

Introduction to the Nuclear Energy Advanced Modeling and Simulation Workbench

Bradley T. Rearden*, Robert A. Lefebvre, Adam B. Thompson, and Brandon R. Langley

Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6170, USA, * reardenb@ornl.gov

Nicolas E. Stauff

Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439

Abstract - *The mission of the US Department of Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program is to develop, apply, deploy, and support state-of-the-art predictive modeling and simulation tools for the design and analysis of current and future nuclear energy systems using computing architectures from laptops to leadership-class facilities. The NEAMS Workbench is a new initiative that will facilitate the transition from conventional tools to high-fidelity tools by providing a common user interface for model creation, review, execution, output review, and visualization for integrated codes. The Workbench can use common user input that includes engineering scale specifications that are expanded into application-specific input requirements through the use of customizable templates. The templating process enables multi-fidelity analysis of a system from a common set of input data. Expansion of the codes and application templates available in the Workbench will broaden the NEAMS user community and will facilitate system analysis and design. Current and planned capabilities of the NEAMS Workbench are detailed here.*

I. INTRODUCTION

The mission of the US Department of Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program is to develop, apply, deploy, and support state-of-the-art predictive modeling and simulation tools for the design and analysis of current and future nuclear energy systems using computing architectures from laptops to leadership-class facilities. The tools in the NEAMS ToolKit will enable transformative scientific discovery and insights otherwise not attainable or affordable and will accelerate the solutions to existing problems as well as the deployment of new designs for current and advanced reactors. These tools will be applied to solve problems identified as significant by industry and consequently will expand validation, application, and long-term utility of these advanced tools.

The NEAMS program is organized along three product lines: the Fuels Product Line (FPL), the Reactors Product Line (RPL) and the Integration Product Line (IPL).

NEAMS FPL and RPL provide many advanced tools, such as the BISON and MARMOT fuel performance tools based on the Multiphysics Object-Oriented Simulation Environment (MOOSE) from Idaho National Laboratory [1, 2] and the PROTEUS neutronics code and the NEK5000 computational fluid dynamics code from the Simulation-based High-efficiency Advanced Reactor Prototyping (SHARP) tools from Argonne National Laboratory (ANL).[3] However, these advanced tools

often require large computational resources, can be difficult to install, and require expert knowledge to operate, causing many analysts to continue to use traditional tools instead of exploring high-fidelity simulations.

The NEAMS IPL is responding to the needs of design and analysis communities by integrating robust multiphysics capabilities and current production tools in an easy-to-use common analysis environment that enables end users to apply high-fidelity simulations to inform lower-order models for the design, analysis, and licensing of advanced nuclear systems.

The NEAMS Workbench is a new initiative that will facilitate the transition from conventional tools to high-fidelity tools by providing a common user interface for model creation, review, execution, output review, and visualization for integrated codes. The Workbench can use common user input that includes engineering scale specifications that are expanded into application-specific input requirements through the use of customizable templates. The templating process enables multi-fidelity analysis of a system from a common set of input data. Expansion of the codes and application templates available in the Workbench will broaden the NEAMS user community and will facilitate system analysis and design. Users of the Workbench will still need to license and install the desired computational tools, but it will provide a more consistent user experience and ease the transition from one tool to the next.

II. MULTI-FIDELITY MULTIPHYSICS

The NEAMS Workbench will enable analysts to select the fidelity of each type of physics for use in the simulation, choosing from a variety of tools integrated from many projects and code teams. A conceptual design of tools that could be integrated for advanced reactor analysis is shown in Fig. 1. The current production tools with advanced components supported by the NEAMS program such as those from the Argonne Reactor Codes (ARC) [4] and the SCALE Code System [5] are shown in light gray, MCNP [6] and other production tools are shown in dark grey, tools from the NEAMS ToolKit are shown in maroon, and tools from Consortium for Advanced Simulation of Light Water Reactors [7] are shown in black. With the Workbench, it will be possible to use the same fundamental engineering input data to create code-specific input models for each tool, enabling analysts to learn more about specific phenomena by performing reference high-fidelity analyses to confirm approximations or assumptions in fast-running lower-order design calculations.

Multiphysics capabilities, such as the SHARP and MOOSE tools, will continue to provide fully coupled capabilities under the Workbench. For traditional single physics codes that form a multiphysics suite with coupling through manipulation of the output of one code to serve as input for the next, translation tools will be developed and integrated into the workflow manager to extract and format

the needed data in a manner that does not require intrusive changes to the physics codes themselves.

III. USER INTERFACE

The NEAMS Workbench will leverage the Fulcrum user interface, which was developed for 2016 release of SCALE 6.2. Fulcrum introduced several new concepts as an integrated user interface for nuclear analysis, builds on decades of user experience from thousands of users, and replaces eight user interfaces from the 2011 release of SCALE 6.1. Fulcrum is a cross-platform graphical user interface designed to create, edit, validate, and visualize input, output, and data files. Fulcrum directly connects the user with the text form of the input file while providing inline features to assist with building correct inputs. Fulcrum provides input editing and navigation, interactive geometry visualization, job execution, overlay of mesh results within a geometry view, and plotting of data from file formats. An error checker interactively identifies input errors with spelling mistakes or data entry omissions or duplications for all supported codes. The input validation engine identifies allowed data ranges and interdependencies in the input and reports inconsistencies to the user. The layout of panels in Fulcrum is highly configurable to accommodate the preferences of users, as shown in Fig. 2.

