ORNL/TM-2021/2268

# NEAMS Workbench Workflows

Robert A. Lefebvre
Marc-Olivier G. Delchini
Nicolas E. Stauff
Patrick C. Shriwise
Brandon R. Langley
Benjamin S. Collins

**September 30 2021**

**Approved for public release.
Distribution is unlimited.**

OAK RIDGE
National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

**ORNL/TM-2021/2268**

Nuclear Energy and Fuel Cycle Division

# NEAMS WORKBENCH WORKFLOWS

Robert A. Lefebvre
Marc-Olivier G. Delchini
Nicolas E. Stauff
Patrick C. Shriwise
Brandon R. Langley
Benjamin S. Collins

September 2021

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

# CONTENTS

# ABSTRACT

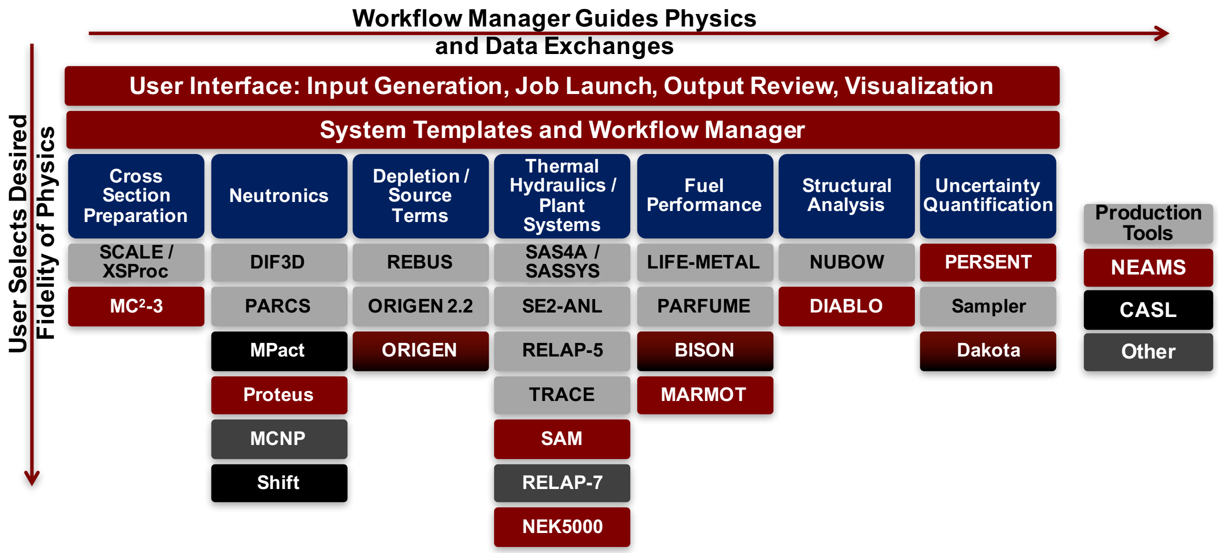
Modeling and simulation efforts frequently involve loosely coupled workflows. Automation of workflows, data exchanges, and simulation software execution is a productive and often desired feature or improvement for enhancing consistency of results and productivity of the analyst. It facilitates cross-collaboration of different analysts of multidisciplinary simulation by streamlining data exchange processes across local and high performance computing resources. This report proposes a high-level, extensible workflow module to enable consistent and efficient use of tools integrated into the Nuclear Energy Advanced Modeling and Simulation (NEAMS) Workbench with as little user intervention as possible.

