

A Templated Approach for Multi-Physics Modeling of Hybrid Energy Systems in Modelica



M. Scott Greenwood
M. Sacit Cetiner
T. Jay Harrison
David Fugate

August 2017

Approved for public release.
Distribution is unlimited.

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website <http://www.osti.gov/scitech/>

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website <http://classic.ntis.gov/>

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website <http://www.osti.gov/contact.html>

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.

Reactor & Nuclear Systems Division
Electrical and Electronics Systems Research Division

**A TEMPLATED APPROACH FOR MULTI-PHYSICS MODELING OF HYBRID
ENERGY SYSTEMS IN MODELICA**

M. Scott Greenwood
M. Sacit Cetiner
T. Jay Harrison
David Fugate

Date Published: August 2017

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

LIST OF FIGURES	v
LIST OF TABLES	vii
ABSTRACT.....	1
1. INTRODUCTION	1
1.1 MODELICA	3
2. TEMPLATED MODELING	4
2.1 SIGNAL BUSSES	6
2.2 TEMPLATES	8
2.2.1 Inheritance and Replaceability.....	8
2.2.2 Specific Subsystem	8
2.2.3 Sub-System Category.....	11
2.2.4 Overall Hybrid System	12
3. TEMPLATE USE DEMONSTRATION.....	13
3.1 PRIMARY HEAT SYSTEM.....	15
3.2 ENERGY MANIFOLD	17
3.3 SELECTION OF BASIC SUB-SYSTEM DYNAMICS	19
4. CONCLUSIONS AND FUTURE WORK	22
5. ACKNOWLEDGEMENTS.....	22
6. REFERENCES	23
APPENDIX A. EXAMPLE GUIDE TO USE THE TEMPLATE	A-1

LIST OF FIGURES

Figure 1. Representative NHES demonstrating a possible coupling scenario of thermal and electrical energy with additional systems (e.g., an industrial process [Reforming and Enhancement] and an electrical energy storage system [Battery]) (Bragg-Sitton et al. 2014).	2
Figure 2. Prediction of electrical grid variability for US regions in 2050.	2
Figure 3. Reactor core subchannel model and resulting total thermal output based on the boundary conditions and input signals: (a) example of Modelica code used to create a component, (b) process to create a more complex model from a collection of individual components, (c) implementation of the connected models to simulate a test case, and (d) resulting behavior of model.	3
Figure 4. Top-level implementation of an HES using a templated approach: numbers correspond to	5
Figure 5 (a). Overall Modelica package for investigation of HESs, (b) template structure for creating new subsystem categories and specific subsystem models within a category, and (c) example of a specific implementation of a primary heat system using the template approach.	6
Figure 6. Various signal busses used in Modelica HES modeling; colors represent specific uses of a bus, while icons indicate whether a bus is intended for signal sharing only within subsystems or for exposure of signals between all subsystems.	7
Figure 7 (a) Implementation of subsystem sensor/actuator sub-bus containing the signals associated with the specific subsystem of interest (e.g., in this case the International Reactor Innovative and Secure [IRIS] model) and additional variables common among all subsystems within a specific class of subsystems, (b) full sensor/actuator bus, containing all sub-busses from each subsystem used in the model, and (c) illustration of how the organizational structure is interpreted upon simulation.	7
Figure 8. Example of inheritance and replaceability as used in Modelica: the <i>extends</i> clause signals inheritance from the base class; features of the base class are then modified by assigning values, adapting equations, or replacing models with their own individual inheritance structures using the <i>replaceable</i> clause.	8
Figure 9 (a) Directory structure of the template of any specific implementation of a subsystem; “BaseClasses” directory establishes models that allow for inheritance from the appropriate subsystem category while limiting the acceptable replaceable models (e.g., control systems) only to the specific sub-system to which they belong, and (b) graphical workspace in which the modeler views base classes for signal busses, control systems, event drivers, and data.	9
Figure 10. Use of <i>extends</i> and redeclaration of replaceable models enables a specific subsystem to share a common architecture for development of categorically defined hybrid systems while enabling subsystem-specific models to restrict options to only pertinent selections such as various control systems strategies or specific subsystem records.	10
Figure 11. Signal sub-bus of the specific model IRIS within the primary heat system category; the sub-bus enables variables unique to the specific subsystem to be available to other models.	10
Figure 12. Directory structure of the subsystem category template containing templates for specific subsystems, along with a base class directory providing the common architecture for all specific implementations within the subsystem category.	12
Figure 13. Standard implementations of subsystems generated using inheritance capability to improve tractability and speed of implementing variations of an overall system (Figure 4 is the graphical view of “PartialExample_A”).	13

Figure 14. Base Modelica architecture of an HES. Subsystems have been replaced with place holder systems (Figure 4 and Figure 6).	14
Figure 15. Case 1: Hybrid system based on the template structure using a loosely coupled industrial process: the empty-set symbol indicates that no energy storage or secondary energy supply are in this example.....	14
Figure 16. Case 2: Hybrid system based on the template structure using a tightly coupled industrial process.	15
Figure 17. Templated model of the primary heat system based on the IRIS reactor.	16
Figure 18. Control system for IRIS primary heat system based on total core power and system pressure.	17
Figure 19. Energy manifold for distributing and gathering steam/water streams from subsystems (bypass valve dampens pressure responses of the system when tracking demand profiles, and the splitter and mixing volumes dampen transient thermal behavior).	18
Figure 20. Energy manifold control system with the turbine distribution valve actuated to meet the demand profile set points of the balance of plant, while the bypass valve acts exactly opposite to the turbine distribution valve to dampen pressure swings within the system.	19
Figure 21. Pressurizer pressure in the primary heat system with oscillatory behavior due to the dynamics of the energy manifold and the liquid heater compensation raising the pressure from the minimum 15.0 MPa to the nominal 15.5 MPa.	20
Figure 22. Core thermal power and control rod reactivity behavior necessary to keep the primary heat system operating at a nominal thermal power level of 1,000 MW.	20
Figure 23. Balance of plant generated power compared to the set point provided by the supervisory control system, in which the dynamic behavior of the turbine and bypass valves work together to meet the specified power and dampen impacts on the connected subsystems.	21
Figure 24. Result of power delivered to the grid compared to the set point demanded; periods of over and under production are evident, and the shift in differential power between the two cases demonstrates impact that variations of system architecture and subsystems can have on HES performance.....	22

LIST OF TABLES

Table 1. Potential subsystems of an HES 5

ABSTRACT

A prototypical hybrid energy system (HES) couples a primary thermal power generator (i.e., nuclear power plant) with one or more additional subsystems beyond the traditional balance of plant electricity generation system. The definition and architecture of an HES can be adapted based on the needs and opportunities of a given local market. For example, locations in need of potable water may be best served by coupling a desalination plant to the HES. A location near an oil refinery may have a need for emission-free hydrogen production. The flexible, multidomain capabilities of Modelica are being used to investigate the dynamics (e.g., thermal hydraulics and electrical generation/consumption) of such a hybrid system. This paper examines the simulation infrastructure created to enable the coupling of multiphysics subsystem models for HES studies. A demonstration of a tightly coupled nuclear hybrid energy system implemented using the Modelica based infrastructure is presented for two representative cases. An appendix is also included providing a step-by-step procedure for using the template-based infrastructure.