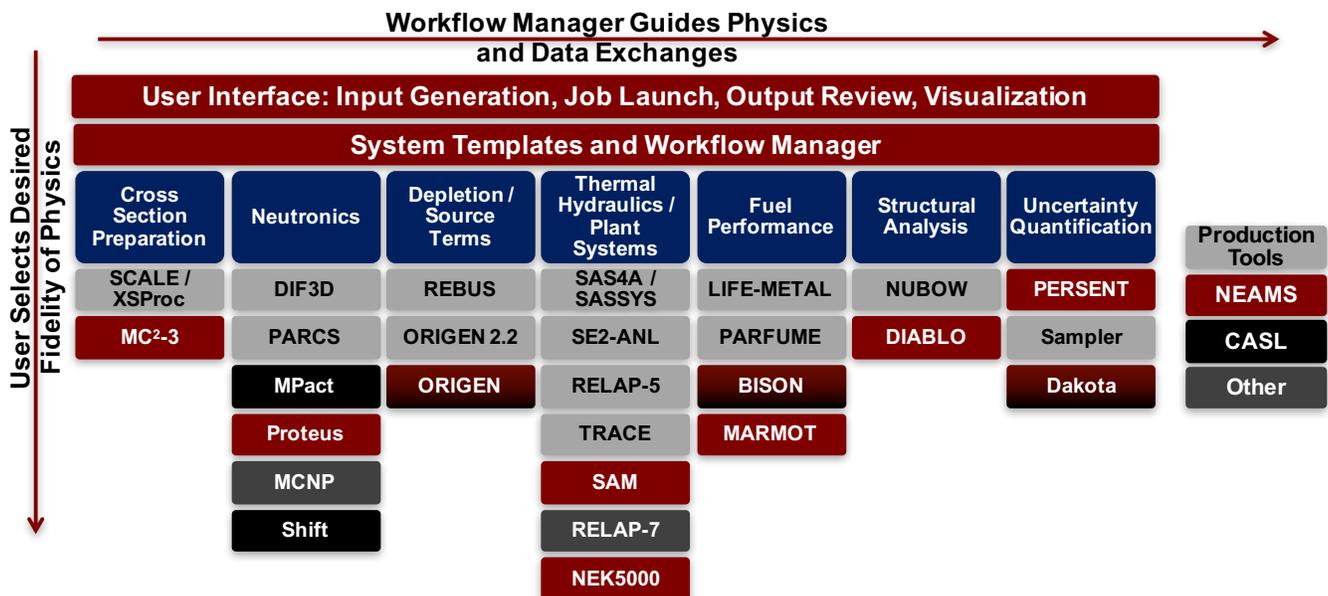


Fig. 1. Conceptual design of tools integrated in the NEAMS Workbench for advanced reactor analysis.

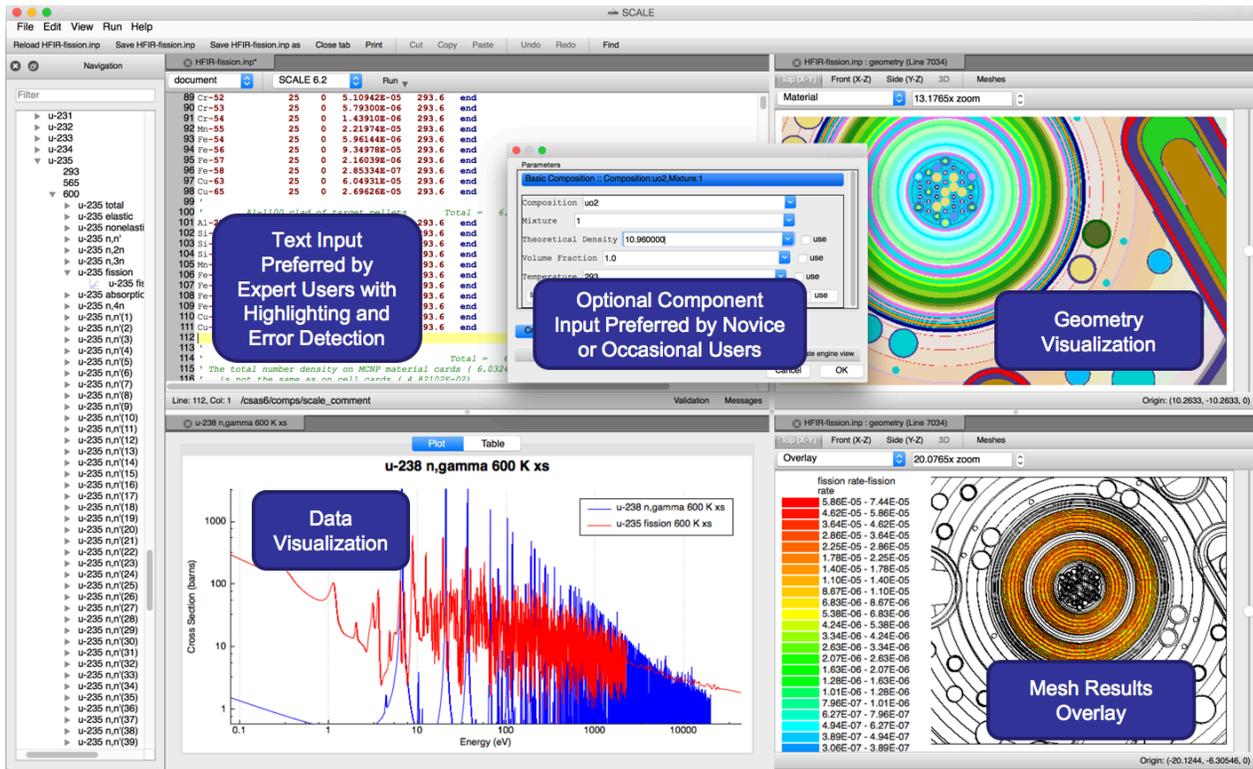


Fig. 2. The NEAMS Workbench leverages the Fulcrum user interface from SCALE.