## 1. INTRODUCTION

The Nuclear Energy Advanced Modeling and Simulation (NEAMS) Workbench provides a common analysis environment for executing NEAMS software [1], to include neutronics [2,3], thermal hydraulics [4], fuel analysis and performance [5], and associated model uncertainty quantification and optimization [6, 7, 8]. The workbench includes the open-source Workbench Analysis Sequence Processor (WASP) [9] toolkit for domain-specific language processing and workflows. Together, NEAMS Workbench and WASP provide a flexible platform from which additional analysis capabilities can be added, as demonstrated by the development of the Economics Dispatch Genetic AlgoRithms code, known as EDGAR [10]. The NEAMS Workbench and its Miniconda-managed Python runtime environment layer is well positioned to facilitate loosely coupled workflows on desktop machines and high-performance computing (HPC) platforms. The following sections describe the specific problem scope and propose high-level design for a workflow manager and its modules in the NEAMS Workbench.

## 2. PROBLEM SCOPE

The workflow module is intended to automate data transfers and consolidate input parameters into a concise workflow input file. User input errors are significantly reduced when consolidating parameters and automating data exchanges. Furthermore, the workflow module can optimize execution by managing concurrent branches of the workflow. Specifically, the workflow module must facilitate data and loosely coupled application execution flow in manner that preserves transparency of operation and maximizes efficiency for the user and the user's workflow. Figure 1 is excerpted from the introductory Workbench paper [11], and it highlights the original high-level vision that the proposed workflow module intends to realize.


Workflow Manager Guides Physics and Data Exchanges
User Selects Desired Fidelity of Physics
User Interface: Input Generation, Job Launch, Output Review, Visualization
System Templates and Workflow Manager

| Cross Section Preparation | Neutronics | Depletion / Source Terms | Thermal Hydraulics / Plant Systems | Fuel Performance | Structural Analysis | Uncertainty Quantification | Production Tools |
|---|---|---|---|---|---|---|---|
| SCALE / XSProc | DIF3D | REBUS | SAS4A / SASSYS | LIFE-METAL | NUBOW | PERSENT | NEAMS |
| MC$^2$-3 | PARCS | ORIGEN 2.2 | SE2-ANL | PARFUME | DIABLO | Sampler | CASL |
| | MPact | ORIGEN | RELAP-5 | BISON | | Dakota | Other |
| | Proteus | | TRACE | MARMOT | | | |
| | MCNP | | SAM | | | | |
| | Shift | | RELAP-7 | | | | |
| | | | NEK5000 | | | | |

**Figure 1. Original conceptual design of the workflow execution and data exchanges.**

A hypothetical workflow might involve (1) a mesh-based neutronics code, (2) a mesh generator, and (3) a nuclear cross section data generator. Many of these workflow steps can be completed by alternate codes that the user finds preferable. The workflow module must provide a classification of modules that facilitates overriding code-specific parameters. This design element will serve to consolidate class-specific parameters and will reduce the burden on module developers as they subclass a workflow component for a specific application. Figure 2 illustrates the data flow for the example workflow involving cross sections, meshing, and neutronics analysis.

