

# Dynamic Modeling of Small Modular Nuclear Reactors using MoDSim

Richard Hale<sup>1</sup>

Sacit Cetiner<sup>1</sup>

David Fugate<sup>1</sup>

Lou Qualls<sup>1</sup>

John Batteh<sup>2</sup>

Michael Tiller<sup>3</sup>

<sup>1</sup>Oak Ridge National Labs, Oak Ridge, TN USA

<sup>2</sup>Modelon, Inc., Ann Arbor, MI USA

<sup>3</sup>Xogeny, Inc., Canton, MI USA

halere1@ornl.gov

cetinerms@ornl.gov

fugatedl@ornl.gov

quallsal@ornl.gov

john.batteh@modelon.com

michael.tiller@xogeny.com

## Abstract

As part of the advanced small modular nuclear reactor (AdvSMR) R&D program, Oak Ridge National Laboratory (ORNL) is developing a Dynamic System Modeling Tool (MoDSim) to facilitate research and development related primarily to instrumentation and controls (I&C) studies of small modular reactors (SMRs).

The primary objective is to produce a demonstration product of a dynamic system modeling tool for SMRs. Functional Mockup Interface (FMI) has been used to allow the development of scoping models for non-Modelica users. This tool includes a web-based interface using Xogeny's FMQ platform for model configuration with local application deployment for simulation using FMI Add-in for Excel from Modelon; this toolchain is designed to allow plug and play access for users of various skill levels. The web-based interface allows true web-based access and solutions without requiring local applications. The initial installation of this tool has been tested on a liquid-metal small modular reactor (ALMR) concept modeled using Dymola and exported via FMI. This tool allows simulation to be performed within Excel without expertise in the native simulation language (Modelica) or model development and simulation environment (Dymola). This toolchain fulfills the Department of energy (DOE) project scope goal of developing a tool "*in a common and familiar environment to support a range of research activities requiring dynamic behavior simulation, modeling tools with easily re-configurable modules that reduce data input to typically available system level plant data*".

*Keywords: nuclear reactors; thermofluid systems; Dymola; Excel; web-based simulations; Advanced Liquid Metal Reactor.*

## 1 Introduction

Small modular nuclear reactors (SMRs) are modular nuclear power plants that are smaller (300 MWe or less) than current-generation base load plants (nominally 1,000 MWe or larger). SMR designs may include factory-fabricated reactors that can be transported by truck or rail to a nuclear power plant site. Small modular reactors offer the advantages of lower initial capital investment, intrinsic safety, scalability, and siting flexibility at locations unable to accommodate larger reactors. In addition, SMRs offer the potential to construct and operate a first reactor and later add more reactor modules at the same site (phased construction). This approach offers economic advantages but could generate numerous configurations of reactor modules, power conversion, and heat sink that will require unique capabilities from control and support systems.

The objectives of this project are to establish a configurable framework for the development of a dynamic simulation environment for SMRs using pre-developed simulation modules, initiate the development of selected modules, and demonstrate their use within an initial integration framework.

The DOE Work Scope includes the development of an interface and workflows (or guided processes) that allow the creation of self-consistent reactor power system designs without in-depth knowledge of the native modeling language or simulation environment. This interface must allow the user to provide input information, process input information

derived from the ruleset (i.e., preconfigured inputs based upon the user's choices) and create executable files to calculate a time-dependent response for the system. Currently the workflows established in the interface for novice users are limited to configured architectures that have been precompiled in Dymola. Novice users can select the subsystems and modify parameters for the generation of results without requiring knowledge of Dymola or Modelica. Additionally, the current interface tool also allows advanced users to create new systems, new models, even new architectures via access to source code in GitHub and use with Dymola. GitHub (<https://github.com/>) is a web-based platform that allows users to share, modify, and comment on code for project collaboration. The use of GitHub integration provides a framework for advanced users to push changes back to the ORNL project team and collaborate as part of the Open SMR community.

The ability to configure models easily (plug-and-play) with software tools requires a formal model architecture. This configurable model architecture is required to enable configuration via web applications. The configurable architecture also enables potential scripting of executable system configurations and FMU generation. The web-based application makes use of this hierarchical architecture structure to simplify the creation of new models from assembled components.

While it is difficult within the page length constraints to include all aspects of the project, it is very important that both unique aspects of this project be described within this paper. Thus, this paper includes the application of this approach to a novel space-advanced nuclear reactors as well as the generation of FMU simulations and application of web-based solutions. Unfortunately, a detailed discussion of both aspects is not possible within the publication constraints. Additional detail will be provided in further publications.

## 2 SMR Dynamic Modeling

### 2.1 LANGUAGE (Modelica)

Traditionally, modeling of complex reactor systems has been based on extensive, complex Fortran-based subroutines. Considerable time and effort were necessary to understand and manipulate these models. In contrast, the Modelica language has built-in features and open-source-toolsets for modeling fluid power systems. Physical systems are modeled by connect-

ing classes of components in series or parallel, and by specifying the important parameters of the objects. Fluids are defined as special classes that model the medium behavior using semi-empirical correlations or first-principles equation-of-state models that yield the full state of the medium as a function of two known states (e.g., pressure and enthalpy). These models can be sophisticated enough to support a wide range of operating regimes for fluids at the cost of computation time—and sometimes numerical stability. These property functions are frequently called during execution of simulations to calculate various engineering variables such as heat transfer coefficients and friction factors. The properties of water are already built into the standard Modelica Media library for use with the Modelica Fluid library. In addition to the Modelica Fluid Library, this project has made extensive use of the ThermoPower Library developed by the University of Milan [1]. The built-in water class has a complex set of routines that support working with water from sub-cooled to supercritical regions. As part of the project, ORNL implemented a number of heat transfer media, including liquid salts (flibe, flinak,  $\text{KFZrF}_4$ ) and liquid metals (sodium, NaK, PbBi eutectic) for use in Modelica. The Modelica models were based on the components and systems presented in [3] for the Power Reactor Innovative Small Module (PRISM) design concept. For validation and verification purposes, the results from running these models were compared with the results documented for the PRISM concept.

