

Inquiry- and Research-based Teaching in a Course on Model Checking

Sebastian Krings, Philipp Körner, Joshua Schmidt

Niederrhein University of Applied Sciences, University of Düsseldorf

State of the Art

- Model checking research connects theoretical and practical aspects
- New algorithms are often implemented inside well-known model checkers
 - ▶ In development for many years
 - ▶ Complex
 - ▶ Large code volume
 - ▶ High entry barrier

However

This is seldom taken into account by university courses, which often remain on the theoretical level.



State of the Art ...

Our course wasn't any better.



State of the Art ...

Learning objectives:

- Students can present and compare different techniques
- Students know common algorithms and can implement them
- Students can summarize selected literature and are able to criticize
- Students can write their own specifications and evaluate them

So obviously

A purely frontal lecture-based course is the way to go.

State of the Art ...

Oh boy was our course missing the point.



Major Shortcomings

- Learning results reduced due to missing hands-on experience
- Scope of thesis topics is limited, as students have not learned how to appropriately address practical problems
- Missing experience in project work, tool usage and working collaboratively
 - ▶ Students do not meet expectations from industry
 - ▶ Skills could be acquired en passant in a programming project



Course Redesign

- High-level idea
 - ▶ Acquire the theoretical foundations by identifying and analyzing common software errors
 - ▶ Align these foundations with the body of knowledge
 - ▶ Design and implement a novel model checker as independently as possible
- Important aspects for success (Baron, et al.)
 - ▶ Selecting appropriate learning goals
 - ▶ Begin with problem-based learning before project work
 - ▶ Enable self-assessment and revision
 - ▶ Develop an atmosphere and social structures that support participation



Major Challenges

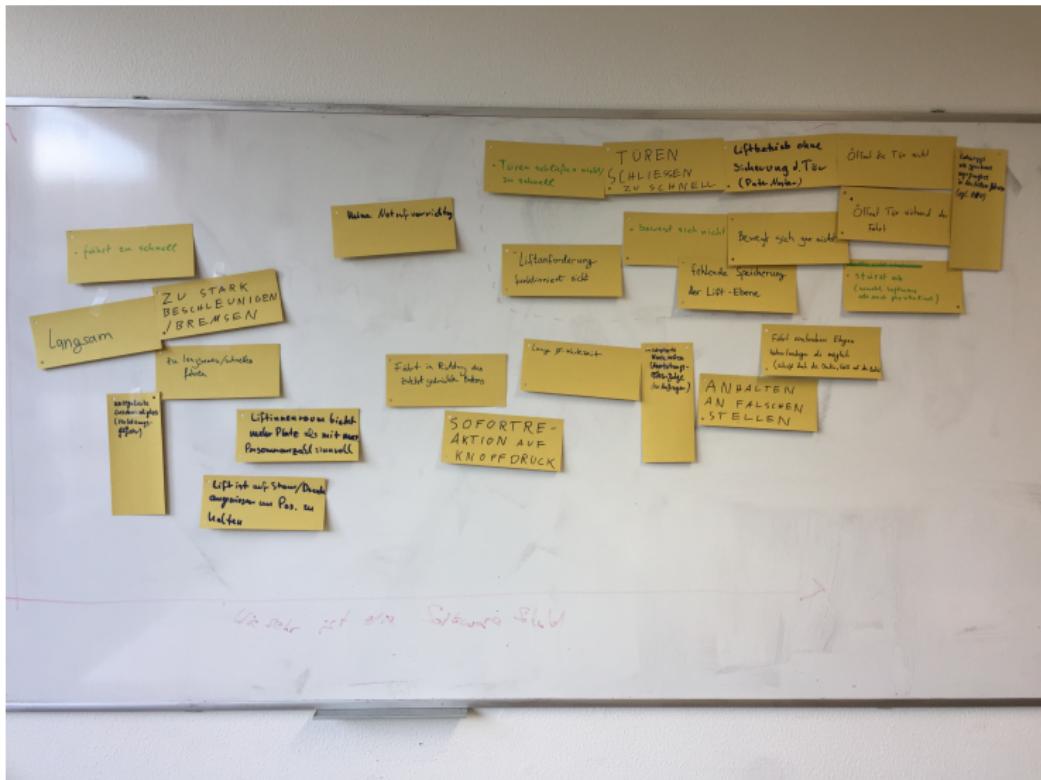
- Cognitive requirements are higher
 - ▶ Switch from knowledge reproduction to production
- Progression is less linear
- Individual workload is increased
 - ▶ Motivation and commitment has to be increased
- Research has to be controlled to avoid getting off track
 - ▶ As much freedom as possible, but guarantee intended learning outcomes
- Exams have to be prepared carefully to meet didactic requirements and exam regulations