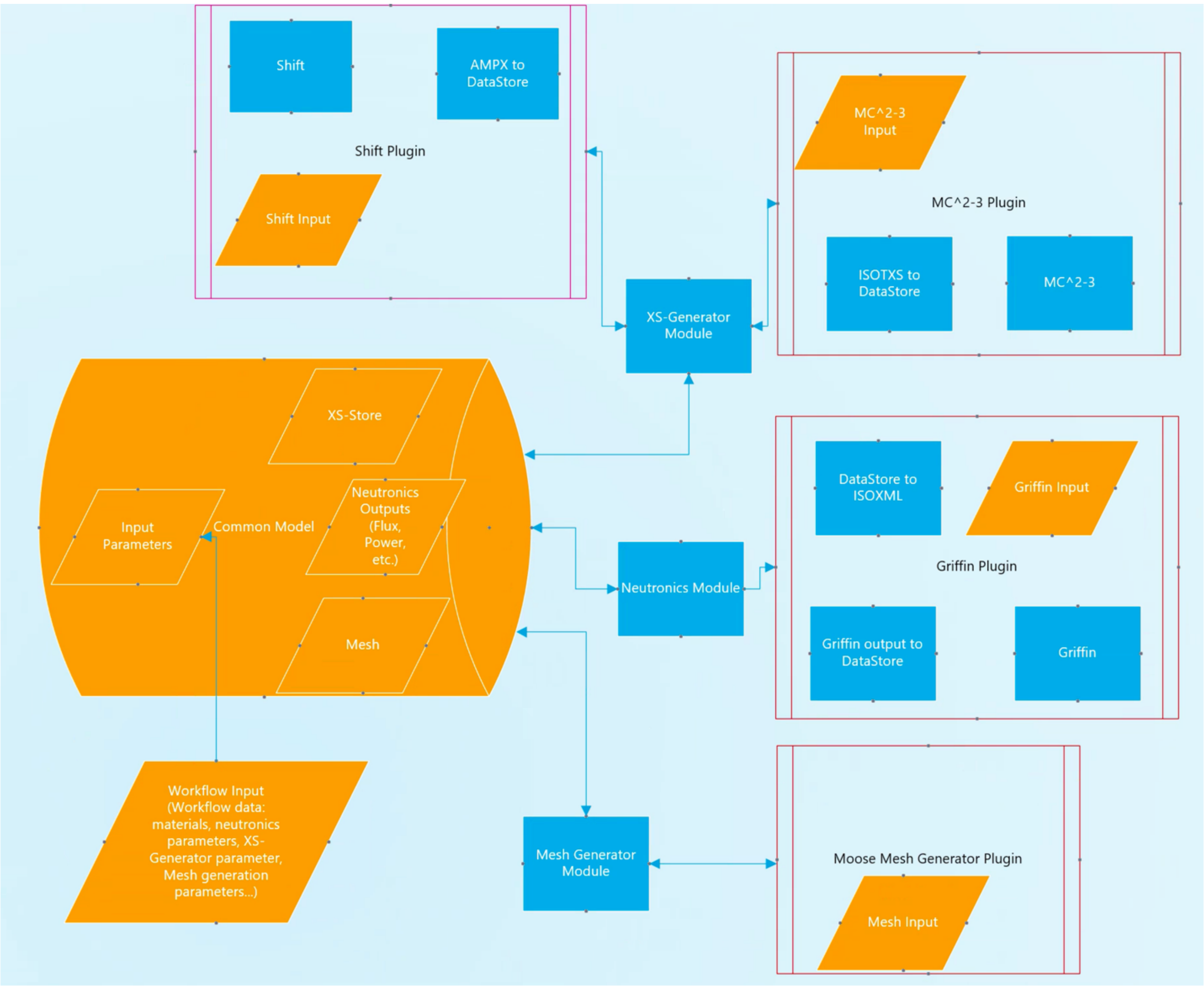


**Figure 2. Example dataflow for cross-section, mesh, and neutronics.**

The workflow input will describe the fundamental workflow parameters needed by the incorporated workflow modules. These parameters will be captured in a common model from which each module can retrieve inputs and store outputs. Figure 2 also shows that there are two cross section (XS) generator module plugins, highlighting an important aspect of the design, which is to facilitate reuse of class-specific common logic. The MC$^2$-3 code available from Argonne National Laboratory and the Shift code from Oak Ridge National Laboratory are both part of the NEAMS software suite. The XS-Generator module will retrieve and check necessary data from the common module, after which the application-specific plugin will format input files via templates or alternate serialization methods and execute the analysis. Once execution is completed, the module will capture the workflow-requested data artifacts and cache them in the common model for access by downstream modules. The workflow module responsible for general management must be able to execute the applications as directed by the workflow parameters. For example, serial execution of the modules may be desired because of computational resource limitations. Alternatively, when opportunities for concurrent module execution exist, the workflow manager will reduce execution time via concurrent invocation where possible. Scenarios involving execution of the workflow, potentially in part, on remote computing resources must also be supported.

