Software for Science U!REKA Edinburgh 2017

Marten Teitsma
Willem Brouwer
Frank Aldershoff
Kees Rijsenbrij

Software for Science I

Software is changing the way we conduct science, in terms of the sophistication of the analyses we perform, and the volume of data we can process. [4]

Software for Science I

Software is changing the way we conduct science, in terms of the sophistication of the analyses we perform, and the volume of data we can process. [4]
When do scientist use software or more general, computer technology?

Software for Science I

Software is changing the way we conduct science, in terms of the sophistication of the analyses we perform, and the volume of data we can process. [4]

When do scientist use software or more general, computer technology?

- Developing hypotheses and finding literature.
- Gathering data.
- Analysis of data.
- Publication of results.

Software pervades every domain of science [5]

Software for Science II

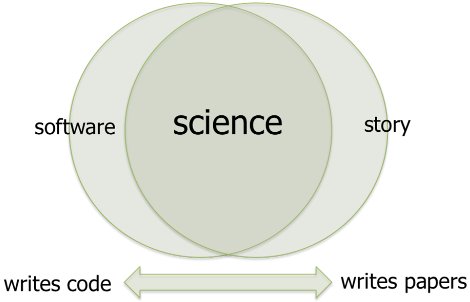

Figure: In many areas of research, science is now produced at the intersection of the 'software' and the 'story'. From: [3]

Software for Science III

Specific problems with respect to software for science [7]:

- quickly changing requirements,
- bespoke software [2],
- pressure to rapidly produce scientific publications,
- time is the constraint instead of functionality [2],
- competition between maintainable and performance code,
- lack of metrics,
- many scientist adopt and use software critical to their research on non-scientific reasons [5],

Reproducibility

Replication is the ultimate standard to judge scientific claims.

Reproducibility

Replication is the ultimate standard to judge scientific claims. But more than 70% of researchers have tried and failed to reproduce experiments. About 80% blamed this on the unavailability of methods and code. [1]

Reproducibility

Replication is the ultimate standard to judge scientific claims. But more than 70% of researchers have tried and failed to reproduce experiments. About 80% blamed this on the unavailability of methods and code. [1]

Standard is not realistic because of:

- costs of research,
- time needed to do research,
- problems software for science as mentioned.

Reproduction of research is better than no replication. We should create a culture of reproducibility [6]

Sustainability

Sustainable software is software you use today and which will be available in the future (Software Sustainability Institute).

Intrinsic sustainability (aspects of the software itself) [2]:

- documentation
- testability
- modular
- third party libraries
- usefulness
- scalable

Extrinsic sustainability (the environment of the software) [2]:

- open available
- shared or co-owned
- resourced
- actively maintained
- independence from infrastructure
- supported

FAIR

The FAIR guiding principles for data are [8], to be:

- Findable.
- Accessible.
- Interoperable.
- Reusable.

By use of metadata stating all sorts of information about data.

The minor

Students develop software to make scientific research possible.
Characteristics of the minor:

- 30 ects (half a year),
- theoretical and practical elements,
- international.

Collaboration with ALICE

Amsterdam University of Applied Sciences is associate member of the ALICE collaboration. This and other relations are embedded in the minor Software for Science.

Collaboration with ALICE

Amsterdam University of Applied Sciences is associate member of the ALICE collaboration. This and other relations are embedded in the minor Software for Science.

Research is being done by:

- interns (midterm and final),
- teams of students during a minor and summer school,
- lecturers/researchers.

Educational elements

Elements of the minor are:

- personalised educational route,
- boot camp (3 or 4 weeks full time),
- courses (10 weeks 2 times 4 hours),
- developing a product with a team of 3 or 4 students
- lecturers/researchers,
- coaching,
- writing and research skills:
  - technical report (team product),
  - research paper (individual product)
- assessment.