### 2.2 SOLVER (Dymola)

The simulation development and execution platform chosen for this SMR modeling effort is Dymola [4]. Dymola is a commercial modeling simulation environment developed by Dassault Systemes. Dymola has been used extensively within the automotive, aerospace, robotics and process system disciplines in part due to the extensive set of libraries developed and available within Dymola. The libraries include thermal and fluid dynamics. As a full simulation environment, a graphical interface to the model objects, as well as tools for the compilation and debugging of the models and analysis of the results are also included. To meet the project objective to provide access to dynamic simulation capability without special tools, export of Functional Mockup Units (FMUs) from Dymola is used in conjunction with FMI Add-in for Excel (FMIE) [5] from Modelon to allow simulation in Microsoft Excel. Co-simulation FMUs are generated for dynamic modeling in FMIE.

## 2.3 Architecture

Modelica and Dymola provide a highly convenient means to create reconfigurable model architectures which include high-level rules for systems, interfaces, and connections. As shown in Fig. 1, the ALMR Power Reactor Inherently Safe Module (PRISM) end-to-end plant systems architecture includes the following systems:

1. Direct Reactor Auxiliary Cooling System (DRACS),
2. Primary Heat Transport System (PHTS),
3. Intermediate Heat Exchanger (IHX),
4. Intermediate Heat Transport System (IHTS),
5. Steam Generator (SG),
6. Power Conversion System (PCS), and
7. Electrical Grid.

These systems will be defined later in the report in more detail. In addition to the plant systems, the following modules are included in the architecture:

1. Event Driver (ED),
2. Control System (CS), and
3. Communication bus.

The ED module generates signals for exciting the system. It performs two basic functions:

1. operational transients and
2. injection of faults.

Operational transients include power changes, load following commands, and intentional reconfiguration. A fault injection function introduces failures or performance degradation in identified plant components. Anticipated operational occurrences (AOOs) are handled through the fault injection function. The CS module performs continuous-time control functions for all actuation interfaces in plant systems based on sensory feedback.

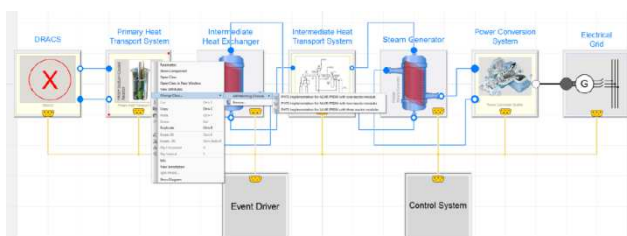


Fig. 1 SMR modeling architecture

## 2.4 Models

A small modular reactor consists of several typical components and subsystems. These include the reac-

tor core, piping and primary pumps that make up the reactor and primary systems. Additional piping, heat exchangers, pumps and steam generator are also included that make up the intermediate loop subsystem. Finally, turbines, and process system components are included in the power conversion system as well as generators and other grid related components in the grid subsystem. A discussion of the specifics of these systems is included in the sections below.

The dynamic models are implemented by equations in Modelica using a hierarchical modeling approach. The component models are often based upon various model libraries that are available as open source or as commercial products. Additionally, a user can create component models from their own Modelica code to supplement the existing libraries. Some of the reactor modeling includes component models created by the ORNL team.

The ALMR reactor design that has been modeled is based on the variation proposed by GE in the mid-1990s and documented in GEF-00793 [3]. The end-to-end system models developed are based on the systems, components, and equations included in this reference. For validation and verification purposes, the results from running these models were compared with the results documented for the PRISM concept. System behavior was confirmed compared to previous model studies.

### 2.4.1 Reactor Core

Most of the component models for SMR's are specific implementations of traditional thermal power components. However, the reactor core is unique and requires special consideration. As a result, the equations for modeling the core will be considered.

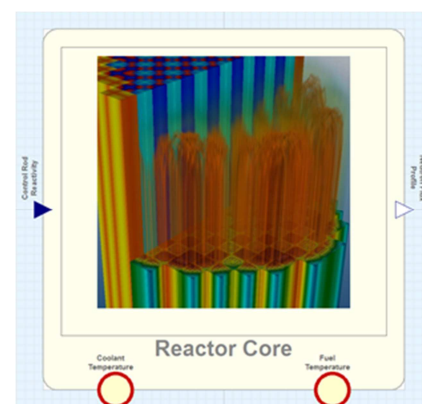


Fig. 2 Reactor core component

The prompt portion of normalized heat generation is implemented with the *point kinetics* equations.

$$\frac{dn}{dt} = \frac{\rho_t - \beta}{\Lambda} n(t) + \frac{1}{\Lambda} \sum_{i=1}^6 \beta_i c_i(t) \quad (1.a)$$

$$\frac{dc_i}{dt} = \lambda_i [n(t) - c_i(t)] \quad (1.b)$$

where  $n(t)$  is normalized reactor power,  $c_i(t)$  is the normalized concentration of the  $i$ -th group delayed neutron precursor,  $\beta_i$  is the fraction of the  $i$ -th group precursor,  $\lambda_i$  is the decay constant for the  $i$ -th group precursor,  $\rho_t(t)$  is the total reactivity, and  $\Lambda$  is mean neutron generation time. The rate equations are subject to steady state initial conditions, i.e.,  $\dot{n} = \dot{c}_i = 0$ .

The delayed portion of normalized heat generation is implemented with the following:

$$Q_{n-decay} = 0.1 \left[ (t + 10)^{-\frac{1}{5}} - (t + T_s)^{-\frac{1}{5}} + 0.87(t + T_s) + 2 \times 10^7)^{-\frac{1}{5}} - 0.87(t + 2 \times 10^7)^{-\frac{1}{5}} \right] \quad (2)$$

where  $t$  is time after shutdown and  $T_s$  is the operation time prior to shutdown—both in seconds.