The configuration files that define how each code input is structured, the requirements and interdependencies of each input block, and the types and content of the dialog boxes available for generated assisted input are all provided in text files that can be created and edited by any user of the Workbench. In this way, the members of any team can integrate their tools into the Workbench, whether they are production tools that are widely available, custom-developed analysis tools in a proprietary environment, or university-led prototypes for the exploration of new algorithms. Custom-developed configuration files can be contributed back to the Workbench developers for consideration for inclusion in the production version. An example of the interactive input error detection is shown for a SCALE input in Fig. 3, where a user inadvertently typed `uo2` (`u-zero-2`) instead of `uo2` to specify uranium dioxide as a material. The first error at the bottom of the screen demonstrates how Fulcrum is comparing the current input against the list of allowed values in this context and is recommending some alternatives to assist the user. A second error is shown where the user was specifying the uranium enrichment for this material. Here an input rule is applied that the weight percent values must sum to 100%. In this case, the user inadvertently entered a total of 101%, and Fulcrum identified that error as well.

Leveraging the configuration files for each code supported by the Workbench, the Fulcrum user interface

provides context-aware assistance in auto-completing the input, which can be useful when learning the intricacies of a new code, or when transitioning between many codes as is the intent of the Workbench. An example from SCALE is shown in Fig. 4. In the left image, the user inserts the cursor at the desired point in the input, presses the key combination of `ctrl-space`, and Fulcrum lists all available options in this context, geometry shapes in this example. Selecting a *configurable* option launches an additional interactive dialog, shown on the right, to provide additional assistance in populating this portion of the input.

The Fulcrum interface currently supports visualization capabilities from SCALE for Monte Carlo geometry with results overlay as shown in Fig. 5. Plotting capabilities are available for nuclear data and covariance data plotting as well as a variety of computed quantities using line plots, histograms, bar charts, and surface plots, as shown in Fig. 6 and Fig. 7. The capabilities of Fulcrum are being expanded beyond those required by SCALE, especially to support analysis with NEAMS finite element codes, by integrating visualization capabilities from VisIt [8] and ParaView [9].

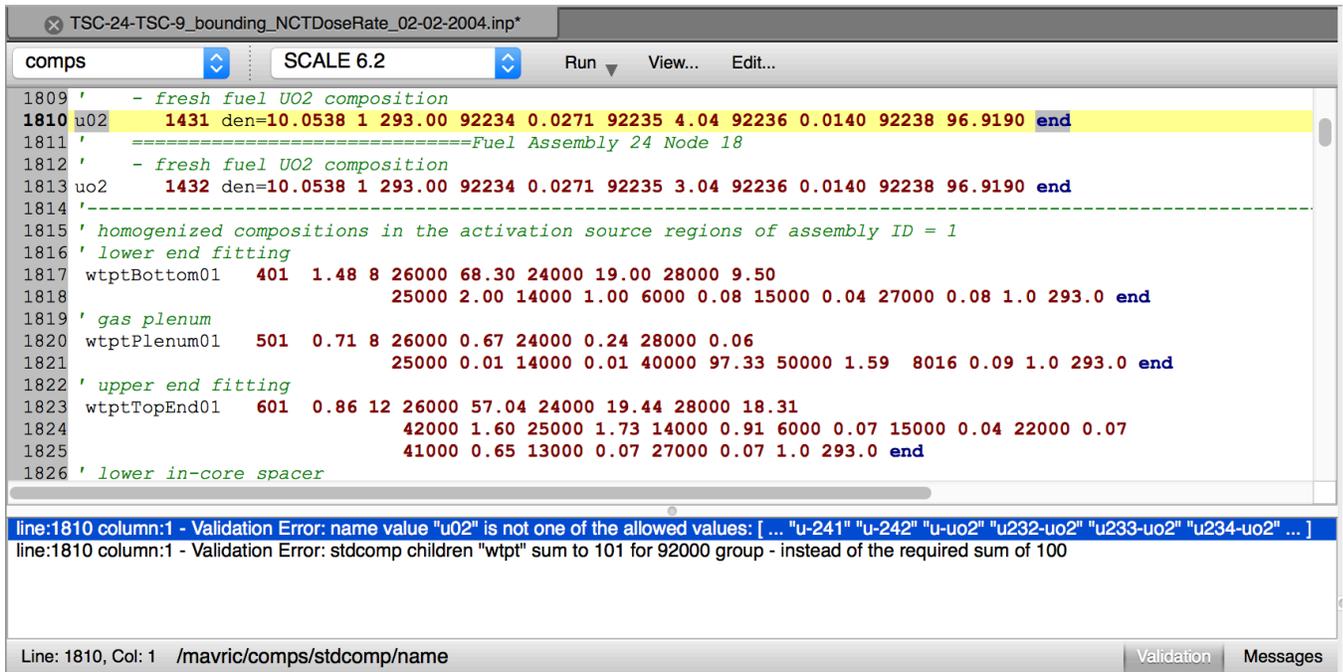


Fig. 3. Fulcrum user interface with interactive input error detection.