Problem-based: Hazard Collection for Elevator



Problem-based: Sorted Hazard Collection



Course Sessions

After problem-based introduction, iterate between

- Project management
- Implementation
- R&D sessions
- Reflection & evaluation sessions



Project Management: Kanban

Backlog	To Do / Ready	In Development	Testing / In Review	Done
<ul style="list-style-type: none"><input checked="" type="checkbox"/> Issues which can not be resolved right now, either because they are blocked or because we do not have the resources to handle them right now. Added by wsysib<input checked="" type="checkbox"/> Model checker performance drop for CounterErr2 #86 opened by leuschel<input checked="" type="checkbox"/> Implement Cardinality of finite Sets #47 opened by Marekis enhancement	<ul style="list-style-type: none"><input checked="" type="checkbox"/> Issues which can be acted upon. However, nobody is working on them yet. Added by wsysib<input checked="" type="checkbox"/> Exceptions logged but not shown in the UI #93 opened by heinzware<input checked="" type="checkbox"/> Spurious counter-example found in the presence of deferred sets #92 opened by leuschel	<ul style="list-style-type: none"><input checked="" type="checkbox"/> Issues somebody is working on at the moment. Added by wsysib<input checked="" type="checkbox"/> Generalized Union #48 opened by Marekis enhancement	<ul style="list-style-type: none"><input checked="" type="checkbox"/> Resolved issues that have not been tested. Please test them and close + move them to done if they are indeed resolved. If further issues are discovered during testing, move them back to "In Development". Added by wsysib	<ul style="list-style-type: none"><input checked="" type="checkbox"/> We are done! Added by wsysib<input checked="" type="checkbox"/> Implement equal(Ast,...) Method to enable Node being checked for equality #101 opened by x-mos-x enhancement<input checked="" type="checkbox"/> Implement LTL model checking #102 opened by wsysib enhancement<input checked="" type="checkbox"/> Büchi Automaton of G (1≥1) includes only accepting states #104 opened by wsysib bug invalid

Project Management: Kanban

Backlog 3

- Issues which can not be resolved right now, either because they are blocked or because we do not have the resources to handle them right now.

Added by wysiib
- Model checker performance drop for CounterErr2**

#86 opened by leuschel
- Implement Cardinality of finite Sets**

#47 opened by Mareikes
enhancement

To Do / Ready 3

- Issues which can be acted upon. However, nobody is working on them yet.

Added by wysiib
- Exceptions logged but not shown in the UI**

#93 opened by heinzware
- Spurious counter-example found in the presence of deferred sets**