1. INTRODUCTION

Electricity markets in the United States are undergoing significant shifts in the traditional market structure. Factors such as mandates for renewable energy, the need for overall carbon reduction, and the emergence of low cost natural gas supplies have strained the profitability of traditional base load electricity suppliers, including nuclear power plants.

Because typical base load power generating stations (e.g., nuclear power plant) traditionally have only one customer—the grid—diversification of the customer portfolio in an integrated or hybrid manner may be advantageous. The term *hybrid energy system* (HES) has been coined to describe the integration of traditionally independent systems, and those that use nuclear generating stations as the primary thermal power supplier are defined as *nuclear hybrid energy systems* (NHESs). A representative NHES is depicted in Figure 1.

The HES approach that directly couples base load energy suppliers and various energy customers (thermal and/or electric) with dynamic energy allocation may be profitable and preferred in future energy markets. Possible scenarios include *ad hoc* energy allocation for product options that could be more profitable than traditional electricity generation. A description of economic incentives is beyond the scope of this paper, but these incentives may include (1) a desire to mitigate the possible load-following need—and subsequent cost increases—that significant renewable penetration may impose on base load power plants through energy storage, or (2) a desire to more effectively deploy energy to large industrial energy users. For example, Figure 2 is a representative summary of a recent study by the Electric Power Research Institute (EPRI) on the impact of renewable energy generation on grid variability (James et al. 2015). Given current economic and political trends, future electrical grids will require highly variable operations that impose significant technical and economic challenges for power producers. Introducing hybrid energy systems may help create a path to integrating highly variable generation resources into markets that remain economically sound and do not compromise grid reliability.

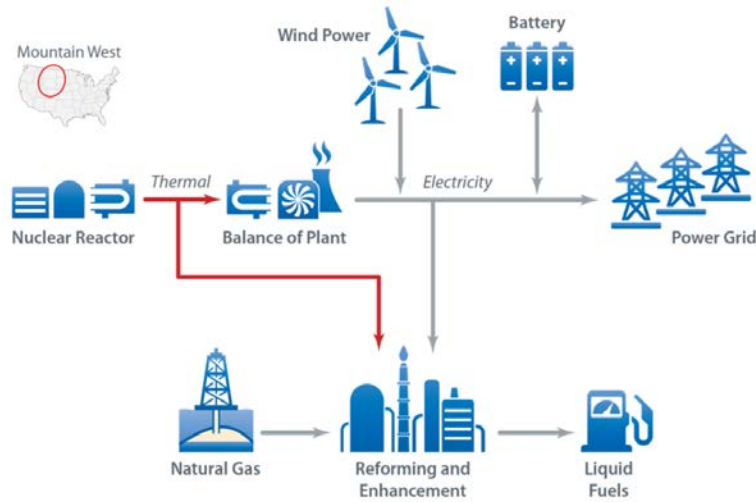


Figure 1. Representative NHES demonstrating a possible coupling scenario of thermal and electrical energy with additional systems (e.g., an industrial process [Reforming and Enhancement] and an electrical energy storage system [Battery]) (Bragg-Sitton et al. 2014).

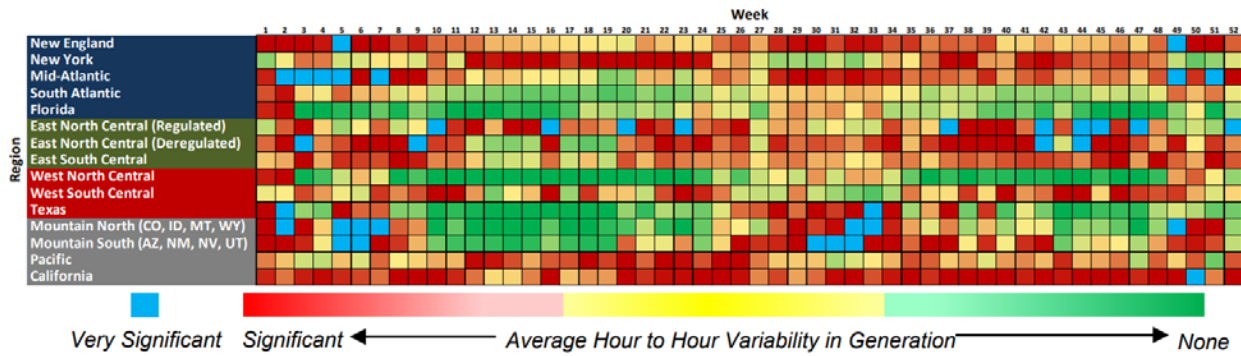


Figure 2. Prediction of electrical grid variability for US regions in 2050. Cell colors indicate variability: regions approaching red and blue have demands that will be difficult and expensive for the electrical grid to meet, especially for power producers operating under traditional market paradigms (James et al. 2015).

To facilitate initial investigations of hybrid systems, the complications of attempting to integrate multiple independent codes for specific modeling paradigms are avoided, as are the subsequent complications of coupled dynamic simulations. Instead, it is preferable to rely on the flexibility of modern system modeling programming languages such as Modelica to model all systems jointly. However, even within a common development environment there are difficulties in ensuring all necessary information is specified by the user and available to the various portions of the model. This point is exacerbated when model development is decentralized, incorporating contributions from multiple users from various institutions focusing on separate compartmentalized pieces of the overall model. Therefore, to facilitate collaboration and enhance the probability of developing successful simulations a common development infrastructure is required which requires developers meet requirements in model implementation while providing them as much flexibility as possible in model development. This paper presents the infrastructure or “template” created to enable coupling of multiphysics subsystem models for HES studies in a common, structured format. The paper concludes with a brief demonstration of a tightly coupled NHES implemented using the new Modelica-based infrastructure. Additional information beyond the scope of this paper can be found in (Cetiner et al. 2016; Harrison and Greenwood 2016; Greenwood et al. 2017).

