Guest Editors' Introduction

Introduction to Accelerating Scientific Discovery With Reusable Software

Scott Lathrop
Shodor Education Foundation, Inc. and National Center for Supercomputing Applications (NCSA), University of Illinois

Mike Folk
The HDF Group

Daniel S. Katz
National Center for Supercomputing Applications (NCSA), Computer Science Department, Electrical and Computer Engineering Department, and the School of Information Sciences (iSchool), University of Illinois

Lois Curfman McInnes
Mathematics and Computer Science Division, Argonne National Laboratory (ANL)

Andy Terrel
NumFOCUS

We are pleased to present this special issue on "*Accelerating Scientific Discovery with Reusable Software*." Our goal is to recognize the significant increase in the development and usage of high-quality reusable software that has taken place over the past decade. The intent of this special issue is to provide the computational- and data-enabled research community with information about the benefits that can be achieved from developing and utilizing reusable software that follows standards of quality and good practices. Extensive community efforts are under way to create and promote high quality, sustainable, reusable software that can enhance the quality and reproducibility of computational results, often at little or no cost. We use the term "scientific community" to refer to researchers and scholars in all disciplines, not just science and engineering, and the term "scientific discovery" to refer to research and scholarship in all fields of study.

We have observed a significant and growing focus on the reproducibility of scientific discoveries and the level of investment in sustainable software development to advance scientific discovery. A number of the papers in this special issue highlight these efforts as well as the challenges resulting from the lack of use of high-quality reusable software. Significant investments have been committed to software development, but a large percentage of that investment has been conducted in an ad hoc manner without regard for best practices and standards. This situation has resulted in research publications that are dependent upon software of which only a small percentage can be reused to validate the research conclusions.

When discussing reusable software, we refer to numerical libraries, modules, community

Digital Object Identifier 10.1109/MCSE.2019.2893745
Date of current version 8 March 2019.

Published by the IEEE Computer Society

codes, and software tools as well as good programming techniques that have been in development and use for more than 40 years. These very high-quality software solutions are often overlooked by research teams whose lack of knowledge and awareness leads them to reinvent similar software solutions. More often than not, these research teams have strong domain research knowledge but little software engineering experience or programming expertise. The net effect is that some research teams neglect to pay attention to important details, such as precision and performance. In contrast, scientific teams that have adopted highly developed software solutions and practices have reduced their overall software development time, increased the accuracy and reliability of their results, and decreased their "time to science," enabling them to conduct more complex analyses and advance knowledge in the field.

The special issue papers address the challenges and problems that arise from homegrown approaches. The papers include best practices that have emerged from teams with many years of experience developing sustainable scientific software, discipline-focused case studies, alternative programming paradigms, and strategies for building communities of practice. A brief synopsis of the papers and the lead authors in this special issue are given below.

- Katz *et al.* survey organizational efforts aimed at developing better scientific software and summarize the perspectives covered in the papers in this special issue.
- Kellogg *et al.* describe best practices learned from developing reusable software for the geophysics community, with lessons learned that can benefit other disciplines including strategies for building a community of practice.
- Lanore describes a component-based programming paradigm approach for building and using reusable software and demonstrates it with a Bayesian inference application case study.
- Richard *et al.* describe the computational chemistry framework NWChem and the updated NWChemEx package, which was designed by using standardized approaches, to enable the computational chemistry community to benefit from emerging exascale computing environments.
- Ram *et al.* describe the rOpenSci Project, which focuses on best practices for developing software to support reproducible science.
- Adorf *et al.* describe how to professionally develop reusable scientific software and how to balance this with obtaining rapid scientific results.

We recognize that these papers represent only a small subset of the broad range of reusable software, projects, and approaches currently available. We hope the papers will benefit research teams, lead to adoption of software development standards, provide a foundation for enhancing scientific reproducibility, and thereby advance our knowledge and understanding of the world around us. We further hope that these papers will foster expanded community efforts, elicit recommendations for additional best practices, and raise awareness of other high-quality reusable software available to specific disciplines and the general community. The papers provide a view into some excellent community efforts and a collection of best practices that can be expanded by the community.

We encourage you to contribute to *accelerating scientific discovery* by participating in community efforts such as https://bssw.io, https://software.ac.uk and http://urssi.us to enhance the development and applications of reusable software.

APPENDIX
RELATED WORKS

1) D. S. Katz *et al.*, "Community organizations: Changing the culture in which research software is developed and sustained," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8565942
2) L. H. Kellogg, W. Bangerth, L. J. Hwang, T. Heister, and R. Gassmoller, "The role of scientific communities in creating reusable software: Lessons from geophysics," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8556019

3) V. Lanore, "Fostering reuse in scientific computing with embedded components: Application to high-performance Bayesian inference for bioinformatics," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8554141
4) R. Richard *et al.*, "Developing a computational chemistry framework for the exascale era," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8565936
5) K. Ram *et al.*, "A community of practice around peer-review for long-term research software sustainability," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8554180
6) C. S. Adorf, V. Ramasubramani, J. A. Anderson, and S. C. Glotzer, "How to professionally develop reusable scientific software—And when not to," *Comput. Sci. Eng.*, vol. 21, no. 2, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8558687