The reactivity feedbacks are modeled as follows:

$$\rho_f = \alpha_f (T_{fe} - T_{f0}) \quad (2.a)$$

$$\rho_c = \alpha_c (T_{ce} - T_{c0}) \quad (2.b)$$

$$\rho_t = r\theta_{CR} + \rho_f + r\theta_c \quad (2.c)$$

where  $\rho_f$  is the fuel Doppler reactivity feedback,  $\rho_c$  is the coolant density reactivity feedback, and  $\rho_t$  is the total reactivity feedback.

The axial neutron flux shape is implemented as follows:

$$\Phi(z) = \Phi_{max} \cos\left(\pi \frac{z}{L}\right) \quad (3)$$

where  $z$  is the axial spatial variable that varies between  $-L/2$  and  $L/2$ , and  $L$  is the active length of

the fuel element. This particular representation is good for a wide range of reactor classes, but not quite so for Boiling Water Reactors, where flux shape is significantly skewed due to the boiling process.

For a given number of axial nodes, the integrated nodal flux is calculated with the following expression:

$$\begin{aligned} \phi_i &= \int_{-\frac{L}{2} + i\delta}^{-\frac{L}{2} + (i-1)\delta} \Phi_{max} \cos\left(\pi \frac{z}{L}\right) dz \quad (4) \end{aligned}$$

## 2.4.2 Primary Heat Transport System

The PHTS includes a reactor core model, a fuel pin thermal model, a discretized one-dimensional core flow model, a cover gas model, and a mechanical pump model. The Dymola/Modelica model diagram layer is shown in Fig. 3. The performance parameters of the mechanical pump are taken from the ALMR PRISM primary EM pump performance characteristics curve at 650 V, which is the operating point of all EM pumps under normal conditions. The mechanical pump will later be replaced with an EM pump model in the subsequent phases of the project. The model also includes four 90-degree turn elements to incorporate irrecoverable pressure losses due to turns and sudden expansion and contractions in the flow path. These loss elements are intended to match the total pressure drop during normal operating conditions across the core to the ALMR PRISM specifications.

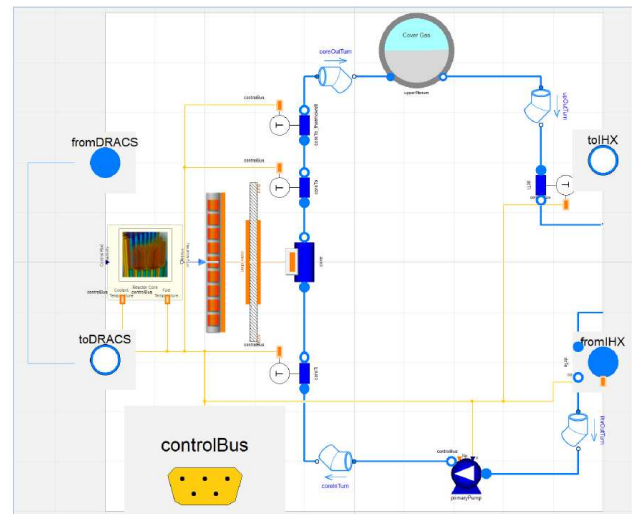


Fig. 3 SMR Primary Heat Transport System

### 2.4.3 Intermediate Heat Transport System

The ALMR PRISM baseline design includes two IHXs connected to a single IHTS. The IHTS flow loop (Figure 4) also includes a mechanical sodium pump, a sodium expansion tank, a sodium-to-water steam generator, and a sodium dump tank. The sodium dump tank is not modeled in the IHTS implementations. However, this tank should not have an impact on simulation results as this safety system is activated only under accident conditions where the sodium in the IHTS reacts with steam or air.

The ALMR PRISM intermediate sodium pump is a vertically oriented, single-stage, double-suction, free-surface, centrifugal pump driven by a constant-speed, 4000-hp (approximately 3000 kW), air-cooled induction motor. An auxiliary pony motor drive provides low flow (10%) capability for decay heat removal and other low-power or standby conditions. Its baseline design is to provide a flow of approximately 1900 kg/s at 96-m static head and 282°C. The main drive motor generates 3000-kW power at 1750 rpm.

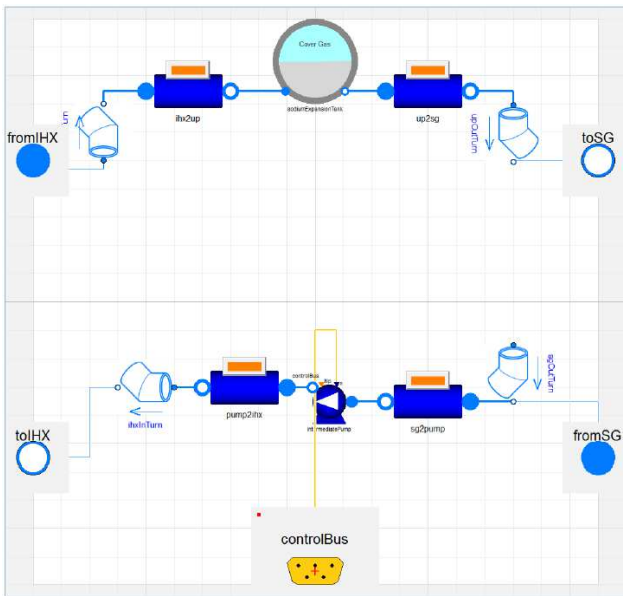


Fig. 4 SMR Intermediate Heat Transport System

### 2.4.4 Intermediate Heat Exchanger

The ALMR PRISM Intermediate Heat Exchanger (IHX) is modeled using two one-dimensional flow elements connected through a radial metal tube element, as shown in Fig. 5. The metal tube element is radially and axially discretized; the number of nodes can be changed by the user. The material properties, that is, density, thermal conductivity, and specific

heat capacity, can also be defined and changed using the Dymola dialog interface.