1.1 MODELICA

Modelica (Modelica Association) is a nonproprietary, object-oriented, equation-based programming language used to consistently model complex physical and cyber-physical systems (e.g., systems containing mechanical, electrical, electronic, hydraulic, thermal, control components). Modeling a dynamic HES typically requires a complex, diverse range of physics. The multi-domain nature of Modelica permits flexibility of using one tool to model the appropriate physics for all systems and their associated control systems. Modelica avoids the limitations and complexities of attempting to integrate complex codes with features and capabilities beyond the scope of initial HES studies. There are several integrated development environments for Modelica, all with different strengths and weaknesses. A complete list of development environments (commercial and open-source) can be found on the Modelica Association website under “Tools” (Modelica Association). This work currently uses Dymola version 2017 and newer versions (Dassault Systems).

Modelica’s modeling method is characterized as shown in Figure 3, which demonstrates (1) scripting of generic models that are (2) linked to create increasingly complex models, followed by (3) verification tests to investigate the model’s behavior. This general, layered approach is common for developing complex systems in Modelica and is a strong feature of the language.

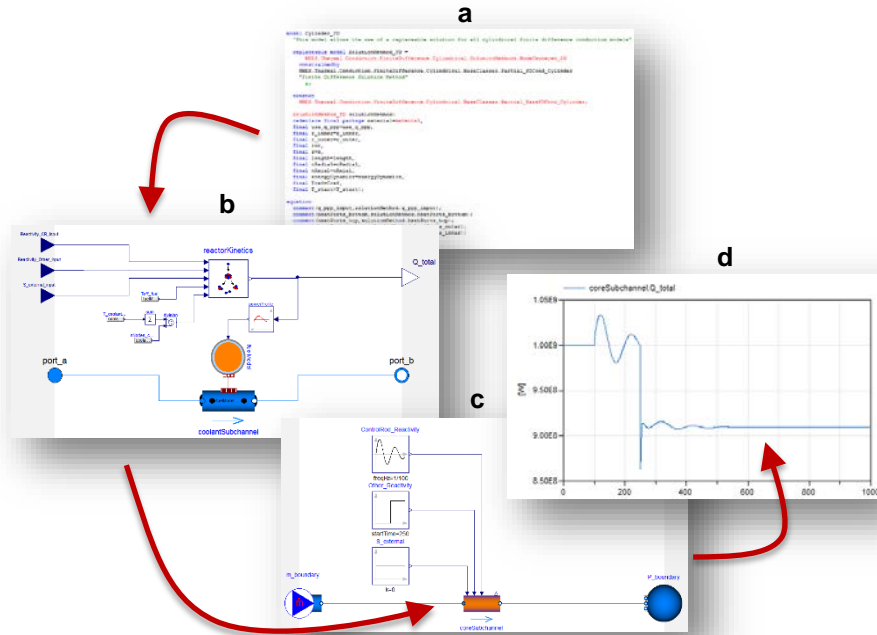


Figure 3. Reactor core subchannel model and resulting total thermal output based on the boundary conditions and input signals: (a) example of Modelica code used to create a component, (b) process to create a more complex model from a collection of individual components, (c) implementation of the connected models to simulate a test case, and (d) resulting behavior of model.

2. TEMPLATED MODELING

A templated approach for developing various subsystems an HES is important for facilitating the investigation of potential integrated system candidates and the dynamic behavior of the overall system. The primary goals of the templated approach are listed below:

- I. Develop templates for specific implementations of subsystems that handle all potential interrelated subsystem signals (e.g., control systems and fluid connections) thereby allowing the user to focus on modeling their respective subsystem
- II. Implement templates for categories of subsystems that allow users to easily interchange a subsystem at the overall architecture level according to the case study of interest while minimizing required user knowledge of other subsystems
- III. Provide the necessary handles and structure within the templates to implement and verify nominal and initial conditions at subsystem interfaces
- IV. Ensure that all templates are as generic as possible to accommodate a large variety of any given subsystem, with minimal or no modification to the underlying infrastructure

Nine potential subsystem categories have been identified for hybrid systems in its current configuration. Figure 4 presents a top-level view or architecture of a templated Modelica-based model which is referenced throughout the remainder of this article. Table 1 provides a textual summary of each of the nine subsystem categories as presented in Figure 4. The templated models and supporting file structure upon which Figure 4 is based are shown in Figure 5. The following subsections will describe key aspects of the template implementation and details on how the goals identified above are accomplished.