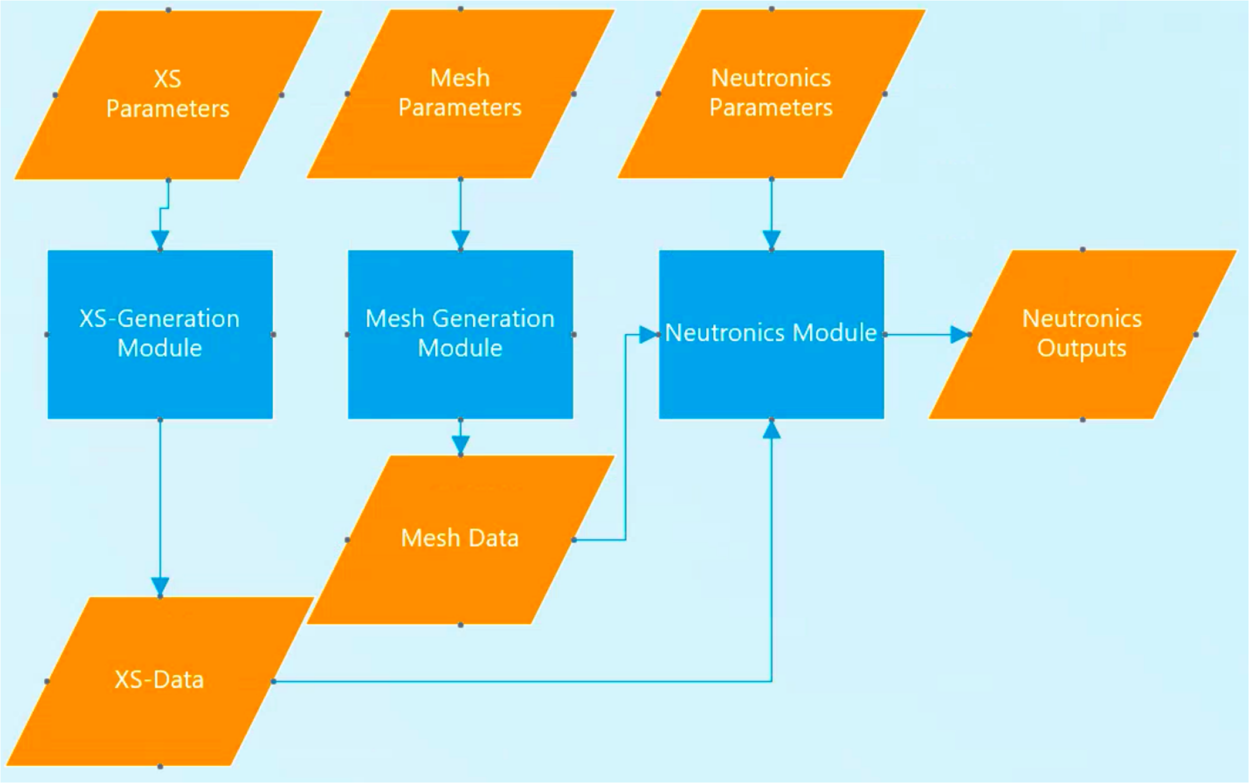


**Figure 3. Example serial execution workflow for cross-section, mesh, and neutronics.**

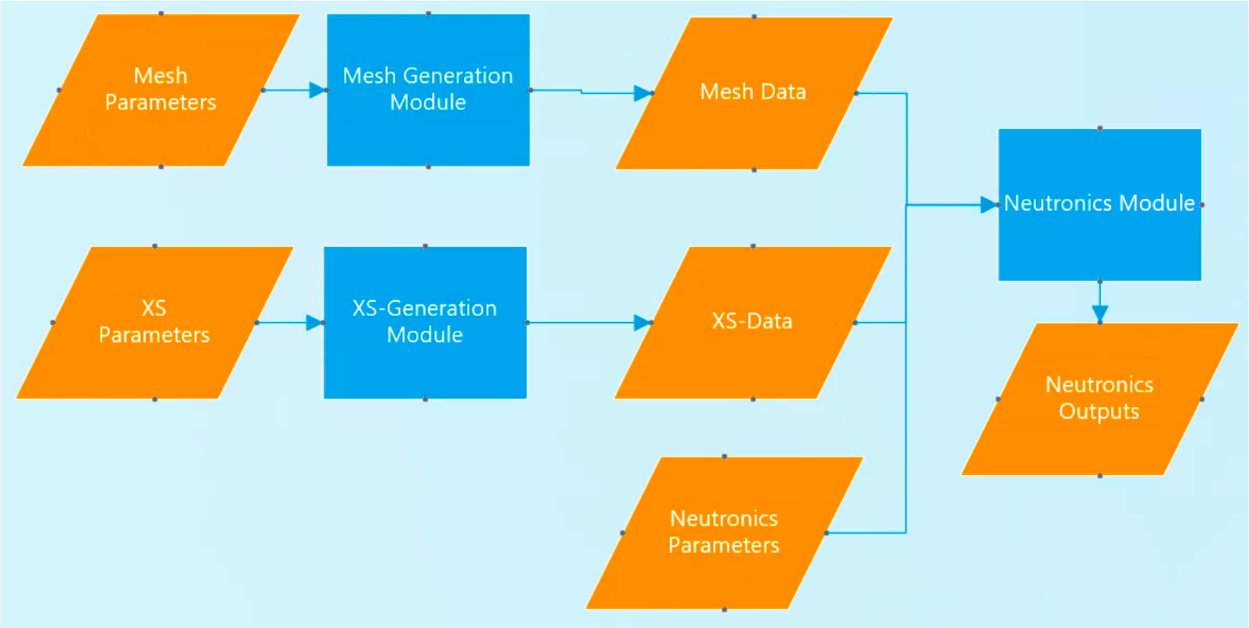


**Figure 4. Example concurrent execution workflow for cross-section, mesh, and neutronics modules.**

In addition to data exchange and application execution, the NEAMS Workbench workflow module must convey plugin input requirements to the user through the workbench's interactive input editor. Therefore, the module plugin input requirements must be described in an intuitive, extensible, and accessible manner.

# 3. WORKFLOW SPECIFICATION

The workflow will be executed by the workflow manager, a Python interface that can use the WASP domain-specific language interpreter to process an ASCII