Confirmability and reproducibility of scientific insights and artefacts

From coding to results (a management summary)

M. Aschermann, Ph. Kraus
Clausthal University of Technology
30. June 2017

Motivation – Problem Statement

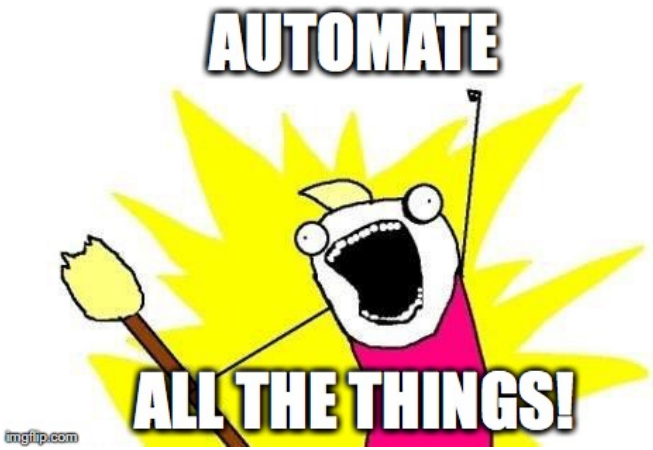
- In publications, academic research often only applies *confirmability*, *transparency* and *reproducibility* to (formal) models and analysis.
- Typical workflow to answer research questions:
  modelling → coding → simulating → analysing → more coding ...
- A correct model does not imply a correct, i.e. error-free implementation
  ⇒ correctness of simulation results? ⇒ correctness of model?
- Often no implementation referenced in paper (→*transparency*)
  ⇒ how to assess correctness and ensure reproducibility without?
- DFG guidelines [1, 2] state that members of their funded projects have to document and publish research data for useful reuse (raw or structured data) and make it available for at least 10 years.
- DFG documents[1] mentioning source code, e.g. [3, 4] clearly advocate disclosure and publication of source code.

[1] http://www.dfg.de/service/suche_de/index.jsp?q=quellcode

Motivation – Our goals

- We propose a workshop to answer questions like
  - How to ensure high code quality, reproducibility, correctness?
  - How and where can we publish our code and data?
  - How to cite code and data used in publication?
- Individually help PhD candidates chosing the right technology related to [Problem Statement]
- Provide PhD candidates with necessary workflows
- PhD candidates should focus on research
  $\Rightarrow$ automate as much as possible
  - documentation
  - code quality and correctness