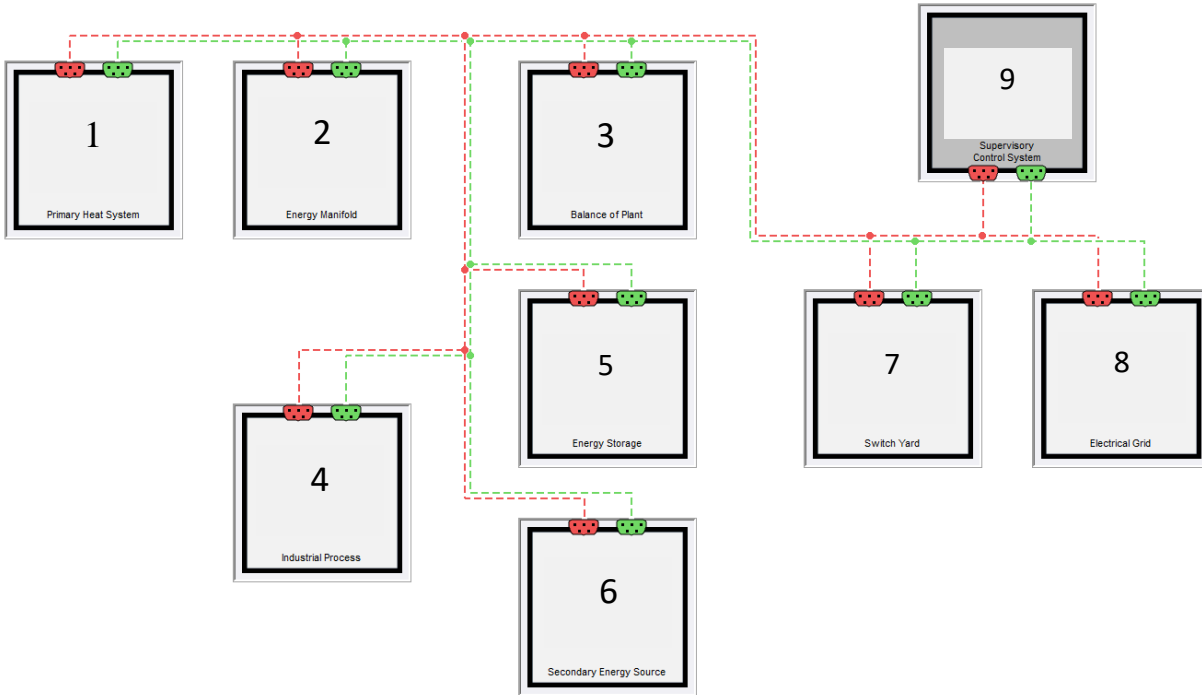


Figure 4. Top-level implementation of an HES using a templated approach: numbers correspond to Table 1; red and green dotted lines are connection lines for the signal busses described below.

Table 1. Potential subsystems of an HES

Identifier	Category	Description	Specific example
1	Primary heat system	Provides base load heat and power	Nuclear reactor
2	Energy manifold	Distributes thermal energy between subsystems	Steam distribution
3	Balance of plant	Serves as primary electricity supply from energy not used in other subsystems	Turbine and condenser
4	Industrial process	Generates high-value product using heat from energy manifold/secondary energy supply and electricity from switch yard	Steam electrolysis or desalination
5	Energy storage	Serves as energy buffer to increase overall system robustness	Batteries and firebrick
6	Secondary energy source	Delivers small amounts of topping heat required by industrial processes or cover rapid grid dynamics not met by the remainder of the system	Gas turbine makeup
7	Switch yard	Distributes electricity between subsystems, including the grid	Electricity distribution
8	Electrical grid	Sets the behavior of the grid connected to the NHES	Large grid behavior (not influenced by NHES)
9	Control system center	Provides proper system control, test scenarios, etc.	Control/supervisory systems and event drivers



Figure 5 (a). Overall Modelica package for investigation of HESs, (b) template structure for creating new subsystem categories and specific subsystem models within a category, and (c) example of a specific implementation of a primary heat system using the template approach.

2.1 SIGNAL BUSSES

The Modelica programming language enables a class of components referred to as *expandable connectors* (Modelica Association 2017)(Section 9.1.3), which enable creation of multilevel models containing variables that can be ignored in the simulation process if they are not used. Expandable connectors also permit additional variables to be added as needed. This capability is the cornerstone for enabling template creation of diverse systems.

Figure 6 shows different variations of signal busses created for HES template modeling. Note that the colors of the icons themselves are purely aesthetic for differentiating between the various bus types as each bus is an *expandable connector*. Red busses are designed for variables generally defined as *sensor* or *output signals* (e.g., temperature and pressure measurements). Green busses are for designed variables

generally defined as *actuator* or *input signals* (e.g., resistance heater voltage and pump impeller rotational speed). Yellow busses combine all signal types and require other methods (i.e., variable labeling) to differentiate signals to provide information to the model developers. Each color of bus contains two types, a *signal sub-bus*, depicted with three black circles, and a normal *signal bus*, depicted with five black circles. Sub-busses are specific to a subsystem and cannot have additional sub-busses defined within them, while busses may contain multiple sub-busses and any additional variables desired. The signal busses are readily identifiable within the various figures presented in this publication. Figure 7 demonstrates implementation of a sensor/actuator bus for a model within the primary heat system category.

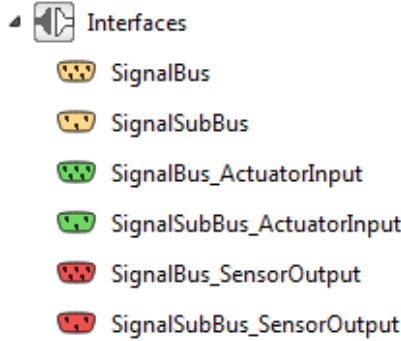


Figure 6. Various signal busses used in Modelica HES modeling; colors represent specific uses of a bus, while icons indicate whether a bus is intended for signal sharing only within subsystems or for exposure of signals between all subsystems.

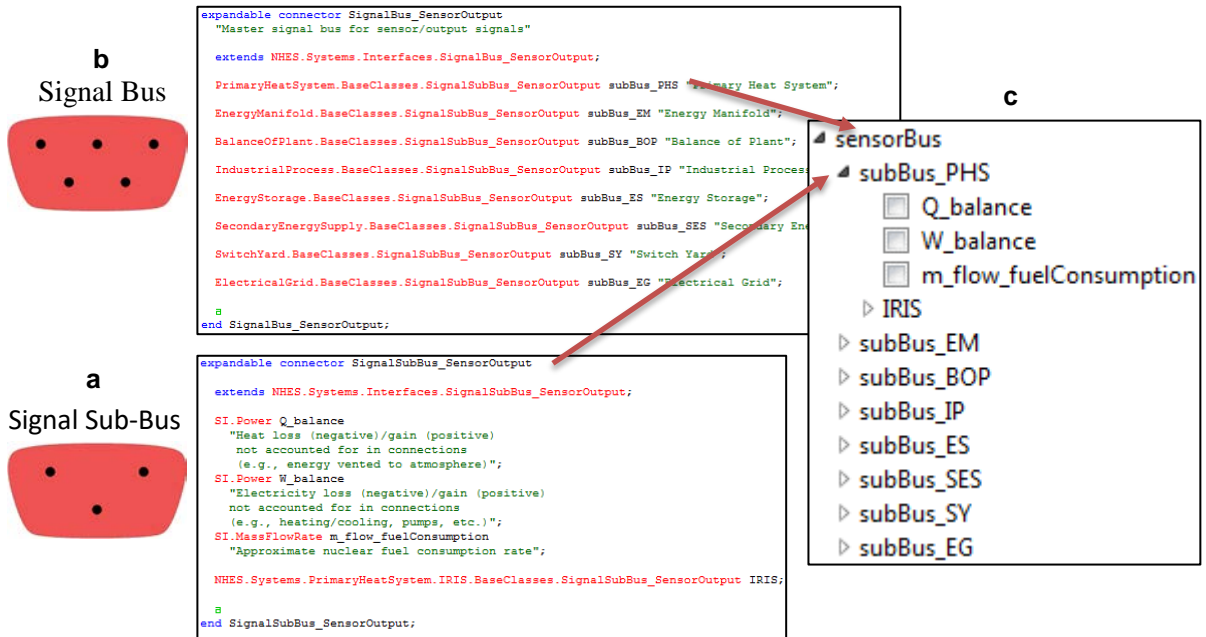


Figure 7 (a) Implementation of subsystem sensor/actuator sub-bus containing the signals associated with the specific subsystem of interest (e.g., in this case the International Reactor Innovative and Secure [IRIS] model) and additional variables common among all subsystems within a specific class of subsystems, (b) full sensor/actuator bus, containing all sub-busses from each subsystem used in the model, and (c) illustration of how the organizational structure is interpreted upon simulation.

