO.



## Inleiding: Waarom deze module?

De Radboud Universiteit Nijmegen en het Olympus College participeren, samen met de HAN en een 30-tal andere middelbare scholen uit de regio Arnhem–Nijmegen, in het Sprint-UP project. Doelen van dit samenwerkingsproject zijn o.a.

- Inhoudelijke vernieuwing bèta-onderwijs.
- Leerlingen een aantrekkelijker en vollediger beeld geven van de wereld van bèta en techniek.
- Studenten en junioronderzoekers kennis laten maken met leraarschap en mogelijk interesseren voor een loopbaan in het onderwijs.

Een aanvulling specifiek in dit project voor informatica:

- Leerlingen kennis laten maken met een vervolgopleiding Informatica binnen het WO.

Sprint-UP wordt gesubsidieerd door het Platform Bèta Techniek. Binnen de Informatica bestaat er bij uitstek behoefte aan dit soort initiatieven. Docenten Informatica op middelbare scholen hebben vrijwel nooit Informatica gestudeerd (een wezenlijk verschil met de situatie bij andere bètavakken), het wetenschapsgebied van de Informatica verandert zeer snel, en de instroom/uitstroom van ICT-opleidingen bij het WO is structureel volstrekt onvoldoende gelet op de behoefte vanuit de samenleving. Wij

vermoeden dat één van de redenen waarom scholieren zo weinig voor ICT opleidingen kiezen is dat ze geen idee hebben van wat het wetenschapsgebied van de Informatica inhoudt en aan welke fascinerende fundamentele vragen er momenteel onderzoek wordt verricht. Al met al zijn er veel goede redenen voor een nauwere samenwerking tussen VO en WO op het gebied van de Informatica!

## Waarom model checking?

Bij het Informatica-onderwijs op middelbare scholen wordt relatief veel aandacht besteed aan programmeren. In de universitaire wereld en het bedrijfsleven is er echter sprake van een duidelijke trend waarbij *modellen* het voornaamste product van systeemontwikkeling worden, denk bijvoorbeeld aan de opkomst van de Unified Modeling Language (UML) en Model Driven Architecture (MDA). Hierbij worden systeemeisen, gedrag en functionaliteit allemaal beschreven in termen van (grafische) modellen. Idealiter worden deze modellen gebruikt voor de communicatie met klanten en belanghebbenden, het analyseren van het ontwerp, het genereren van code en het testen van de uiteindelijke implementatie. De beschikbaarheid van modellen vereenvoudigt tevens onderhoud en hergebruik. Wij menen dat het wenselijk is om in het middelbaar onderwijs meer aandacht te besteden aan deze ontwikkelingen, en in het bijzonder aan het *modelleren* van systemen (op een hoog niveau van abstractie) en het *analyseren* van modellen. Binnen de door ons ontwikkelde module gebruiken leerlingen het *model checking* pakket Uppaal voor het modelleren en analyseren van systemen (ontwikkeld door de universiteiten van Uppsala en Aalborg met bijdragen van de Radboud Universiteit Nijmegen, en gratis beschikbaar voor niet-commercieel gebruik). Binnen de universitaire wereld is er recent veel aandacht voor model checking. Zo is in 2007 de ACM Turing Award, vaak aangeduid als de Nobelprijs voor de Informatica, uitgereikt aan de onderzoekers Clarke, Emerson en Sifakis voor hun baanbrekend onderzoek op dit terrein. Model checking is een succesvolle technologie om aan te tonen dat een ontwerp van een computersysteem voldoet aan de eisen die er aan gesteld worden. Met behulp van efficiënte algoritmen doorloopt een model checker automatisch de toestandsruimte van een systeemmodel en bepaalt of het systeem voldoet aan zijn specificatie. De model checker Uppaal beschikt over een uitstekende grafische user interface en na een korte training van één les kunnen leerlingen er al mee uit de voeten. Ze hebben dan geen weet van de (prachtige) achterliggende theorie en algoritmen, maar dat hoeft ook helemaal niet.