The default medium on both sides of the IHX is sodium. However, both fluid media can be selected via the system dialog interface. A number of fluid models are provided in the drop-down menu for both shell-side fluid and tube-side fluid, such as incompressible sodium (default), compressible sodium, incompressible NaK, and compressible NaK. Additionally, the menu selection includes a number of incompressible salt models; however, selecting incompatible media may result in numerically unstable configurations as the default initial and boundary conditions are specified for the ALMR PRISM design in the implementations. In the later phases of the project, the menu selections will be bound to a certain reactor class logic, which will limit the number of fluid media options for a certain implementation. The fluid flow and the heat transfer models in the shell side and the tube side can also be selected via the system dialog interface.

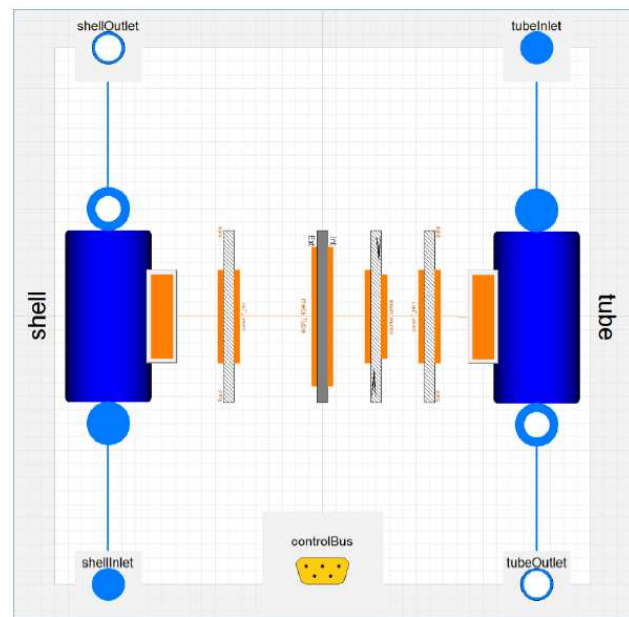


Fig. 5 SMR Intermediate Heat Exchanger

### 2.4.5 Steam Generator

The ALMR PRISM steam generator is a vertically oriented, shell-and-tube counter-flow heat exchanger with water/steam on the tube side and sodium on the shell side. The tubes are straight and of double-wall construction. The double-wall tubes provide improved reliability by significantly reducing the probability of water or steam leaking into the sodium.

The Modelica implementation of the ALMR PRISM steam generator is shown in Fig. 6. The ALMR PRISM steam generator is not a helical coil construc-



tion; therefore, it uses different analytical formulations for representation of fluid heat transfer and pressure drops. Furthermore, in contrast to the IHX model as shown in Fig. 5, the steam generator model employs two cylindrical tubes, one inner tube one outer tube, to account for the double-walled construction of the tube bundle. The other tube model of the steam generator implementation introduces a small thermal resistance to account for small interface separation.

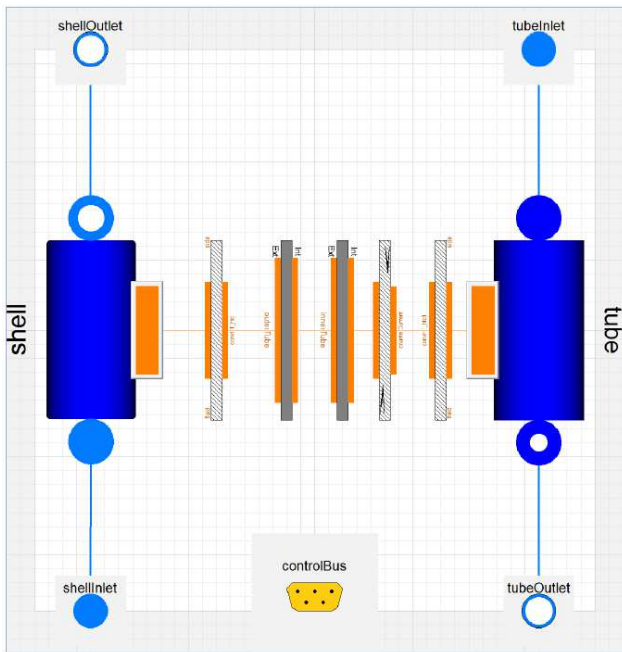


Fig. 6 SMR Steam Generator

### 2.4.6 Power Conversion System

The power conversion system model (Fig. 7) is comprised of several components, including the turbines, generator, flow control valve, and pump. Standardized modules that include representative equations/curves are used for these modules. No specialized modules for the ALMR were developed. Current implementation is a simplified version of the actual power conversion system where the feedwater heater trains, blowdown coolers, and the deaerator are eliminated. The implementation also uses a constant rotor input (on the left side of the rotor shaft as a boundary condition represented by the “shaftSpeed” element) to eliminate additional control requirements to maintain shaft angular speed. The rotor mass and rotational mechanics are represented by the rotor element.

The “steamInlet” and “condensateReturn” ports are implemented as vector objects whose number of elements must match the number of steam

generators, which in turn matches the number of reactor modules in a power block. In the current implementation, up to three steam lines can be allocated; however, this limitation will be removed later. Each steam line contains a “stopValve” element (shown as “stopVx”) to control steam flow from each stream. It should be noted that the “stopValve” elements are not modulating elements; they have binary states of OPEN or CLOSED. The valves’ opening and closing dynamics are represented by a linear ramp with a user-specified time constant to switch from one state to another. The streams from three steam generators are mixed via two “flowJoin” elements.