2.2 TEMPLATES

Two primary templates are used in the current modeling approach: the categorical subsystem template and the template for specific implementation within a subsystem category. These subsystems can be used to create systems of interest to the modeler such as that presented in Figure 4. The following discussion elaborates Modelica's *inheritance and replaceability* capability and its application in the creation of the various templates

2.2.1 Inheritance and Replaceability

Modelica can extend the behavior and properties of an existing model or base class to the construction of a more specific implementation using the *extends* clause. This process allows the reuse or inheritance of data and behavior (variables, equations, etc.) from the base class. Furthermore, the use of inheritance also permits certain modifications to the base class (e.g., variable/parameter changes). As models that share a common base class contain the same underlying requirements, these models can be replaced while retaining all common information, only requiring the modeler to modify new required information. Figure 8 shows an example of the use of inheritance and replaceability. Graphical interface level modification and replaceability is also possible, depending on the development environment used. An extended discussion of these features can be found in Chapter 7 of the Modelica language specification document (Modelica Association 2017).

```
model IRIS
  extends BaseClasses.Partial_SubSystem_A(
    replaceable package Medium = Modelica.Media.Water.StandardWater,
    redeclare replaceable CS_Dummy CS,
    redeclare replaceable ED_Dummy ED,
    redeclare Data.IRIS data,
    W_balance(y=pump.W + liquidHeater.Q_flow),
    FuelConsumption(y=(1 + 0.169)*coreSubchannel.Q_total/(200*1.6e-13*6.022e23/0.235)),
    port_a_nominal(
      p=6.19e6,
      T=497,
      m_flow=502.8),
    port_b_nominal(p=5.8e6, T=591));
```

Figure 8. Example of inheritance and replaceability as used in Modelica: the *extends* clause signals inheritance from the base class; features of the base class are then modified by assigning values, adapting equations, or replacing models with their own individual inheritance structures using the *replaceable* clause.

2.2.2 Specific Subsystem

Fundamental building blocks identified for any specific implementation of a subsystem include the following:

- Physical model
- Nominal/start values of physical variables that cross the model boundary (i.e., connectors)
- Basic design data
- Control system
- Event driver
- Various signals that leave and enter the system
- A location to place additional variables passed to the signal busses
- A location to store components created only for that subsystem
- Base classes required to enable the proper options for inheritance at the overall system level

The building blocks are all encapsulated by the directory structure and the graphical image of a specific sub-system shown in Figure 9. Each building block identified will be further discussed below in the context of Figure 9 to elucidate its implementation and purpose.

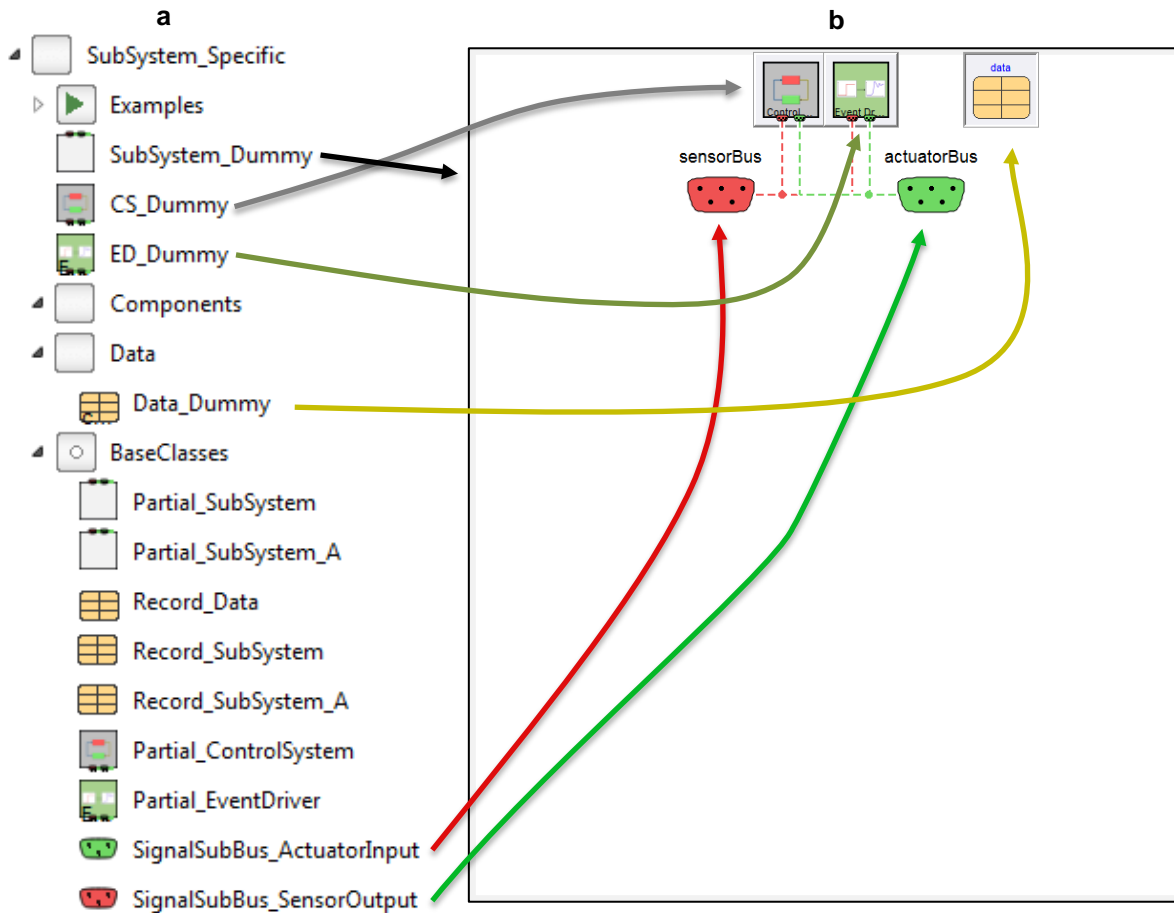


Figure 9 (a) Directory structure of the template of any specific implementation of a subsystem; “BaseClasses” directory establishes models that allow for inheritance from the appropriate subsystem category while limiting the acceptable replaceable models (e.g., control systems) only to the specific sub-system to which they belong, and (b) graphical workspace in which the modeler views base classes for signal busses, control systems, event drivers, and data.