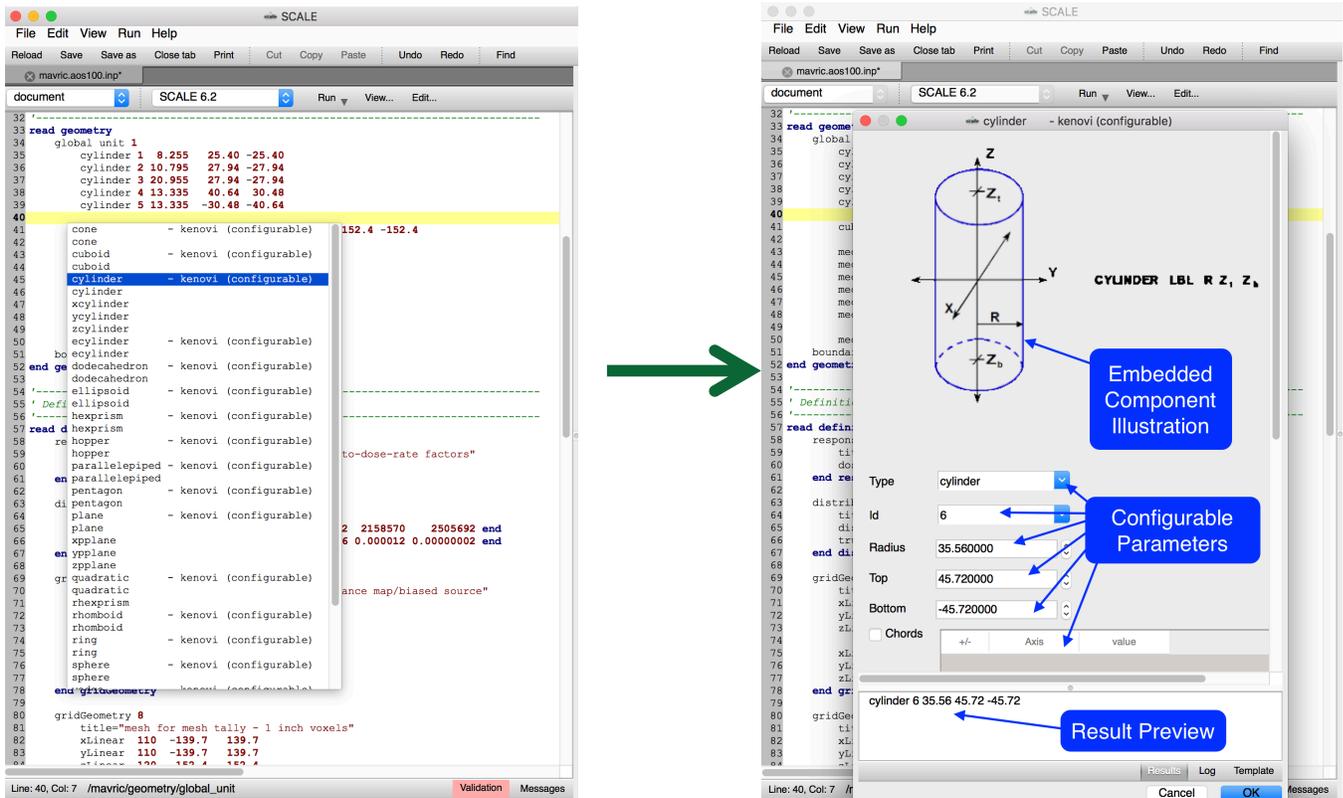


Fig. 4. Fulcrum auto-completion and forms-based input.