## Doelstellingen en opzet

De door ons ontwikkelde module omvat twaalf lessen en wordt afgesloten met een toets. Doel van de module is om leerlingen te leren werken met een model checker, in het bijzonder aanpassingen maken aan bestaande modellen, zelf eenvoudige modellen construeren, en gewenste eigenschappen van systemen formaliseren en uitrekenen met behulp van een model checker. De meeste lessen (acht van de twaalf) zijn practica waarbij de leerlingen zelf aan de slag gaan met Uppaal om diverse systemen te modelleren en analyseren:

- een geldautomaat
- een verkeerskruising

- een spoorwegovergang
- logische puzzels zoals “De wolf, de geit en de kool” en “De kannibalen en de missionarissen”
- een mutual-exclusion-algoritme (voorkomt dat deelprogramma’s elkaar verstoren door tegelijkertijd een gemeenschappelijke *resource* te gebruiken, bijvoorbeeld een printer of gedeeld bestand).
- een algoritme voor het kiezen van een leider in een ring
- het plannen van activiteiten in een werkplaats

Bij de meeste practicumopdrachten geven we de leerlingen een model, dat ze vervolgens moeten aanpassen. De resterende vier lessen worden gebruikt voor inleidingen over

- betrouwbaarheid van software
- model checking met Uppaal
- mutual-exclusion-algoritmen
- algoritmen voor het kiezen van een leider

De keuze van de diverse toepassingen is vrij willekeurig en varieert van jaar tot jaar. Belangrijk is dat de problemen zich eenvoudig laten uitleggen, relevant zijn vanuit praktisch oogpunt, maar het gedrag toch zo complex is dat correctheid van voorgestelde oplossingen moeilijk aangetoond kan worden zonder ondersteuning door een model checker. We proberen de leerlingen kennis te laten maken met de elegantie/schoonheid van gedistribueerde algoritmen.

## Ervaringen

Wij zijn nog steeds bezig met de fine tuning van de module en met de verdere ontwikkeling en verbetering van het lesmateriaal. Een aantal voorlopige indrukken/conclusies zijn:

- Verreweg de meeste leerlingen vinden het onderwerp van de module interessant en het werken met Uppaal leuk. De groep leerlingen die informatica volgt is nogal divers, met behalve leerlingen met profiel natuur/techniek bijvoorbeeld ook leerlingen met profiel cultuur/maatschappij. Dit maakt het moeilijk om de gehele populatie aan te spreken.
- De goede grafische interface van Uppaal en de robuustheid van het gereedschap zijn essentieel voor het succes van de module.
- Nauwe samenwerking tussen VO en WO was uiterst belangrijk bij de ontwikkeling van de module. De didactische aanpak die nodig is bij 5 vwo Informatica leerlingen verschilt wezenlijk van de aanpak bij studenten Informatica op een universiteit. Zo is voor 5 vwo leerlingen 50 minuten een verhaal aanhoren eigenlijk te lang, terwijl dit voor studenten meestal geen probleem is. Wanneer vijf leerlingen door vragen en opmerkingen laten blijken dat ze de stof interessant vinden en goed snappen, dan is dit geen indicatie dat dit ook geldt voor de rest. Een ervaren vwo-docent pakt signalen van de leerlingen veel eerder op, ziet sneller waar het tempo lager of hoger moet, bij welke leerlingen het kwartje wel of niet gevallen is, welke vaardigheden extra training behoeven, enz.

- Leerlingen zijn vaak verrassend goed in het meedenken over oplossingen van concrete problemen. Zo bedacht één leerling ter plekke het Le Lann–Chang–Roberts-algoritme voor het kiezen van een leider in een unidirectionele ring. Na één of twee voorbeelden zien leerlingen ook razendsnel waarom allerlei mutual-exclusion-algoritmen incorrect zijn.
- Leerlingen hebben verrassend veel moeite met het omzetten van een informeel idee naar een formeel model. Modelleren is moeilijk. Dit inzicht heeft ons ertoe gebracht om relatief steeds meer cursustijd te gebruiken om de leerlingen zelf te laten modelleren. Modelleren, zo blijkt, leer je toch vooral door het te doen.

Ons streven is om het komende jaar voldoende documentatie en cursusmateriaal te ontwikkelen, zodat de module ook beschikbaar gesteld kan worden aan andere geïnteresseerde docenten.

## Links

1. De cursuswebsite, <http://www.cs.ru.nl/~fvaan/Masterclass/olympus.html>
2. De model checker Uppaal, <http://www.uppaal.com/>
3. Het Sprint-UP project, <http://www.platformbetatechniek.nl/sprintup>