workflow description. From the workflow's ASCII input or script instructions, the common model will be populated, modules will be invoked in the order requested, and a summary of results will be presented.

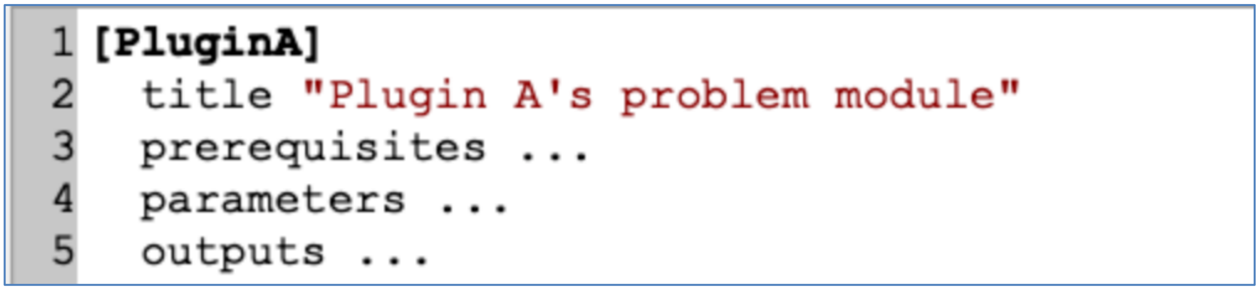
## 3.1 WORKFLOW INPUT

The NEAMS Workbench can provide an integrated development environment for application-specific inputs. These features include content assistance (input autocompletion and navigation), diagnostics (input checking and suggestion), and templating. While the workflow can be scripted via the Python workflow manager and associated classes, an ASCII domain-specific input provides a solution that can be more easily understood and efficiently used, especially for users without scripting experience.