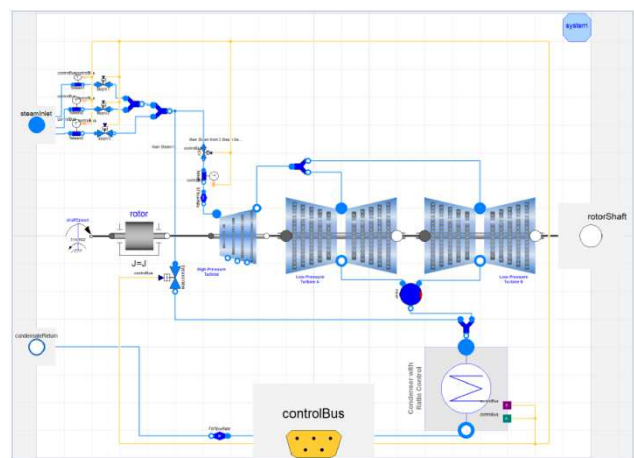


Fig. 7 SMR Power Conversion System

### 2.4.7 Grid

A simplified grid model has been developed. The Modelica implementation of the generator and the electrical grid is shown in Fig. 8. In the current implementation, the electrical generator is modeled as an ideal synchronous generator where the frequency in the electrical connector is the electromotive force (emf) of the generator. The frequency of generated electrical current is by default 60 Hz but can also be adjusted by the user. It should be noted that the generator frequency should match the grid frequency; otherwise, it will cause a generator trip and disconnection from the grid, which will in turn initiate a turbine trip.

The electrical grid is modeled with the *swing equation*, which calculates the *load angle* as a function of differences between the mechanical torque (provided by the turbine shaft) and the electrical torque (resistant torque generated by the electrical load). The load angle dictates the amount of power that can be transferred to the grid. The power sensor and the frequency sensor are used to monitor the load

angle of the network. If the frequency gets out of range, an anticipatory automatic trip is initiated for the turbine and the generator.

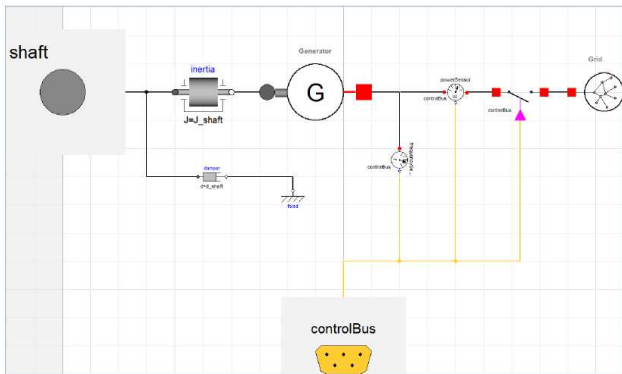


Fig. 8 SMR Grid System model

### 2.4.8 I&C

The purpose of this project is to design a system of modeling tools that allows rapid assessment of control system strategies overlaid upon reactor system models. For demonstration purposes, two different control strategies were chosen to allow multiple selection options to exist in the modeling tool: Strategy #1 – temperature regulation and Strategy #2 – temperature difference regulation.

The purpose of developing these systems models is to investigate potential control strategies for advanced SMR concepts. The I&C studies and concept development include identification of the model global outputs, determination of measurements, control actuation, and control methods and algorithms for a given system. The dynamic model of the process and plant physics provides model results and outputs for candidate measurements (system observability) and model inputs for candidate actuation (system controllability). The dynamic model also provides insights into the system time constants, degree of inherit stability, degree of coupled behavior, and input-output sensitivity. Figures 9-11 show the sensing instruments, model architecture and control logic associated with this simplified I&C control module that is overlain on the reactor systems model.

Figure 12 illustrates an example simulation in Dymola of a control system during energy demand changes. The PHTS primary pump and IHTS intermediate pump and record control rod reactivity are all actuated to regulate the reactor system temperatures as the heat sink changes. The generation of these outputs using the systems modeling approach with Modelica is relatively straightforward for the practiced simulation engineer. The goal of this project is to extend this capability to those engineers

who are not as familiar with these modeling and simulation environments.