Marten Teitsma
Willem Brouwer
Frank Aldershoff
Kees Rijsenbrij

- CERN (European Organization for Nuclear Research, Conseil Européen pour la Recherche Nucléaire)
- LHC (Large Hydron Collider)
- ALICE (A Large Ion Collider Experiment)

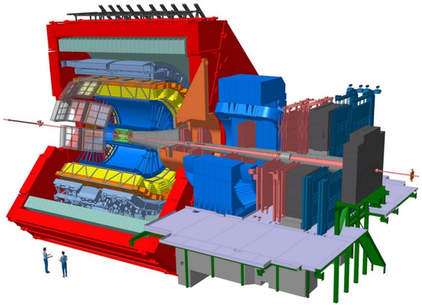

Figure: The ALICE detector

Projects for ALICE

Marten Teitsma
Willem Brouwer
Frank Aldershoff
Kees Rijsenbrij

Research already started:

- data distribution and load balancing, i.e. load balancing software,
- control, configuration and monitoring, i.e. data-flow visualisation software,
- system control and data acquisition, i.e. a logging system.

Figure: Tomb of Hilarus Fuscus

Development of a 4D geographic information system for archaeological purposes.

Figure: Westerbork Telescopes

Possible research subjects will be elements of:

- Square Kilometre Array (SKA),
- Low-Frequency Array (LOFAR).

Analysis

During this semester (2017) we look into:

- determining the target audience,
- specifying projects with ASTRON and eScience Center,
- defining the educational model in further detail,
- coaching interns for and doing research on the projects for ALICE.

Development

From January until June (2018):

- developing educational material,
- developing competence models for each project,
- coaching interns for and doing research on the projects for ALICE, ASTRON and eScience Center,
- advertising the minor.

Implementation

In the beginning of July a summer school is organised.
At the end of August 2018 we will start the minor Software for Science.

Questions?

Marten Teitsma
Willem Brouwer
Frank Aldershoff
Kees Rijsenbrij

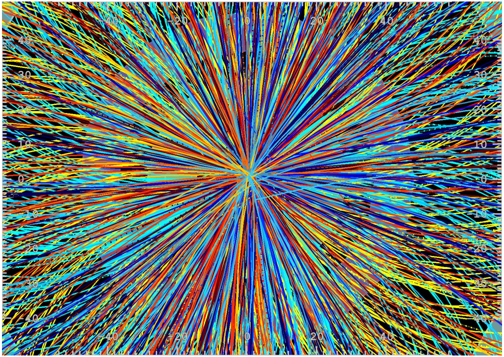

Figure: Events recorded by the ALICE experiment from the first lead ion collisions, at a centre-of-mass energy of 2.76 TeV per nucleon pair. From: cdsweb.cern.ch/record/1305399

Contact me: m.teitsma@hva.nl

Monya Baker.
Is there a reproducibility crisis? a nature survey lifts the lid on how researchers view the'crisis rocking science and what they think will help.
Nature, 533(7604):452–455, 2016.

Mario Rosado de Souza, Robert Haines, and Caroline Jay.
Defining sustainability through developers eyes: Recommendations from an interview study.
In *Proceedings of the 2nd Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE 2014)*, 2014.

Caroline Jay, Robert Haines, Markel Vigo, Nicolas Matentzoglu, Robert Stevens, Jonathan Boyle, Alan Davies, Chiara Del Vescovo, Nicolas Gruel, Anja Le Blanc, David Mawdsley, Dale Mellor, Eleni Mikroyannidi, Richard PeterRollins, Andrew Rowley, and Julio Vega.
Identifying the challenges of code/theory translation: report from the code/theory 2017 workshop.
Research Ideas and Outcomes, 3:e13236, 2017.

Caroline Jay, Rawan Sanyour, and Robert Haines.
Not everyone can use git: Research software engineers recommendations for scientist-centred software support (and what researchers really think of them).
Journal of Open Research Software, 2017.

Lucas N Joppa, Greg McInerny, Richard Harper, Lara Salido, Kenji Takeda, Kenton O'hara, David Gavaghan, and Stephen Emmott.
Troubling trends in scientific software use.
Science, 340(6134):814–815, 2013.

Roger D Peng.
Reproducible research in computational science.
Science, 334(6060):1226–1227, 2011.

Francisco Queiroz, Raniere Silva, Jonah Miller, Sandor Brockhauser, and Hans Fangohr.

Good usability practices in scientific software development.
arXiv preprint arXiv:1709.00111, 2017.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al.
The fair guiding principles for scientific data management and stewardship.
Scientific data, 3:160018, 2016.