The workflow input must manage the following application components:

- Plugin-specific parameters used in the application's analysis
- Plugin-specific outputs that are artifacts requested by the user and needed by downstream modules as prerequisites of their execution
- Plugin-specific prerequisites that must be fulfilled for successful execution

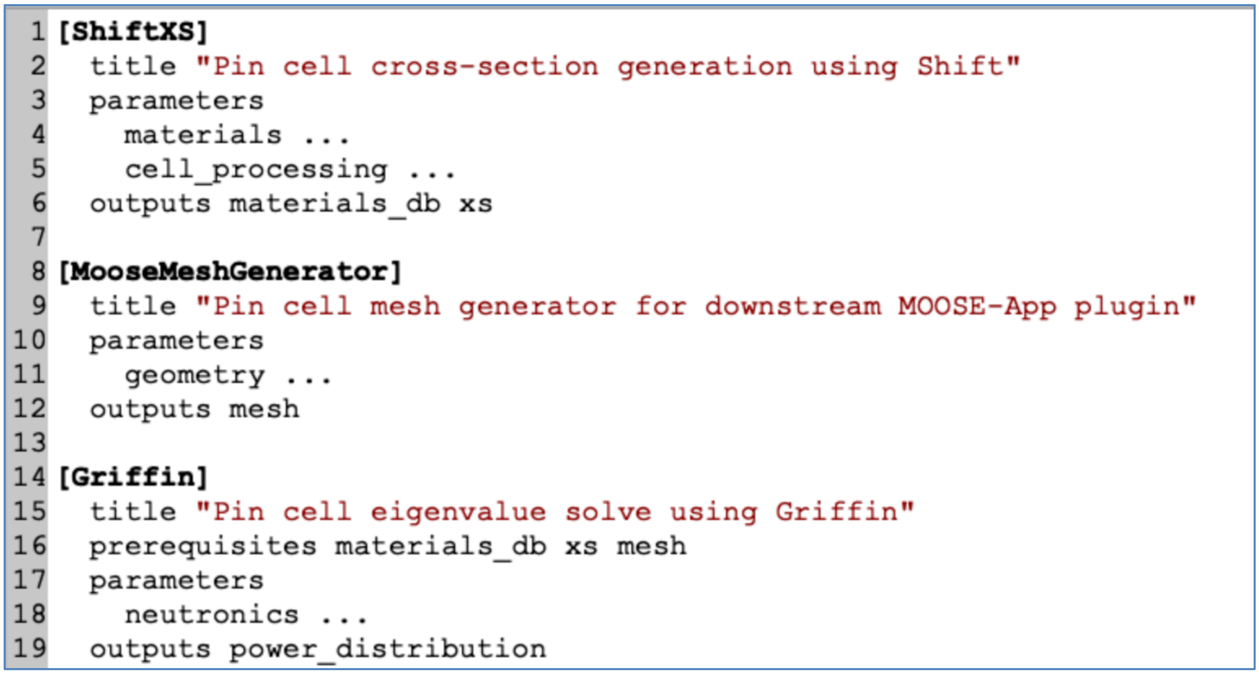
Figure 5 illustrates an example snippet of a single workflow plugin's ASCII input.



```
[PluginA]
  title "Plugin A's problem module"
  prerequisites ...
  parameters ...
  outputs ...
```

