Introduction to the Minitrack Software Sustainability: Strategies for Long-Lasting and Usable Software

Maytal Dahan
Texas Advanced Computing Center,
The University of Texas at Austin
maytal@tacc.utexas.edu

Joe Stubbs
Texas Advanced Computing Center,
The University of Texas at Austin
jstubbs@tacc.utexas.edu

Sandra Gesing
Discovery Partner Institute
University of Illinois System
ssgesing@uillinois.edu

Abstract

The focus on software usability, long-lasting and reproducible software is a timely one that spans various domains of science and significant investment of research funding across the US, Europe, U.K, and elsewhere. The three concepts — usability, sustainability and reproducibility are interconnected with each other and cover a wide range of application areas. They affect all layers of the software process – from enabling reproducing experiments via an easy user interface to using containerization for application portability. This minitrack focused on the broad spectrum of submissions that deal with complex scenarios such as containerization, strategies for long-lasting software, usability and user interface issues, handling data curation and provenance and more.

1. Introduction

The three concepts usability, sustainability and reproducibility are interconnected with each other and cover a wide range of application areas. They affect all layers of the software process - from enabling reproducing experiments via an easy user interface to using containerization for application portability. Such concepts are also relevant in the building of Science Gateways (also known as virtual laboratories or virtual research environments), which by definition serve communities with end-to-end solutions tailored specifically to their needs. Software survivability involves a wide scope that can potentially include the following topics:

- Web-based solutions (web sites, science gateways, virtual labs, etc.)
- Application Programming Interfaces (APIs)
- Computational and Data-Intensive Workflows
- Novel approaches in containerization

- Sustainability practices in software development
- System architectures for testing and continuous integration
- Emerging best practices in Machine Learning software
- Best practices and Key Success Factors for usability, survivability and reproducibility

This minitrack, Software Sustainability: Strategies for Long-Lasting and Usable Software, introduces the wide variety of accepted papers to HICSS-55. It focused on the broad spectrum of submissions that deal with complex scenarios such as containerization, strategies for long-lasting software, usability and user interface issues, handling data curation and provenance and more.

2. Accepted Papers

The minitrack received four papers with an excellent breadth of topics from software architecture and complexity to Application Programming Interface (API) design for large-scale data aggregation and processing, to accelerating research with gateway frameworks. We accepted two papers to this minitrack that introduce the following topics:

- The architecture of complexity
- Application Programming Interface (API) Design for data-intensive applications

One paper selected for this minitrack, "The Architecture of Complexity Revisited: Design Principles for Ultra-Large-Scale Systems", examines a set of ultra-large-scale (ULS) systems, including naturally-occuring systems and systems created by humans, and it presents three architectural principles embodied by ULS systems. It goes on to describe tactics that can be used to design ULS systems

HICSS

effectively, organized into "Building Blocks", "Aggregation" and "Interaction". The paper contributes timely guidelines for building ULS systems that can be sustained, but the ideas presented are likely to be of use outside of the context of ULS systems.

Lastly the authors of the paper, "Design and Implementation of Web APIs for Supporting Data Product Visualization and Dissemination in Science Gateways" presents the requirements, design and implementation of an Application Programming Interface (API) developed to support online access and visualization of large climatological data from around Hawai'i. The authors tackle challenges related to aggregating data from climate sensor stations deployed across Hawai'i, generating zip packages of large file collections, and sending gridded map files directly to end users. Packaging and deployment based on Docker containers is also described.

3. Conclusion

These papers show a range of applications and impact of software sustainability in production and research software. They cover crucial aspects such as reproducibility and cultural approaches. We hope you will join us for interesting presentations and lively discussions on software sustainability, reproducibility, challenges, and solutions for our evolving landscape. We aim at continuing with this minitrack in the future at HICSS and encourage authors to contribute their research and viewpoints on software sustainability with its many facets and areas.