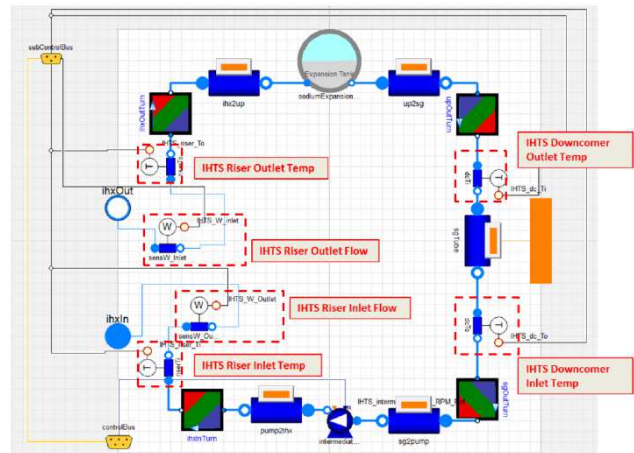


Fig. 9 SMR I&C sensing instruments

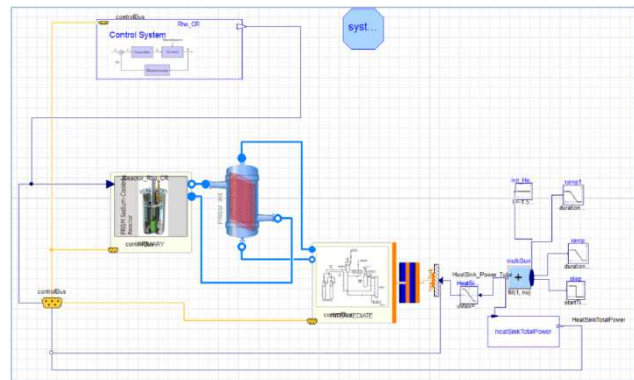


Fig. 10 SMR I&C

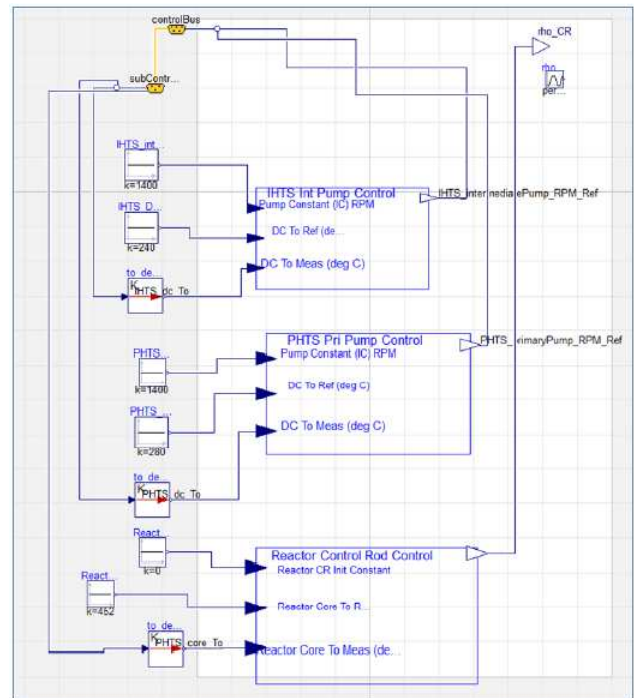


Fig. 11 SMR I&C architecture

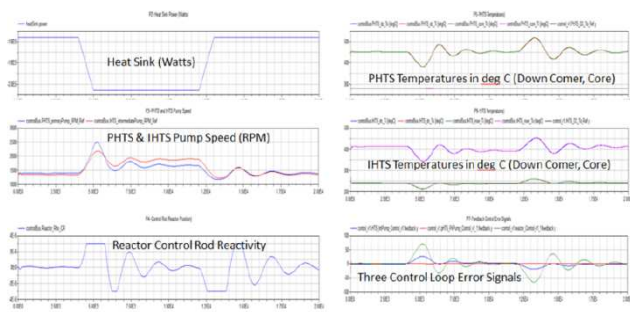


Fig. 12 Initial control system testing results

### 2.5 FMI Add-in for Excel

Although simulation is a valuable tool for science and engineering, few scientists and engineers have the proper skill set or day to day need for modeling and simulation. As a result, the investment of time and resources to establish and maintain a modeling and simulation capability is often not justified. An open framework would provide an alternative approach which will open modeling and simulation capabilities to activities that cannot justify the investment in traditional modeling tools.

The development of these models initially requires some knowledge of Modelica and Dymola. Although most engineers may not be familiar with Modelica or Dymola, many engineers are familiar with and fairly skilled in Excel. Therefore, the project has built the initial interface tool around an Excel platform that allows the simple and intuitive generation, manipulation, plotting, and reporting of results using the FMIE [5] tool developed by Modelon. This tool (Fig. 13) allows the integration of Excel with compiled co-simulation FMUs for simulation. Many simulation environments (not just Modelica based) now allow the export of these FMUs. The use of the open standard FMI provides an opportunity for a flexible toolchain in terms of tools used for core model development, simulation, and engineering analysis along with associated licensing.

FMIE can process a model that has been compiled into an FMU by simulation platform like Dymola in a manner that facilitates performing dynamic simulations in Excel. The tool also supports changing model parameters, adjustment of the simulation start and stop times, for single or multiple simulations which can be run in parallel. The resultant input and output data variables can be selected for display in graphical or tabular form in a worksheet. FMIE also provides a scripting API which can be used for automation of analysis set-up, simulation, plotting, and post-processing.

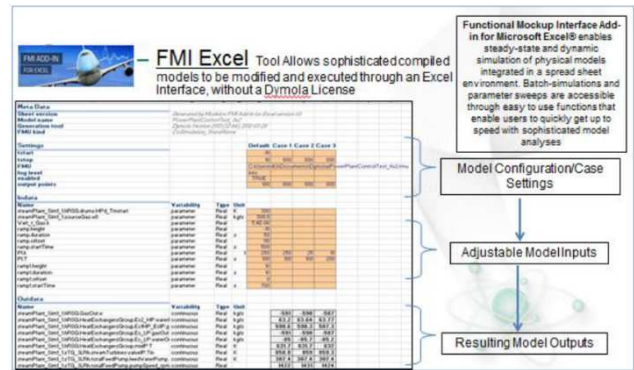


Fig. 13 Modelon FMIE interface

### 2.6 Collaboration

The initial user interface for MoDSim has been written in Excel (Fig. 14). This makes for easy integration with FMIE used to simulate the Modelica models outside of Dymola. Excel macro options that allow for the modification of inputs, the display of outputs, and the generation and plotting of results (Fig. 15) are provided along with the ability to display the metadata about the simulation. This interface tool, however, only serves as the prototype in terms of features and layout for the web application interface that is being developed. The principal limitation associated with the prototype Excel interface is its use as a local application on a machine. The widespread collaboration that is the intended goal is only truly achievable from a web-based platform.

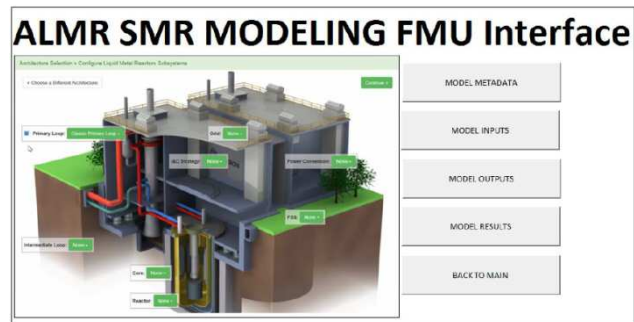


Fig. 14 SMR Excel interface

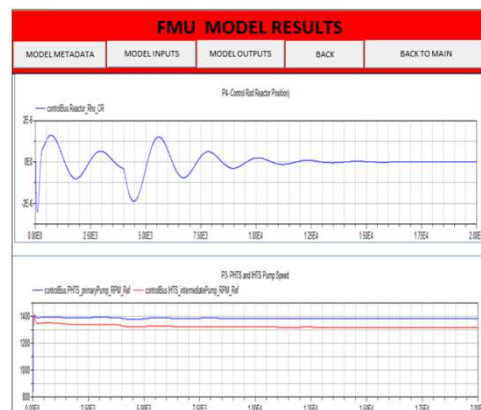


Fig. 15 SMR Excel interface - results



The heart of the collaborative effort is based on using an online open-source version control, issue tracking, and file sharing application. The application chosen is GitHub. Users can obtain a GitHub account and request access to the ORNL SMR folder (ORNL-WebSMR) displayed in Fig. 16. Access to this folder allows the user to work directly in the text-based Modelica models and make modifications, comments, etc. on the models. Additionally, issues can be raised and tracked as well as versions of the models created, archived, and documented. This allows collaboration within the user community.