**Figure 5. Example workflow module plugin input sections.**

The optional prerequisites section lists components created elsewhere, managed by the common model, and required by the plugin. The parameters section provides a location for entry of application input options. These parameters can be as explicit as desired by the plugin developer or as simple as a filepath to a working application input. The outputs list the artifacts that the plugin must generate for use by downstream plugins. Figure 6 illustrates an example ASCII input used for the example workflow described in Section 2.



```
[ShiftXS]
  title "Pin cell cross-section generation using Shift"
  parameters
    materials ...
    cell_processing ...
  outputs materials_db xs

[MooseMeshGenerator]
  title "Pin cell mesh generator for downstream MOOSE-App plugin"
  parameters
    geometry ...
  outputs mesh

[Griffin]
  title "Pin cell eigenvalue solve using Griffin"
  prerequisites materials_db xs mesh
  parameters
    neutronics ...
  outputs power_distribution
```

**Figure 6. Example workflow for cross section, mesh, and neutronics.**

# 4. IMPLEMENTATION PLAN

The proposed workflow requires Python modules for the common model and workflow manager and interface classes for each physics code, along with plugin classes that realize each integrated application. Additionally, utility modules will be needed to assist in data transformations. For example, code A generates an artifact with format X, but the subsequent code B requires the artifact with format Y. A utility module will exist to convert the class of artifact (e.g., cross section data) from the X to the Y format.

The implementation will occur in the following phases.

## 4.1 PHASE 1 - PROTOTYPE

The initial implementation scope will serve to enable common analysis workflows referencing existing application inputs. The initial `CommonModel` and `WorkflowManager` classes will be defined, as well as the workflow input schema. A pilot workflow will be chosen in consultation with NEAMS teams, and the appropriate interface, physics, and data transfer classes will be authored. Phase 1 will conclude with a demonstration of the pilot workflow operating standalone and within the NEAMS Workbench on a desktop machine.

## 4.2 PHASE 2 – INITIAL IMPLEMENTATION

Phase 2 will seek additional workflows to test the Phase 1 implementation. In parallel, updates will be made that will enable physics plugins to communicate parameters to the workflow input. These parameters will be used in the application-specific input and will allow workflow authors to reuse nominal analysis inputs for greater effect. The plugin parameters system must be robust enough to handle a potential 1 × 1 mapping of the native application input and the plugin parameter input, a problem made tractable using programmatically accessible input schemas used by workbench-integrated codes.

### 4.3 PHASE 3 - ADDITIONAL CAPABILITIES AND FEEDBACK

Phase 3 will aim to mature the implementation and increase the number of workflow capabilities (plugins, utilities, accessible parameters). The implementation will receive an interface usability review, and any necessary changes will be supported by published documentation to facilitate ease of extension by external collaborators. The ability to run part or all of a workflow on remote computing resources will be added. Finally, the NEAMS Workbench Dakota integration, which provides sensitivity analysis (SA), uncertainty quantification (UQ), and optimization capabilities [12], will be demonstrated, and any required changes will be made to facilitate streamlined user analysis of SA/UQ and optimization.

### 4.4 PHASE 4 – INTERACTIVE FLOW DIAGRAM USER INTERFACE

Once Phase 3 is complete and the interfaces are established, a graphical user interface (GUI) will be developed to support a more interactive flow diagramming experience. Graphical widgets will present the physics and utility modules and will illustrate the relationship between their input and outputs. The GUI has the added benefit of allowing the user to automatically visualize the workflow described in the ASCII workflow input. Fundamental operations of the flow diagram user interface are as follows:

1. Create and delete modules
2. Add and remove data interconnects
3. Edit module parameters
4. Arrange module position on the diagram canvas for purposes of visual illustration

## 5. SUMMARY

This milestone report proposes the high-level requirements and implementation plan for NEAMS Workbench workflows. Through collaboration with teams within the NEAMS campaign and users abroad, the workflow components will be developed and improved to realize additional application plugins with more complex input parameters. Utilities for managing data exchanges and the accompanying input interface will facilitate users' loosely coupled workflows.