Base Classes for Inheritance

In the directory structure shown in Figure 9, the *BaseClasses* directory contains the models needed to establish a common basis for a specific subsystem, variations of a specific subsystem, and interaction between other categories of subsystems at an overall system level. The *Partial_SubSystem* extends from the categorical partial subsystem, which allows the user to swap out different implementations of subsystems while restricting all other potentially replaceable information to the appropriate location in the model. For example, control systems are typically specific to individual implementations of a subsystem. The control mechanisms for a coal plant will be different from those of a nuclear plant, although they both belong to the *primary heat source* category. Therefore, if a hybrid system is created using a coal plant and then it is desired to switch to a nuclear plant, the available selections of control systems will be changed to those appropriate for the nuclear plant. Figure 10 presents the code that enables this logic.

Partial_SubSystem_A, *Partial_SubSystem_B*, etc., are simple extensions of *Partial_SubSystem* that allow for variations of a model to be created without the need to create a completely new subsystem. For example, two different versions of a nuclear plant could be created with a similar physical model, except one has fluid port connections to an emergency cooling system and connections to a steam generator, while the other only has connections to a steam generator.

```
partial model Partial_SubSystem

  extends
    NHES.Systems.Templates.SubSystem_Category.BaseClasses.Partial_SubSystem(
      redeclare replaceable BaseClasses.Partial_ControlSystem CS
      constrainedby BaseClasses.Partial_ControlSystem,
      redeclare replaceable BaseClasses.Partial_EventDriver ED
      constrainedby BaseClasses.Partial_EventDriver);

  replaceable Record_Data data
    a;

  extends Record_SubSystem;

a
end Partial_SubSystem;
```

Figure 10. Use of *extends* and redeclaration of replaceable models enables a specific subsystem to share a common architecture for development of categorically defined hybrid systems while enabling subsystem-specific models to restrict options to only pertinent selections such as various control systems strategies or specific subsystem records.

Signal Busses and Additional Signals

Each subsystem is provided an additional sub-bus, in addition to the category sub-bus shown in Figure 7. The specific subsystem level is where variables that may be passed between models (i.e., physical model and control system or physical model and the overall system model) are defined. To reiterate the previous discussion of signal busses, an important feature of the expandable connector used to create the signal busses is the ability to ignore unused variables. This is useful, as some control systems may not require the use of all available signals, or variations of a specific subsystem may not have signal variables that completely coincide. These undefined variables in the signal bus will be excluded in the simulation, allowing the simulation to proceed successfully. Figure 11 shows the sensor/output signal sub-bus used in the IRIS model referenced in Figure 7.

```
expandable connector SignalSubBus_SensorOutput

  extends NHES.Systems.Interfaces.SignalSubBus_SensorOutput;

  SI.Power Q_total "Total thermal Output of Reactor";
  SI.Temperature T_Core_Inlet "Core inlet temperature";
  SI.Temperature T_Core_Outlet "Core outlet temperature";
  SI.Pressure p_pressurizer "Pressurizer pressure";

a
end SignalSubBus_SensorOutput;
```

Figure 11. Signal sub-bus of the specific model IRIS within the primary heat system category; the sub-bus enables variables unique to the specific subsystem to be available to other models.

Nominal/Start Values for Connectors

Initialization, the consistent and stable instantiation of interrelated variables determining the initial system condition, is a difficult issue which arises in the creation of dynamic models. This issue is often exacerbated when a modeler attempts to link models together. To improve the possibility of success, whenever physical connections (e.g., fluid connectors) are used in a subsystem to interact with other subsystems, nominal conditions must be provided for the variables associated with the connection, and guess values must be started for the variables which by default should reference the nominal values. Requiring this logic allows improved simulation not only of individual models since appropriate boundary conditions recommendations are then embedded in the model, but it also allows a modeler to verify that boundary conditions of integrated systems are equivalent. It is recommended that components within an implementation properly reference the same nominal and initial conditions as those provided to connections, as consistent initialization conditions are often critical in achieving successful simulation.

An example of this is the use of a fluid port which contains variables for mass flow rate, pressure, and specific enthalpy/temperature. Variable definitions for nominal and start conditions for each fluid connection are included in *Record_SubSystem*. As shown in Figure 10, *Record_SubSystem* is included in *Partial_SubSystem*, which is then extended by a specific model. Figure 8 shows the redefinition of the nominal connection values (i.e., *port_a_nominal* and *port_b_nominal*). If any variable is not redefined, then it must have obtained its default value at the record implementation level. In Figure 8, the *port_b_nominal* default value for mass flow rate is defined as the opposite of the flow rate for *port_a_nominal* at the *Record_SubSystem* level.

Graphical Model Layout Best Practices

In addition to all the features and considerations discussed, graphical implementation of models was given consideration. If a horizontal line is drawn at the level of the signal busses in the right image of Figure 9, all replaceable models and additional variables that may be extracted from components (i.e., using the *RealExpression* component from the Modelica Standard Library) should be contained in the empty space, preferably with replaceable models at the top right of the window and variables at the top left. The remaining portion of the window below the signal busses is for implementation of the model. If additional space is required, the workspace can be extended in the appropriate direction. The standardization of the graphical layout helps ensure that models can be rapidly conceptualized and understood without significant knowledge of the specific subsystem implementation.

2.2.3 Sub-System Category

As previously discussed, nine categories of subsystems have been identified in an HES. Each subsystem is based on a template which does not make any assumption regarding the type of category that may be created. The *BaseClasses* at the category level set the minimal requirements common to all specific implementations within a category as identified by the modeler. The minimal requirements identified are (1) an input and output signal bus, (2) a replaceable control system and event driver, and (3) a record containing any additional information common to all subsystems. At the categorical bus level, each specific subsystem has its specific sub-bus defined. Figure 7 shows the inclusion of the IRIS sub-bus, which is the only specific implementation defined in the primary heat system category at the time of publication. Due to the features of the expandable connector, when a specific subsystem is used, the variables that are defined within its sub-bus will be available, while all other specific sub-busses will not exist.