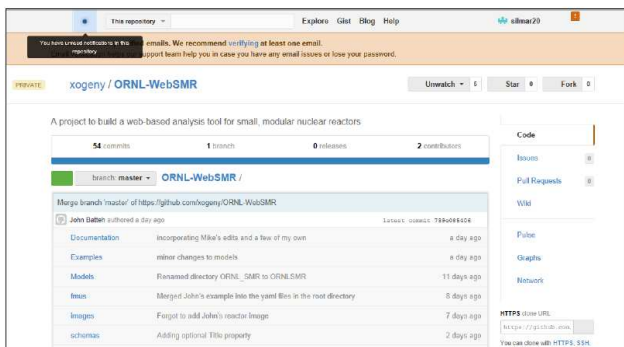


Fig. 16 SMR GitHub model repository

The start page of the SMR web application built by Xogeny, Inc. is shown in Fig. 17. The web platform utilizes the architecture described in Section 2.3 along with configuration files that provide architecture, subsystem choice, parameters, and output information for use in generating the actual web application. These files are generated in the YAML (Yet Another Markup Language) language. The user can browse the available reactor architectures and then selects the one with which to work. Following the selection of the reactor architecture, the user can then configure the subsystems in the architecture as shown in Fig. 18. After subsystem selection, the user then has the option to modify the parameters in each subsystem (web app is populated with the default parameters from the FMU) as shown in Fig. 19.

The current workflow is then to generate a simulation with FMIE as shown in Fig. 20. The web application provides a download of a customized Excel sheet developed by Modelon that utilizes the scripting API in FMIE to automatically download the FMU populated with parameters entered by the user, execute the simulation, and plot the results. The resulting experiment sheet in FMIE can then be used to conduct batch simulations, modify additional parameter values, and automatically compare results between simulations. Figure 21 shows a sample experiment sheet with batch simulation compare. This automation with FMIE provides a one-click work-

flow that couples the web-based configuration and parameter setting with a local simulation running in Microsoft Excel.