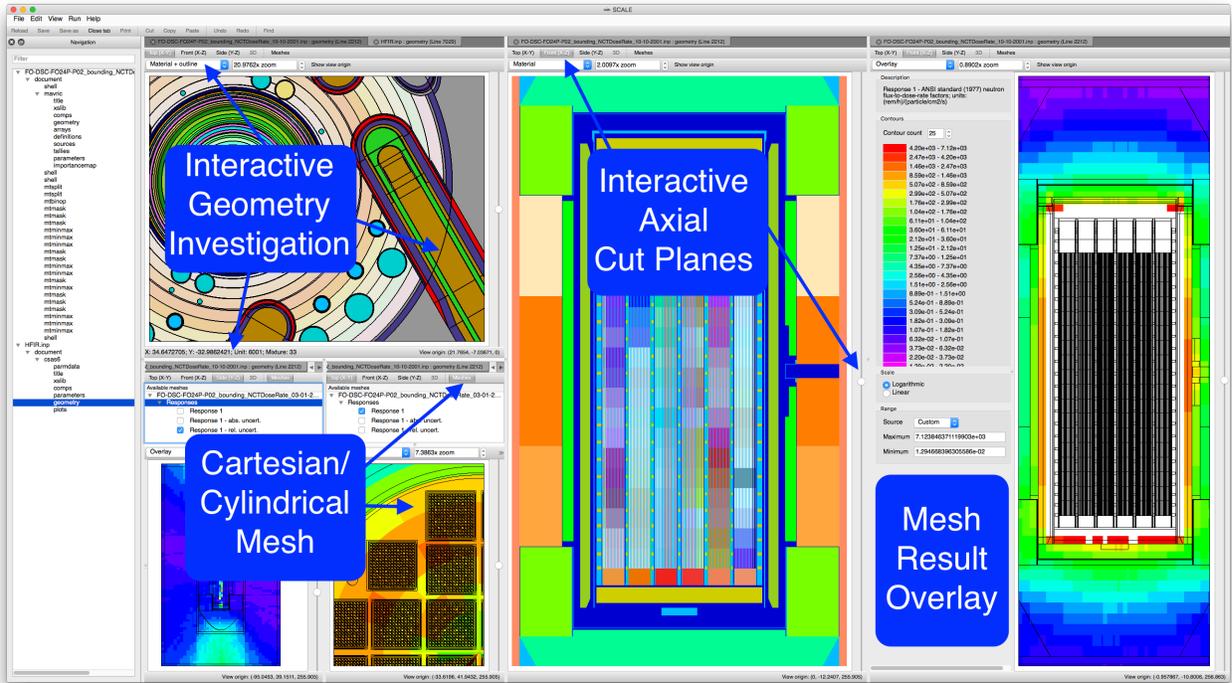


Fig. 5. Fulcrum geometry visualization capabilities.

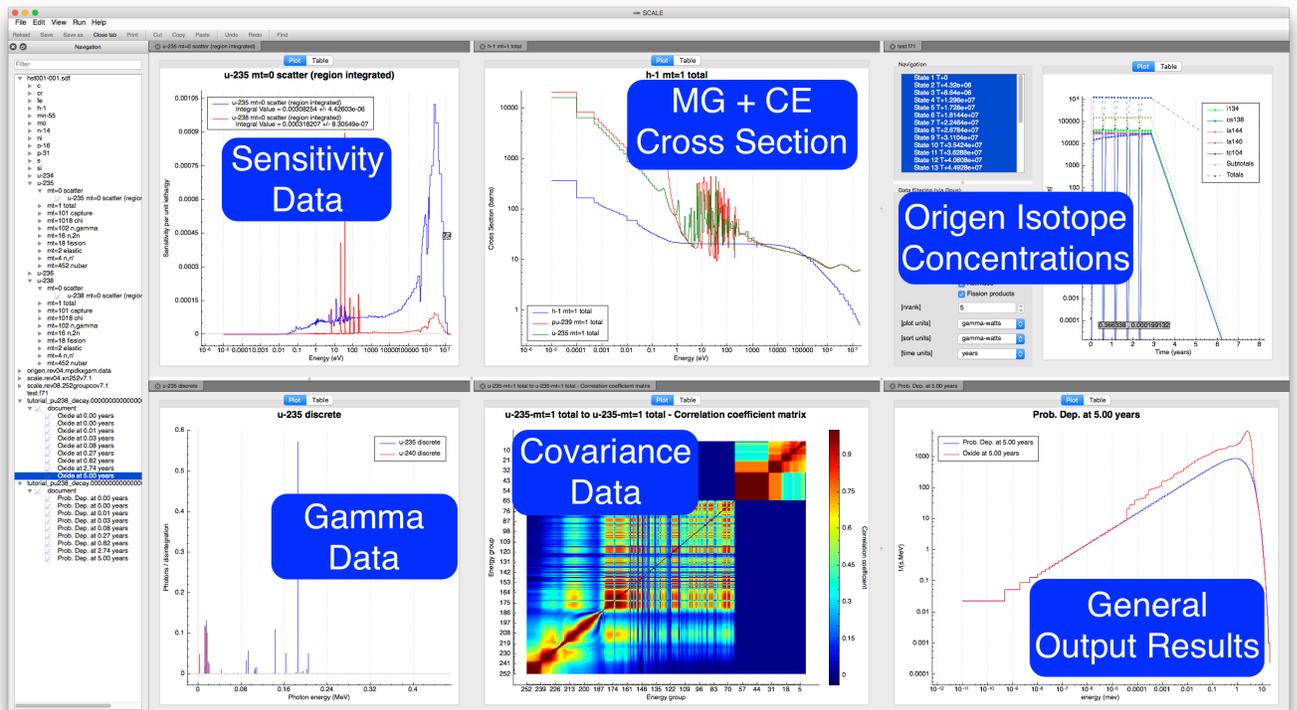


Fig. 6. Fulcrum plotting capabilities.