#92 opened by leuschel

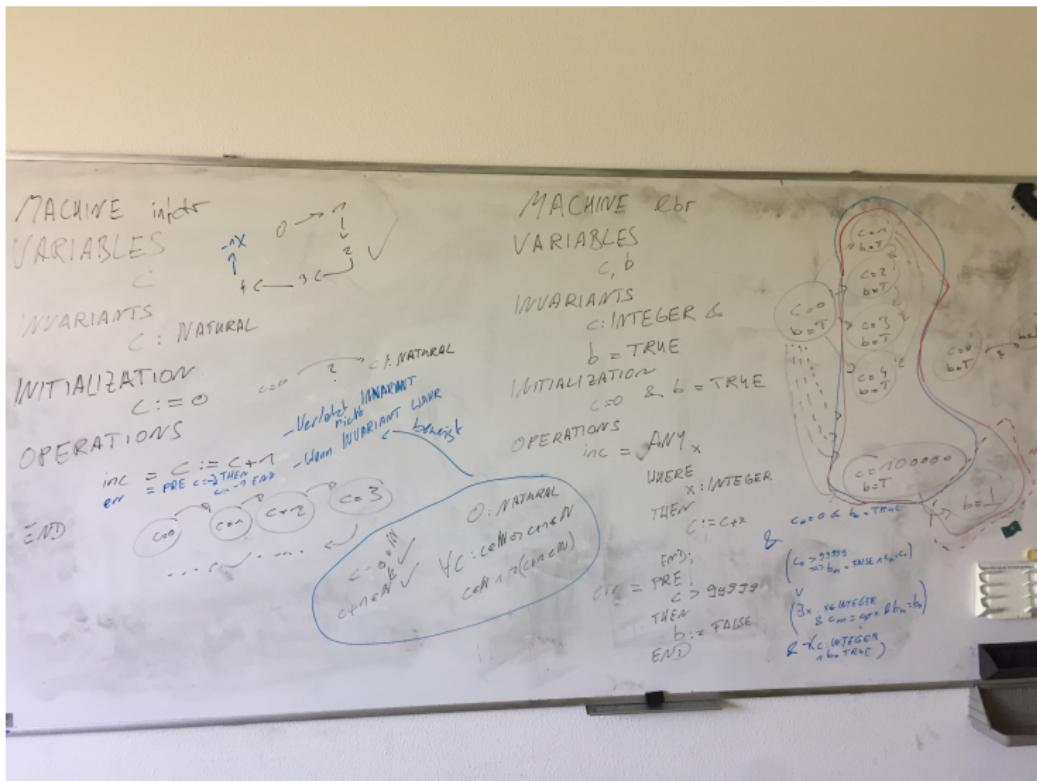


Project Management: Kanban

The image shows a Kanban board with three columns: "In Development", "Testing / In Review", and "Done".

- In Development:** Contains 2 items.
 - A note: "Issues somebody is working on at the moment." added by wysilib.
 - An enhancement ticket: "#48 Generalized Union" opened by Mareike, with status "enhancement".
- Testing / In Review:** Contains 1 item.
 - A note: "Resolved issues that have not been tested. Please test them and close + move them to done if they are indeed resolved. If further issues are discovered during testing, move them back to "In Development"."
- Done:** Contains 100 items, with the first few visible:
 - "We are done!" added by wysilib.
 - A ticket: "#101 implement equalAst(...) Method to enable Node being checked for equality" with status "enhancement".
 - A ticket: "#96 implement LTL model checking" with status "enhancement".
 - A ticket: "#104 Büchi Automaton of G {1=1} includes only accepting states" with status "bug invalid".

R&D: Algorithm Development



Course Evaluation

Different approaches to course evaluation

- Peer review by other teachers
- Direct student feedback
- Grades
- Learning data analytics

Grading

- Exam should measure both theoretical and practical aspects
- Ensure that grading complies with the examination regulations
- Improve constructive alignment (Biggs, 1996):
 - ▶ Formative part: constant participation documented using the Kanban board
 - ▶ Attitude, soft skills observed but hard to grade
 - ▶ Theoretical foundations by summative exam
- Combined exam verifies learning objectives!