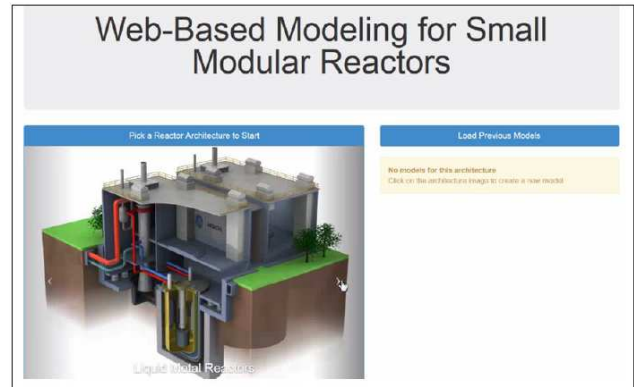


Fig. 17 SMR web app start page

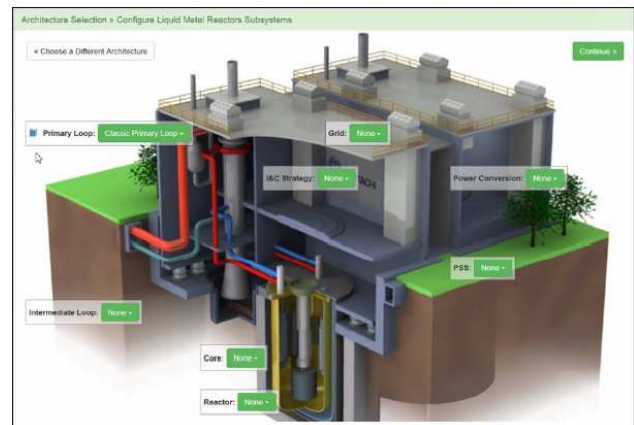


Fig. 18 SMR model configuration

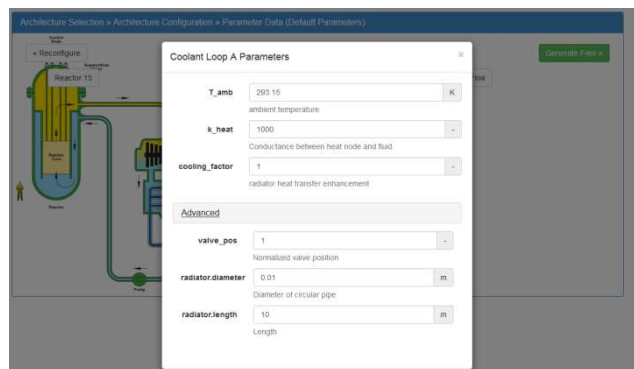


Fig. 19 Parameter specification

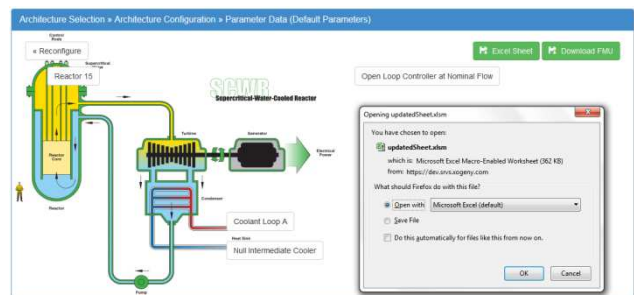


Fig. 20 Simulation generation with FMIE

Model		Generated by Modelon FMI Add-in for Excel version 1.2										
Sheet version		ORNL Demo Systems-FinalHeatMassFlowOpen										
Model name		Dymola Version 2013 FD01 (64-bit), 2012-10-18										
Generation tool		C:\Users\jbatteh\AppData\Local\Temp\FMUD12C\mp_K338T1\3HEN7.Fmu										
FMI kind		Cocimulation_Standard										
Number of processes		0										
Settings		Default	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10
Start time		0										
Stop time		1000										
FMIU		C:\Users\jbatteh\AppData\Local\Temp\FMUD12C\mp_K338T1\3HEN7.Fmu										
Log level		Info										
Enable		TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Output points		100										
Timeout		0										
Inidata		Name	Variability	Type	Unit							
		reactor.rad.T	continuous	Real	K	293.15		300				
		reactor.Q_reactor	parameter	Real	W	1500		1750				
		reactor.C_rad	parameter	Real	J/K	10						
		coolant_loop.radiator.length	parameter	Real	m	10						
		coolant_loop.radiator.diameter	parameter	Real	m	0.01						
		coolant_loop.h_heat	parameter	Real	1	1000						
		coolant_loop.wiring.gas	parameter	Real	1	1						
		coolant_loop.T_amb	parameter	Real	K	293.15						
		coolant_loop.cooling_factor	parameter	Real	1	1						
		controller.mflow_pump	parameter	Real	kg/s	0.01						
Outdata		Name	Variability	Type	Unit							
		reactor.node.Q_flow	continuous	Real	W	1499.94		1749.93				
		reactor.rad.T	continuous	Real	K	486.632		516.181				
		coolant_loop.tank.level	continuous	Real	m	1.0038		1.00455				
		coolant_loop.coolant_bus.T_rad_in	continuous	Real	K	326.643		364.188				
		coolant_loop.coolant_bus.T_rad_out	continuous	Real	K	307.842		310.412				
		coolant_loop.coolant_bus.mflow_pump_sensed	continuous	Real	kg/s	0.01		0.01				
		coolant_loop.coolant_bus.mflow_pump	continuous	Real	kg/s	0.01		0.01				
		controller.coolant_bus.coolant_bus.mflow_pump	continuous	Real	kg/s	0.01		0.01				
Message												
			OK	OK	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled

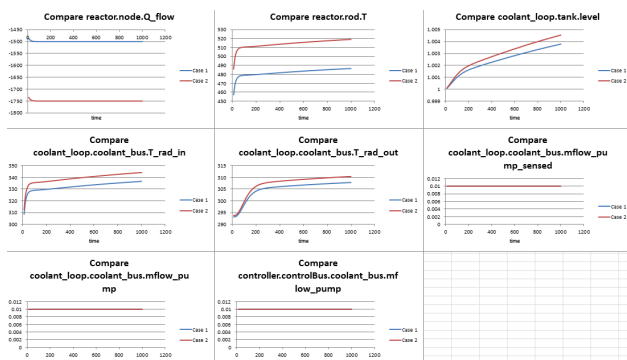


Fig. 21 Sample experiment sheet in FMIE with batch simulation compare

Currently the web platform is an alpha product which still requires local installation of tools, and the alpha version is being used to define the requirements for the beta product. The beta version will be a completely web-based application that can be run from any user's desktop without the need to install any software. ORNL is working with Modelon and Xogeny to facilitate a web-based Modelica simulation application towards this end. The beta product is expected to be out in FY 2015. The goal of the web platform is to support both local and cloud-based simulations within licensing constraints of the development environments. As with any compiled simulation environment, the practical challenges of pre-configured models, initialization, and simulation performance must be handled in a robust way to ensure successful deployment and effective use of the simulation platform.

### 3 Conclusions

A growing number of programs within the DOE portfolio will benefit from the streamlined approach to collaborative modeling that this tool affords. Other actively funded activities in the DOE Advanced Reactors Program are planning to incorporate this tool

into their activities. This tool is well suited for work in the study of hybrid energy systems. ORNL is actively engaged in developing collaborations in the hybrid energy system space that utilized this approach. In particular, ORNL is working with the Idaho National Laboratory to create these hybrid energy system models that can be used to provide high level evaluations of potential energy concepts that make use of multiple energy generation paths. In the next phase of development, we will actively engage these projects and seek additional projects that will also benefit from this tool.

As a final note, the long-term success of this activity will be judged by three items: quality of the models developed, the number of users, and the ease of use. To this end, further development in the web-based platform is an important next step. The web-based platform provides opportunities for cloud-based modeling for both nonsensitive material as well as controlled access for sensitive, proprietary material. Additionally, the ability to create scripts that will allow for rapid assessment of parametric sweeps will simplify the input and output necessary for the user. However, it is critical that multiple projects and programs begin using this tool to generate a library of models that will form the backbone of this collaborative network for SMR dynamic modeling. By generating this simplified modeling platform it is hoped that a new paradigm for systems modeling will be created that will help facilitate the development of the next generation Small Modular Nuclear Reactor, but other complex engineering systems as well.

### References

- [1] F. Casella, A. Leva, "Modelica open library for power plant simulation: design and experimental validation", *Proceedings of the 2003 Modelica Conference*, Linköping, Sweden, November 2003, pp. 41-50.
- [2] ORNL/TM-2013/426, "SMR Dynamic System Modeling Tool Update: September 2013", September 2013.
- [3] GEF-00793, "PRISM Preliminary Safety Information Document", UC-87Ta, December 1987.
- [4] Dassault Systemes, "Dymola 2013 FD01", 2012.
- [5] Modelon, "FMI Add-in for Excel", Version 1.2.1, 2013. <http://www.modelon.com/products/fmi-add-in-for-excel/>