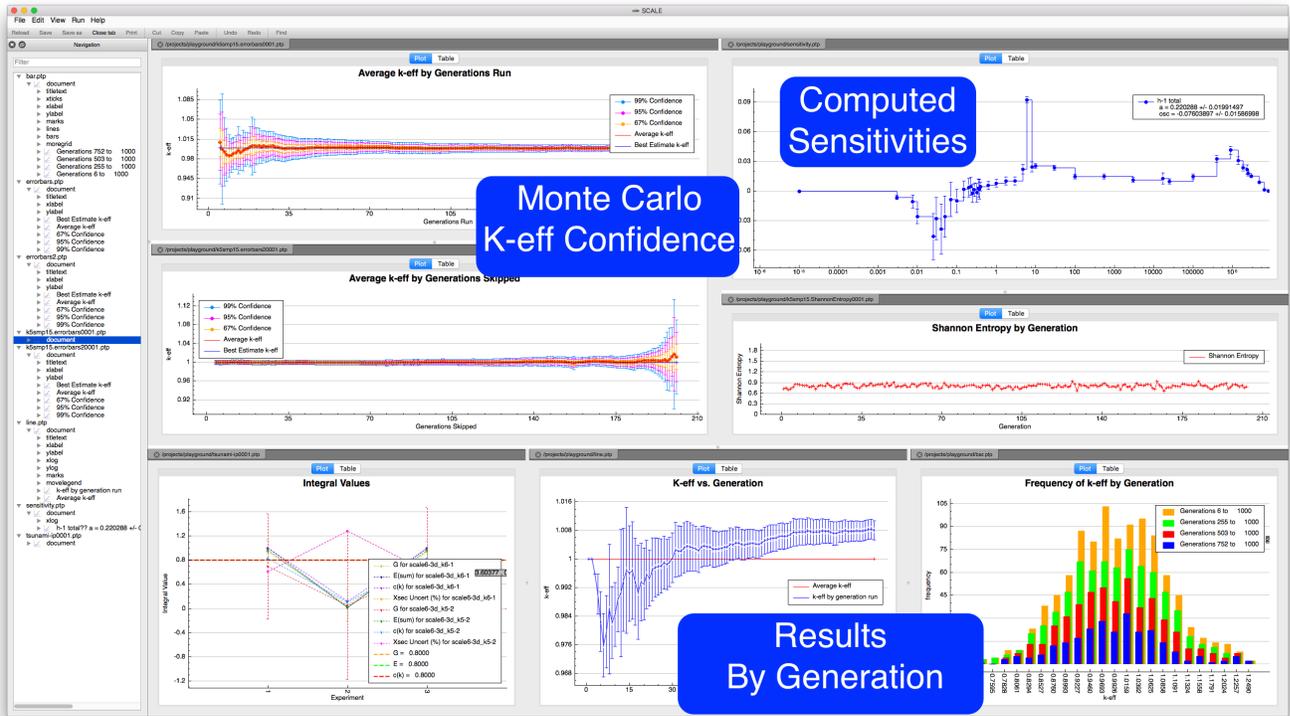


Fig. 7. Additional Fulcrum plotting capabilities.

IV. APPLICATION TEMPLATES

Beyond the Fulcrum user interface, the NEAMS Workbench expands common input parameters into code-specific detailed input for supported systems using the TemplateEngine developed for the Used Nuclear Fuel (UNF) Storage, Transportation & Disposal Analysis Resource and Data System (UNF-ST&DARDS), which has been successfully demonstrated for the detailed fuel depletion and decay, criticality safety, thermal hydraulics, and shielding analysis of thousands of UNF dry storage canisters using high-fidelity as-loaded fuel assembly and canister design data.[10] The purpose of templates is to allow for the creation of building blocks from which application-specific input is defined. The templates repository provides application-agnostic input generation. Templates allow the Workbench to utilize the same set of parameters to succinctly communicate to each application while decoupling Workbench from any specific application. The TemplateEngine processes and expands templates according to the contents and structure of a parameter set. The parameter set is defined as a structured collection of attributes, objects, and arrays of attributes. Objects and arrays can be embedded within one another and themselves. The syntax that represents a parameter set is that of JavaScript Object Notation (JSON). [11] JSON is a standard, lightweight data-interchange format; in which templates can easily be created and edited by the developers of the Workbench as well as the users. Where a

user wishes to represent a system that contains proprietary data, custom templates can be developed and stored on a secure server that can only be accessed by the site-specific installation of the Workbench.

A schematic of the NEAMS Workbench templating process is shown in Fig. 8, where the user provides engineering-style input for macro-level input parameters. The TemplateEngine accesses the archive of templates to build detailed inputs for many codes from the same input parameters. In a very simple example of a neutronics analysis of a UO_2 fuel pin, the input parameters such as enrichment, density, fuel dimensions, and pitch are specified. When using the ARC tools, the Workbench could create detailed inputs for cross-section generation with MC²-3 using explicit dimensions and then use the same information to volume-average the materials and create input for DIF3D. The same information could be applied for a high-fidelity simulation with SHARP by accessing a pregenerated finite element mesh geometry and using MC²-3 cross section data in a PROTEUS calculation. A reference continuous-energy Monte Carlo solution could also be generated with MCNP. In each case, analysts would have access to all tools integrated under the Workbench by leveraging templates created by experts in each tool. The detailed inputs and output for each code are available for quality assurance review, and comparisons of the results for each analysis can be performed to determine the adequacy of assumptions used in each type of analysis.

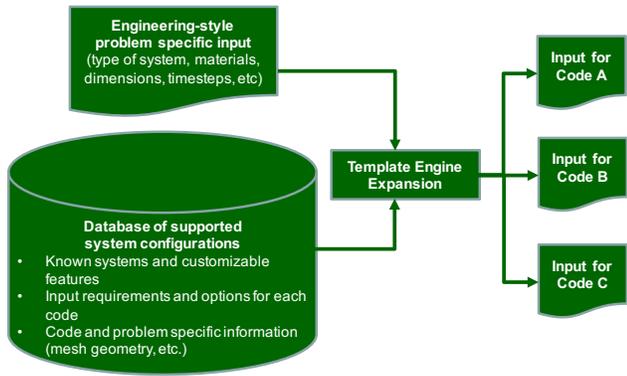


Fig. 8. Overview of TemplateEngine process for generating code-specific input for supported systems.