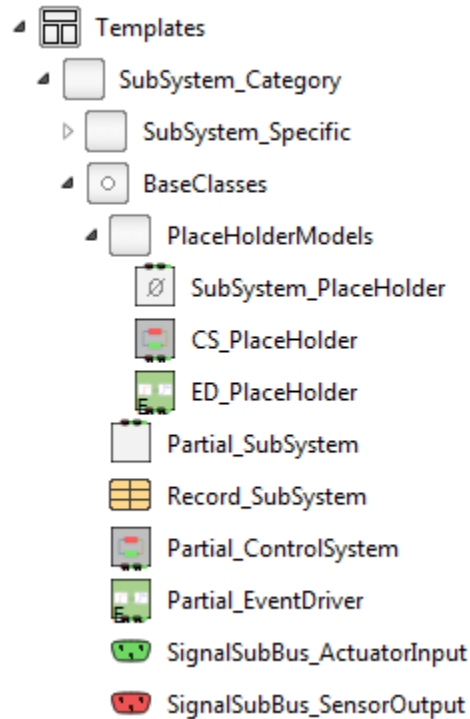


Figure 12. Directory structure of the subsystem category template containing templates for specific subsystems, along with a base class directory providing the common architecture for all specific implementations within the subsystem category.

Place Holder Models

The current templated structure of systems permits subsystems to access variables from other subsystems via signal busses. This functionality was included to provide the greatest flexibility in modeling systems for supervisory control type systems which will likely access variables in a variety of systems to create integrated control methods. As a result, if there are instances when not all models will be used (e.g., an overall system that in one case has energy storage and another case with no energy storage with all else remaining the same), but other subsystems rely on the information defined common to all subsystems in a category, then a dummy model must be available to satisfy the connections with a null value. The *SubSystem_PlaceHolder* model shown in Figure 12 provides the necessary information that permits this behavior. An example of this is provided in Section 3.

2.2.4 Overall Hybrid System

Once subsystem categories and specific implementations have been created, standard implementations that connect all the base information together can be created. For example, the hybrid system that nominally contains nine subsystems (Figure 4) is the example implementation *PartialExample_A* shown in Figure 13.

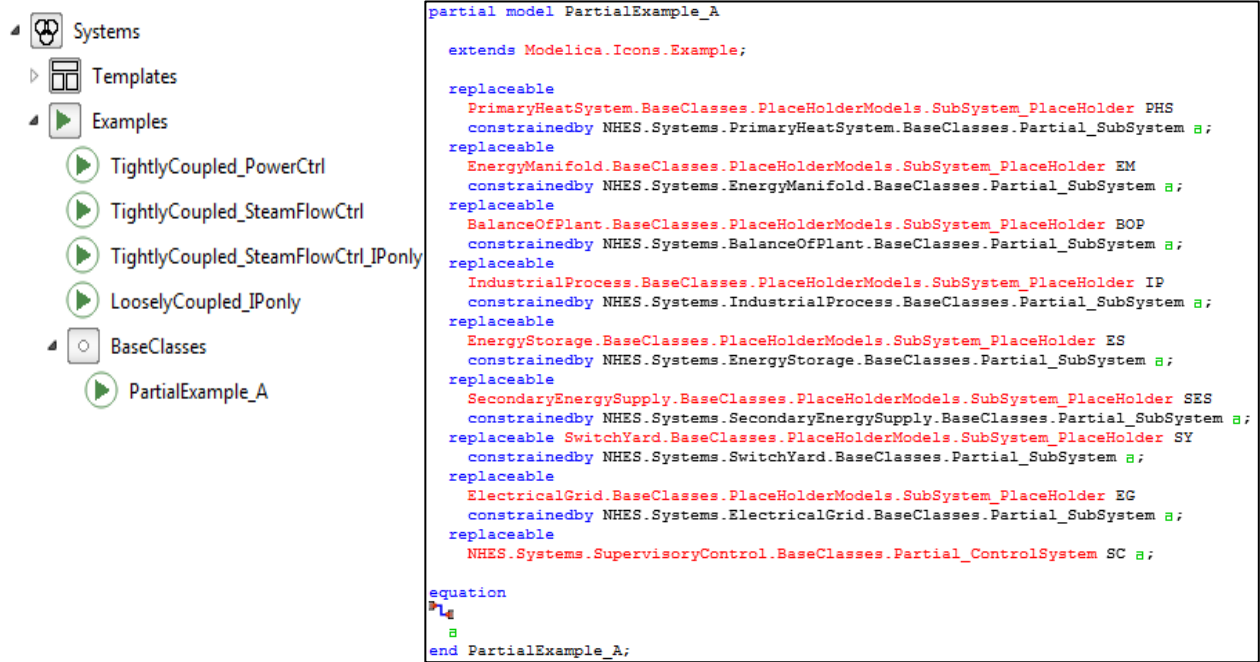


Figure 13. Standard implementations of subsystems generated using inheritance capability to improve tractability and speed of implementing variations of an overall system (Figure 4 is the graphical view of “PartialExample_A”).

Each example included in Figure 13 (e.g., TightlyCoupled_PowerCtrl) extends from *PartialExample_A* and then redeclares each subsystem as required to the specific implementation desired. Once the specific subsystems are chosen, the modeler creates the appropriate connections/boundary conditions while the signal bus connections are automatically accounted for due to the nature of the expandable connector. Nominal and/start values between systems can also be properly referenced to improve probability of simulation success.

3. TEMPLATE USE DEMONSTRATION

Appendix A provides the reader with a step-by-step guide to create subsystem categories and specific implementations using the templated approach. Following the process outlined in the appendix, nine categories with at least one implementation each were created to demonstrate the functionality of the templated approach. Figure 14 presents a blank system architecture constructed from the *Partial_SubSystem* model from each category’s base class package.

To demonstrate the use of the templated approach, two separate case studies using the same architecture were created. The first case (Figure 15) implements a system with only electrical connections between the overall system and the industrial process (i.e., loosely coupled) and with no energy storage or secondary energy supply. The second case (Figure 16) contains a battery model for electrical storage and a natural gas turbine as a secondary energy source. The industrial process is also coupled thermally and electrically (i.e., tightly coupled) with the overall system.