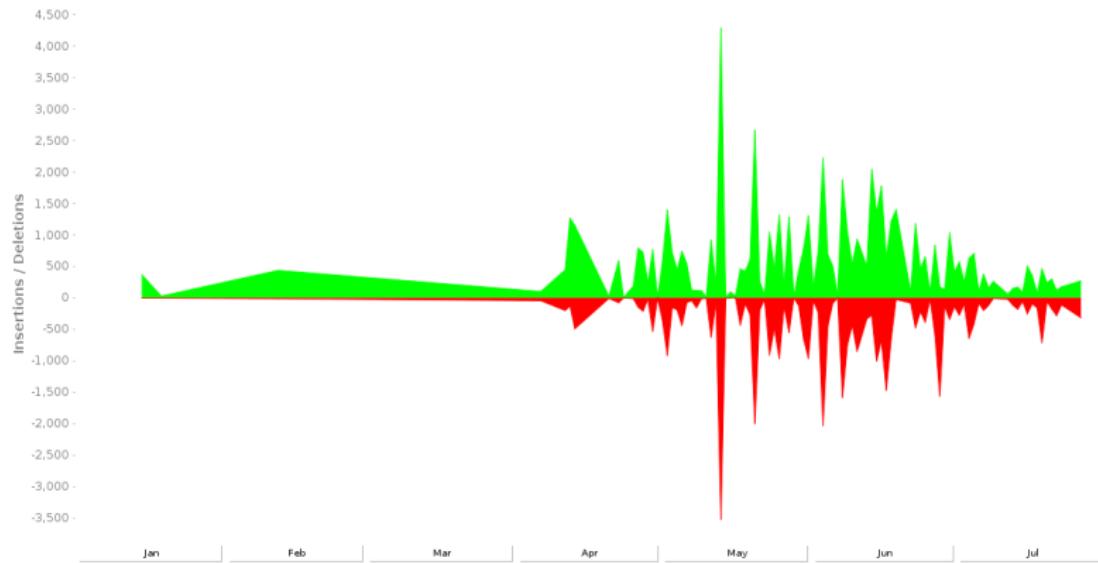


Grades

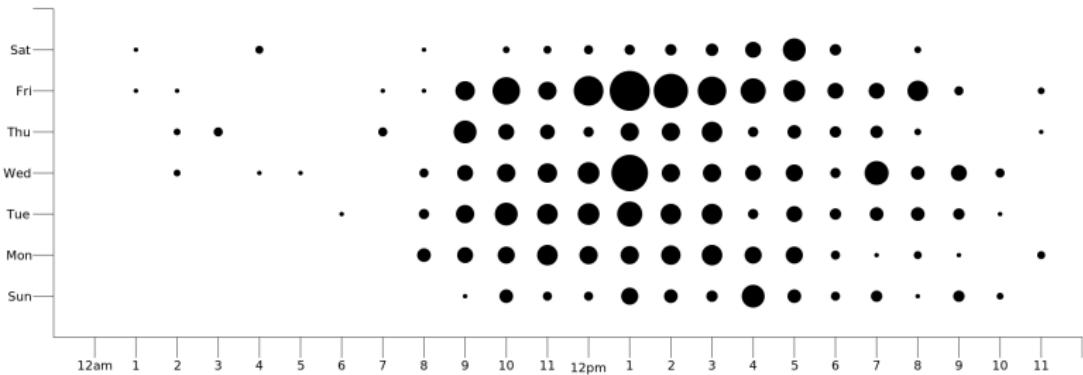
	2014	2015	2016	2017	2018
# Students	2	5	7	6	5
Ø Grade	1.85	2.58	1.71	1.28	1.88



Evaluation: Additions and Deletions to Sources



Evaluation: Activity



Experience Report on an Inquiry-Based Course on Model Checking

Sebastian Krings

Universität Düsseldorf, Hochschule Niederrhein

krings@cs.uni-duesseldorf.de

Philipp Körner, Joshua Schmidt

Universität Düsseldorf

{koerner,schmidt}@cs.uni-duesseldorf.de

Abstract

coming to grips with internal workings. In consequence, typical courses on model checking start on a



Last Step of Research: Publication

- Not part of the course, as we could not reasonable expect anything
- Still 3 students were interested
- Outside of curriculum:
 - ▶ Discuss publication process, peer review, etc.
 - ▶ How to write interesting paper (mostly following Peyton Jones)
 - ▶ Brainstorm possible topics and ideas
 - ▶ Writing, meetups for synchronization



One Question Remains

To what extent where our students
doing relevant research?



Dissemination: Student Article



ECEASST

Writing a Model Checker in 80 Days: Reusable Libraries and Custom Implementation

Jessica Petrasch, Jan-Hendrik Oepen, Sebastian Krings, Moritz Gericke

Institut für Informatik, Universität Düsseldorf
Universitätsstr. 1, D-40225 Düsseldorf

{jessica.petrasch, jan-hendrik.oepen, moritz.gericke}@hhu.de
krings@cs.uni-duesseldorf.de

Abstract: During a course on model checking we developed BMoth, a full-stack model checker for classical B, featuring both explicit-state and symbolic model checking. Given that we only had a single university term to finish the project, a particular focus was on reusing existing libraries to reduce implementation workload.

Conclusions

- Goals met
- Realization more hazzle-free than anticipated
- Highly motivating for students, immediate sense of relevance
- Scaling and proper knowledge propagation difficult

Summary

10 of 10, would teach that way again.

Thank you!

Any questions?