  to make life (i.e. research) easier



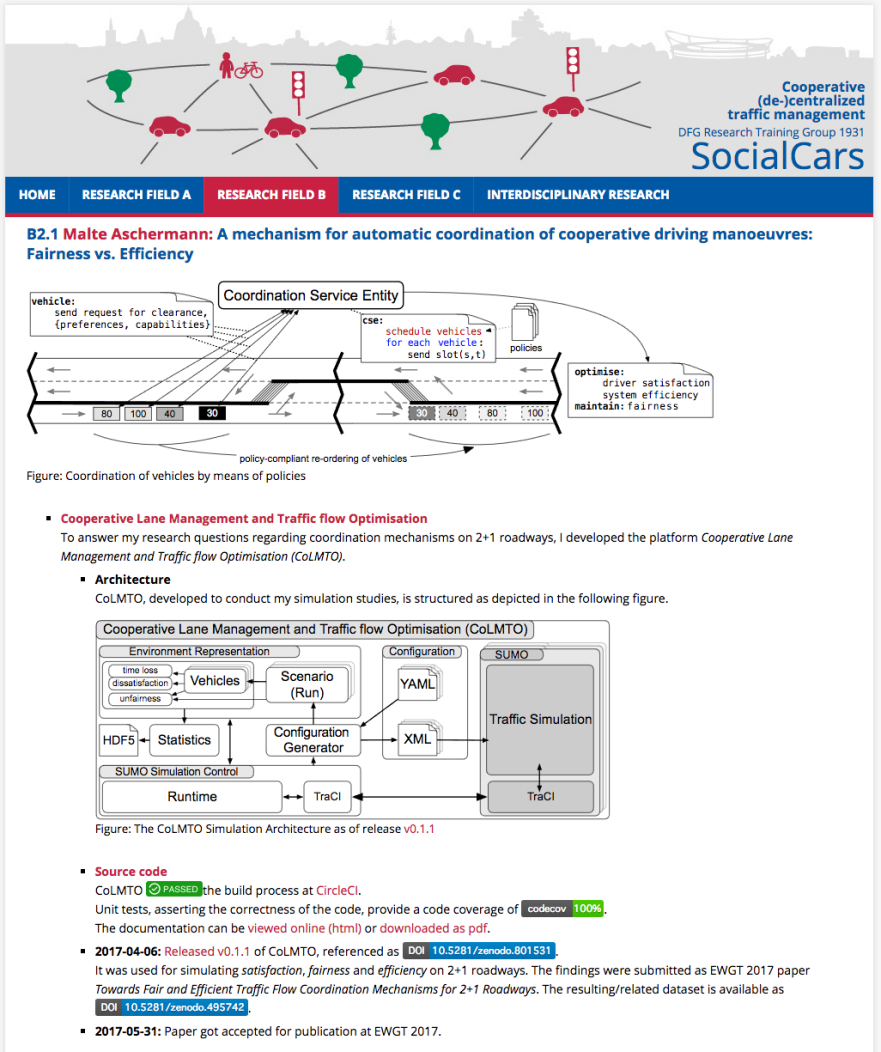
Useful platforms and tools

In the workshop, we would introduce the following platforms and tools

1. Source code repository and documentation (→*transparency*)
   - GitHub and SocialCars GitHub site github.com/SocialCars
   - Hugo to build static sites with markdown
2. Testing and automation of task (→*correctness* and *reproducibility*)
   - Coveralls and Codecov (check whether your code is tested, i.e. correct)
   - FindBugs (checks for common errors in Java code)
   - Circle CI, Travis CI (automate repetitive tasks)
3. Publication of citable code and data (→*long-term availability*)
   - Zenodo (assigns DOIs to datasets and code)

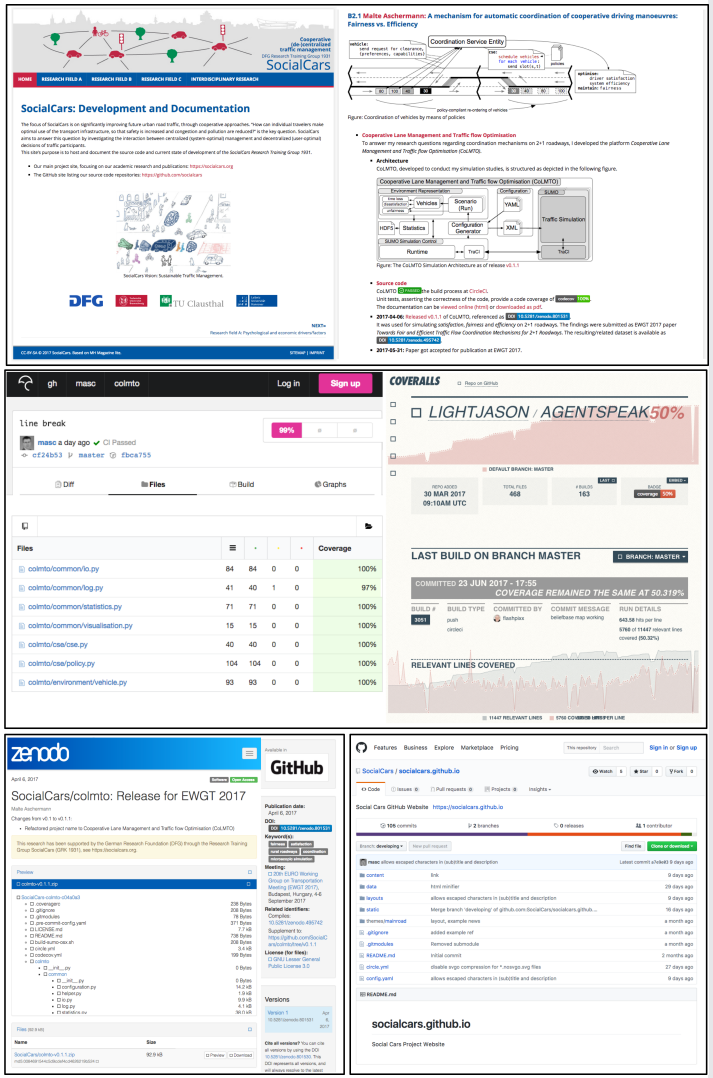
GitHub: Source code repository and documentation

- We propose GitHub to host code in a DFG-compliant manner
- For documentation purposes websites can be hosted, i.e. socialcars.github.io
- Documentation of models and implementation
  (→ see DFG requirements)
- Individual presentation of each PhD candidate's work
- Content written in markdown syntax
- Automatic deployment via CircleCI



Thank you for your attention

- For further discussion and questions we would offer a workshop to individually help each PhD candidate on this topic.



References

[1] Leitlinien zum Umgang mit Forschungsdaten, 2015.
http://www.dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf.

[2] Umgang mit Forschungsdaten, 2017.
http://www.dfg.de/foerderung/antragstellung_begutachtung_entscheidung/antragstellende/antragstellung/nachnutzung_forschungsdaten/.

[3] Basisinformationen zum Forschungsdatenmanagement, 2016.
http://www.dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/basisinformationen_forschungsdatenmanagement.pdf, Kurzfassung von *Forschungsdatenmanagement in den Sozial-, Verhaltens- und Wirtschaftswissenschaften* https://www.ratswd.de/dl/RatSWD_Output3_Forschungsdatenmanagement.pdf.

[4] Merkblatt Informationsinfrastrukturen für Forschungsdaten, 2016.
http://www.dfg.de/formulare/12_14/12_14_de.pdf für Wissenschaftlichen Literaturversorgungs- und Informationssysteme.