

LA-UR-22-23689

Approved for public release; distribution is unlimited.

Title: LSSw Panel Introduction

Author(s): Herring, Angela M.

Intended for: Panel discussion: LSSw panel, April 21, 2022

Issued: 2022-04-21



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LSSw Panel Introduction

Angela Herring

Angela Herring

About me:

- Group Leader, XCP-4, Continuum Models and Numerical Methods
- Project Leader/PI, Lynx project
- Scientific software developer since 2006
- Project leader/PI/Scrum master since 2010
- Specialization in leading multi-disciplinary scientific software teams
- Have led ~8 teams since 2010

Ecosystem:

- Lynx project develops the remap and interface reconstruction libraries for LANL's Next Generation Code effort, Ristra
- Lynx tools are designed to interact well with the Ristra backbone, FleCSI, are not FleCSI applications
- XCP-4 develops and rapidly deploys physics models for multiphysics codes: HE, Turbulence, and numerical methods (remap, mesh smoothing, hydrodynamics, etc)
 - Support four major multiphysics projects

Prompts

From the perspective of the scientific software ecosystem you represent:

- Do you think there is value in designing, implementing, and delivering application-specific libraries, tools, and environments as reusable components?
 - Writing libraries:
 - Reduces overall lines of code which must be maintained
 - Enables shared staffing across codebases
 - Reduces lines of code in multiphysics codebases
- What has worked and not worked well with past efforts in this area?
 - Worked well:
 - Sharing remap code to link two codes together
 - Sharing interface reconstruction code
 - Trickier:
 - Sharing inline remap capabilities
 - Ensuring performance is maintained when accessing mesh data
- What are some near-term opportunities for componentization in your application area?
 - Cross-cutting capabilities:
 - HE libraries
 - Turbulence libraries
 - Smoothing libraries
 - Material model libraries
- How could this kind of software collection be adopted and sustained?
 - Funding must push/enforce code reuse
 - LANL has recently created a project dedicated to sharing libraries