V. CODE INTEGRATION

Because the NEAMS Workbench leverages the Fulcrum user interface from SCALE, it already supports the dozens of verified and validated tools for criticality safety, reactor physics, radiation shielding, radioactive source term characterization, and sensitivity and uncertainty analysis available there. The BISON fuels performance code has been integrated [12], and work is ongoing to support the Dakota suite of iterative mathematical and statistical methods from Sandia National Laboratories that interface to many computational models. Additionally, the NEAMS Warthog tool, which couples

PROTEUS and BISON for multiphysics neutronics/fuels performance calculations, will also be integrated. [13]

A significant effort is under way to integrate the ARC codes from ANL into the NEAMS Workbench. In this work, a new common input is being developed using Standard Object Notation input format developed for SCALE 6.2. This common input will be used to populate templates to run calculations with MC²-3, DIF3D, REBUS, and PERSENT for integrated problem-dependent cross-section preparation, core analysis, depletion, and sensitivity/uncertainty analysis. Previously, each of these calculations required a separate input file, and many of the input formats were so difficult for users that a specialized script was often required to generate them. With the ARC/Workbench integration, users can easily create engineering style input and have the Workbench generate the input for the codes and launch the calculations using a customized runtime environment. A prototype ARC/Workbench input and associated workflow are shown in Fig. 9. Users who access the ARC codes through the Workbench will have easier access to advanced codes of NEAMS, which in the future will be able to leverage common input parameters.

The ongoing and near-term code integration activities are provided in TABLE I, where tools are separated into current production tools or advanced NEAMS tools, with several tools, ARC, SCALE, Dakota, and VisIt fitting in both categories because they are currently deployed tools in production use and are also partially supported by NEAMS for the development of advanced features.

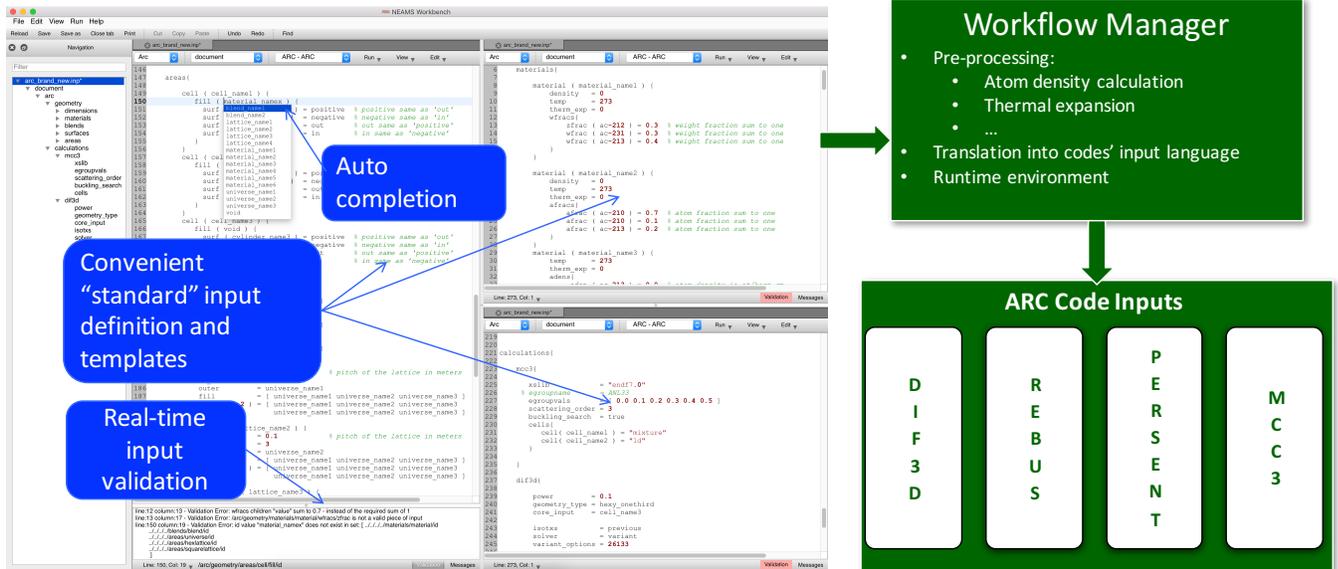


Fig. 9. Prototype ARC/Workbench integration and associated Workflow manager.

TABLE I. Ongoing and Near-Term Code Integration Plans for the NEAMS Workbench

Tool	Application	Production Tool	NEAMS Tool
BISON	Fuel performance		x
NEK5000	Thermal hydraulics		x
Warthog	Multiphysics neutronics and fuel performance		x
ARC	Fast reactor analysis	x	x
SCALE	General-purpose neutronics and shielding	x	x
Dakota	Uncertainty quantification and model optimization	x	x
VisIt	Mesh Visualization	x	x

VI. RUNTIME ENVIRONMENT

A mission of the NEAMS Workbench is to facilitate transition from conventional tools to high-fidelity tools. Many of the codes involve different means of invocation, and some have multiple means of invocation. A generic runtime environment interface was created to facilitate their execution. The intent of the runtime is to provide a consistent interface by which the Workbench can interact with each tool in all necessary modes of operation (e.g., serial, parallel, and scheduled execution).

