Lightining Talks at the First Workshop on Research Software Engineering (RSE) @ KIT

Overview

Name	Talk
Niklas Baumgarten	Software Development Workflow of the Finite Element Software Library M++
Juan Pedro Gutiérrez Hermosillo Muriedas	Continuous Bechmarking and Energy monitoring
Andre Platzer	Software Design Impact Observations from 3 Generations of Theorem Provers
Mattias Ulbrich and Alexander Weigl	Research Software Development of the KeY Theorem Prover
Mathias Krause	OpenLB (www.openlb.org)
Jörg Schaarschmidt	Transforming complex simulation protocols into reusable workflows for rapid prototyping with SimStack
Axel Loewe	Automated Software Metadata Conversion and Publication Based on CodeMeta
Shashank Shekhar Harivyasi	RSE Worldwide

The lightning talks will be presented in the same order as stated in this document.

List of abstracts

Software Development Workflow of the Finite Element Software Library M++

Niklas Baumgarten, IANM3

We present the development workflow of M++ - a finite element methods library for large-scale high performance computations of physical models https://git.scc.kit.edu/mpp/mpp/. In particular, we will talk about our adaption of a continuous delivery workflow including automated benchmarking and archiving of numerically achieved results. Lastly, we present an outlook on future developments to further improve our RSE.

Continuous Bechmarking and Energy monitoring

Juan Pedro Gutiérrez Hermosillo Muriedas, SCC

How are you tracking the performance of your software over time? Have you looked at how energy efficient your software is? What steps or tools are you using for that?

Software Design Impact Observations from 3 Generations of Theorem Provers

Andre Platzer, KASTEL-VADS

Provides a retrospective on the impact of design decisions from the developments of three theorem provers for hybrid systems. While all three theorem provers implement closely related logics of the family of differential dynamic logic, they pursue fundamentally different styles of theorem prover implementations. Since the three provers KeYmaera, KeYmaeraD, and KeYmaera X share a common core logic, yet no line of code, and differ vastly in prover implementation technology, their logical proximity yet technical distance enables us to draw conclusions about the various advantages and disadvantages of different prover software implementation styles for different purposes, which we hope are of generalizable interest beyond theorem proving.

Research Software Development of the KeY Theorem Prover

Mattias Ulbrich and Alexander Weigl, KASTEL / Anwendungsorientierte Formale Verifikation

KeY is a theorem prover for the deductive verification of Java programs. The development of KeY has a long history of over 20 years from which many challenges of the software engineering arises.

OpenLB (www.openlb.org)

Mathias Krause, Lattice Boltzmann Research Group – IANM, MVM

OpenLB is an internationally well regognized open source software for fluid flow simulations on high performance computers. It was esablished in 2006 as research software and community project. The sofware can take great advantage of modern massive parallel CPU-GPU clusters and the community offeres anually taking place spring schools on the method and the software.

Transforming complex simulation protocols into reusable workflows for rapid prototyping with SimStack

Jörg Schaarschmidt, INT

Using high-throughput simulations has allowed researchers to select the most promising materials for experimental studies and gain insights that are otherwise inaccessible through experiments alone. However, the current approach of creating patchwork solutions on a case-by-case basis requires extensive scripting, command-line execution, and a deep understanding of various methods and tools for data preparation, transfer, execution, and analysis. This creates issues with reproducibility, reusability, transferability, and flexibility in protocols. We therefore use and develop the Workflow Framework SimStack to make complex simulation protocols accessible to non-computational specialists.

Automated Software Metadata Conversion and Publication Based on CodeMeta

Axel Loewe, IBT

Different metadata standards exist for different steps in the research software publication process. The Citation File Format (CFF) became very popular to provide information on how users are supposed to cite the software. For software archiving, DataCite is one of the established standards. The CodeMeta standard is an extension of schema.org specifically tailored to research software. All of these standards serve a purpose and are required for specific phases of the software lifecycle. However, research software developers should ideally not be burdened with maintaining a whole set of metadata files in different formats and largely overlapping content. This poses a risk both to data consistency and to adoption of good software publication practices in the first place. Therefore, we developed Python pipelines that put the developers in a position to only maintain a CodeMeta file. CFF and DataCite files are automatically generated based on the CodeMeta file. The open source pipelines (https://www.openCARP.org/CI) can easily be integrated in continuous integration and deployment environments and are being successfully used for the openCARP project (https://www.openCARP.org). Further aspects of the software lifecycle that are covered by the set of continuous integration scripts are creation of tagged releases including the update of

the metadata files, creation of BagIt and BagPack files, deposition of releases in the RADAR research data repository, and the synchronization of documentation files in the repository to the project's webpage. We believe the automated metadata conversion based on CodeMeta can be a useful tool for many research software developers and can facilitate the adoption of good software publication practices by reducing the effort for developers.

RSE Worldwide

Shashank Shekhar Harivyasi, IBCS-FMS

The RSE landscape is international and diverse. With my talk I want to create awareness about the scope of RSE movement around the world and highlight the existence of de-RSE, RSESoc (UK), Nordic-RSE, RSE Asia, RSSE Africa as well as the International Council of RSE Associations. I would also talk a bit about diversity in RSE context (i.e. diversity of skill levels, academic domains and use cases) on top usual demographical diversity.