## 6. REFERENCES

1. Lefebvre, R. A., Langley, B. R., Miller, L. P., Delchini, M. G., Baird, M., and Lefebvre, J. P. *NEAMS Workbench Status and Capabilities*. ORNL/TM-2019/1314. Oak Ridge, TN: Oak Ridge National Laboratory, 2019. https://www.doi.org/10.2172/1570117.

2. N. Stauff "Integration of PyARC/Workbench with New Fast Reactor Modeling and Simulation Capabilities," ANL/NEAMS-20/2, September 30, 2020.

3. Stauff, N., Lee, C., Shriwise, P., Miao, Y., Hu, R., Vegendla, P., and Fei, T. *Neutronic Design and Analysis of the Holos-Quad Concept*. ANL/NSE-19/8. Argonne, IL: Argonne National Laboratory, 2019. https://doi.org/10.2172/1524786.

4. Fan, Y., Delchini, M. G., and Lefebvre, R. A. "Verification of Nek4nuc (Nek5000 Integrated in NEAMS Workbench) via Turbulent Pipe Flow Simulation." Presented at the ANS Winter Meeting and Nuclear Technology Expo, Chicago, IL, November 2020. https://www.osti.gov/servlets/purl/1731054

5. Cunningham, K. M., Powers, J. J., and Lefebvre, R. A. *Modeling the IFR-1 Experiment: A BISON Metallic Fuel Benchmark*. ORNL/TM-2019/1270. Oak Ridge, TN: Oak Ridge National Laboratory, 2019. https://doi.org/10.2172/1649544.

6. Zeng, K., Stauff, N. E., Hou, J., and Kim, T. K. "Development of Multi-Objective Core Optimization Framework and Application to Sodium-Cooled Fast Test Reactors." *Progress in Nuclear Energy* 120 (February 2020). https://doi.org/10.1016/j.pnucene.2019.103184.

7. Stauff, N., Maronati, G., Ponciroli, R., Ganda, F., Kim, T., Taiwo, T., Cuadra, A, Todosow, M., Talbot, P., Rabiti, C., Dixon, B., and Kim, S. *Daily Market Analysis Capability and Results*. ANL/NSE-19/5. Argonne, IL: Argonne National Laboratory, 2019. https://doi.org/10.2172/1511150.

8. Delchini, Marc-Olivier G., Swiler, Laura P., and Lefebvre, Robert A. "Extension of the NEAMS Workbench to Parallel Sensitivity and Uncertainty Analysis of Thermal Hydraulic Parameters Using Dakota and Nek5000." Korea, Republic of: N. p., 2021. Web. https://doi.org/10.1016/j.net.2021.04.005.

9. Lefebvre, Robert Alexander, Langley, Brandon R., and Lefebvre, Jordan P. *Workbench Analysis Sequence Processor*. United States: N. p., 2017. Web. doi:10.2172/1524895.

10. Stauff, N., Maronati, G., Ponciroli, R., Ganda, F., Kim, T., Taiwo, T., Cuadra, A, Todosow, M., Talbot, P., Rabiti, C., Dixon, B., and Kim, S. "Daily Market Analysis Capability and Results." ANL/NSE-19/5. Argonne, IL: Argonne National Laboratory, 2019. https://doi.org/10.2172/1511150.

11. B.T. Rearden, R.A. Lefebvre, A.B. Thompson, B.R. Langley, and N.E. Stauff, "Introduction to the Nuclear Energy Advanced Modeling and Simulation Workbench," M&C 2017 – International Conference on Mathematics & Computational Methods Applied to Nuclear Science and Engineering, Jeju, Korea, April 16–20, 2017.

12. Swiler, Laura Painton, Lefebvre, Robert A., Langley, Brandon R., and Thompson, Adam B. "Integration of Dakota into the NEAMS Workbench." United States: N. p., 2017. Web. doi:10.2172/1372616.