Some tools have no runtime environment, requiring the user to manually conduct all steps associated with running them. For example, the user might be required to copy the problem input file into a specific location with a specific file name, input, and then invoke the application, thus producing temporary files and output file(s). That approach is prone to error and should not be important to the user or to Workbench as the logic is application specific. The solution is to contribute a new runtime script to the runtime repository, where the script overrides the setup, execution, or finalization logic as needed to fulfill the runtime interface. The setup logic might create a working directory, TMPDIR, and then copy the `problem.inp` into the TMPDIR as `TMPDIR/input`. The execution might invoke the application executable, passing application messages back to the calling application (e.g., command console, Workbench). The finalized logic might (1) combine the output files located in TMPDIR into logical order, (2) copy the output back into `problem.out`, residing next to `problem.inp`, and (3) delete the TMPDIR to clean up after itself.

Whatever the specific application logic is, a Workbench script will be available to allow consistent invocation and potentially to provide great convenience for the application's typical command-line user. As the runtime environment matures and additional features (e.g.,

a queuing system) are added for the base class, all incorporated runtimes will benefit.

VII. CONCLUSIONS

The NEAMS Workbench is a new initiative created in response to the needs of design and analysis communities to enable end users to apply high-fidelity simulations to inform lower-order models for the design, analysis, and licensing of advanced nuclear systems.

The NEAMS Workbench will facilitate the transition from conventional tools to high-fidelity tools by providing a common user interface and common user input by templating engineering-scale specifications to application-specific input requirements. The expansion of codes and system templates integrated under the Workbench will broaden the NEAMS tools user community and will facilitate system analysis and design.

ACKNOWLEDGMENTS

This research was sponsored by the U.S. Department of Energy Nuclear Energy Advanced Modeling and Simulation Program. Argonne National Laboratory's work was supported by U.S. Department of Energy (DOE) under Contract number DE-AC02-06CH11357.

REFERENCES

1. D. GASTON, C. NEWMAN, G. HANSEN, AND D. LEBRUN - GRANDIÉ, "MOOSE: A parallel computational framework for coupled systems of nonlinear equations, *Nucl. Eng. Des.*, **239**, 1768–1778 (2009).
2. R. L. WILLIAMSON, J. D. HALES, S. R. NOVASCONI, M. R. TONKS, D. R. GASTON, C. J. PERMANN, D. ANDERS, AND R. C. MARTINEAU, "Multidimensional multiphysics simulation of nuclear fuel behavior," *J. Nucl. Mater.*, **423** 149–163 (2012).
3. Y. Q. YU, E. R. SHEMON, J. W. THOMAS, VIJAY S. MAHADEVAN, RONALD O. RAHAMAN, AND JEROME SOLBERG, *SHARP User Manual*, ANL-NE-16/6, Argonne National Laboratory, Argonne, Illinois (2016).
4. *ARC 11.0: Code System for Analysis of Nuclear Reactors*, Argonne National Laboratory (2014). Available from Available from Radiation Safety Information Computational Center as CCC-824.
5. B. T. REARDEN AND M. A. JESSEE, EDS., *SCALE Code System*, ORNL/TM-2005/39, Version 6.2, UT-Battelle, LLC, Oak Ridge National Laboratory (2016). Available from Radiation Safety Information Computational Center as CCC-834.
6. D. B. PELOWITZ, ed., *MCNP6 User's Manual, Version 1.0*, Los Alamos National Laboratory report

- LA-CP-13-00634, Rev. 0 (May 2013). Available from Radiation Safety Information Computational Center as CCC-810.
7. JOHN A. TURNER, KEVIN CLARNO, MATT SIEGER, ROSCOE BARTLETT, BENJAMIN COLLINS, ROGER PAWLOWSKI, RODNEY SCHMIDT, AND RANDALL SUMMERS, "The Virtual Environment for Reactor Applications (VERA): Design and Architecture," *J. of Comput. Phys.*, **326**, 544–568 (2016).
 8. LLNL: VisIT Visualization Tool (2002–2016). <https://wci.llnl.gov/codes/visit>
 9. Kitware: ParaView (2002–2016). <http://www.paraview.org>
 10. R. A. LEFEBVRE, J. M. SCAGLIONE, J. L. PETERSON, P. MILLER, G. RADULESCU, K. BANERJEE, K. R. ROBB, A. B. THOMPSON, AND J. P. LEFEBVRE, "Development of Streamlined Nuclear Safety Analyses for Used Nuclear Fuel Applications," Submitted to *Nucl. Technol.* (2015).
 11. ECMA-International, *The JSON Data Interchange Format*, ECMA-404, CH-1204 Geneva, October 2013.
 12. R. A. LEFEBVRE, B. R. LANGLEY, AND A. B. THOMPSON, *Report on NEAMS Workbench Support for MOOSE Applications*, ORNL/TM-2016/572, UT-Battelle, LLC, Oak Ridge National Laboratory (2016).
 13. S. W. D. HART, *WARTHOG: Progress on Coupling BISON and PROTEUS*, ORNL/TM-2106/579, UT-Battelle, LLC, Oak Ridge National Laboratory (2016).