In these test cases, up to three subsystems—balance of plant, secondary energy system, and energy storage—are given demand profile set points from the supervisory control system. Subsystem controls monitor the actual response of the system and provide actuation signals to subsystem components both to match the demand set points and to stabilize the system. The primary heat system (3.1) and energy

manifold (3.2) are discussed in more detail to show how a subsystem may look and will be referenced in the discussion of the results which follows.

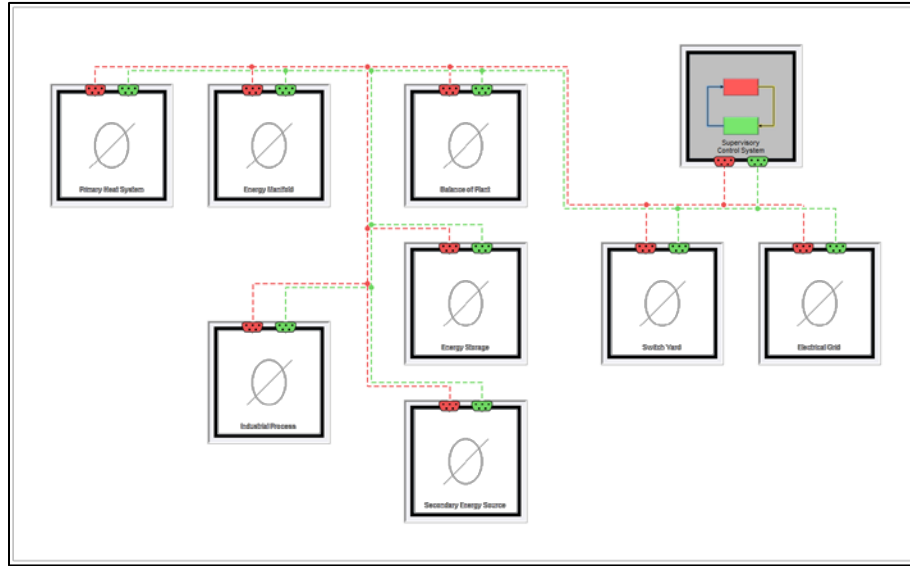


Figure 14. Base Modelica architecture of an HES. Subsystems have been replaced with place holder systems (Figure 4 and Figure 6).

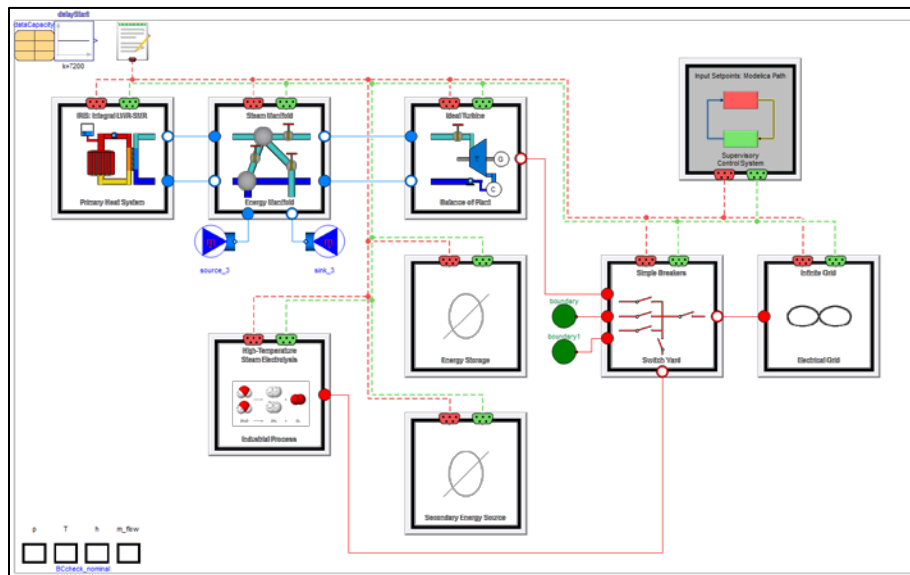


Figure 15. Case 1: Hybrid system based on the template structure using a loosely coupled industrial process: the empty-set symbol indicates that no energy storage or secondary energy supply are in this example.

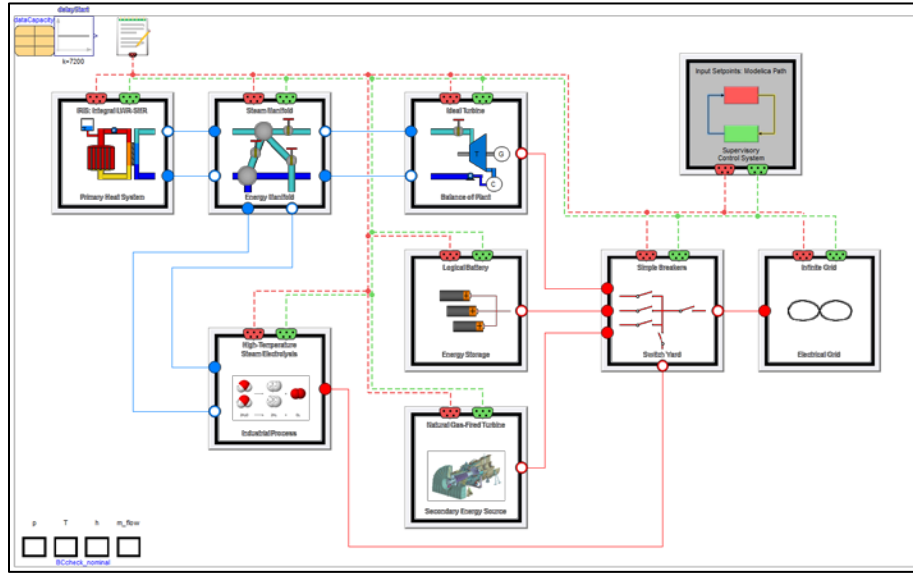


Figure 16. Case 2: Hybrid system based on the template structure using a tightly coupled industrial process.

3.1 PRIMARY HEAT SYSTEM

The primary heat system implemented in this example employs an integral pressurized water nuclear reactor based on the IRIS design (Storrick et al. 2007). Some important physical phenomena captured in the model include the two-phase dynamic interactions of the pressurizer, the generation of steam in a helical coil steam generator, and the reactivity and thermal-hydraulic behavior of the nuclear core. The primary heat system model is shown in Figure 17 with its associated control system in Figure 18. This model integrates the coolant flow geometry and behavior, fuel behavior, and point kinetics neutronics behavior, with feedback from the fuel and coolant temperature.

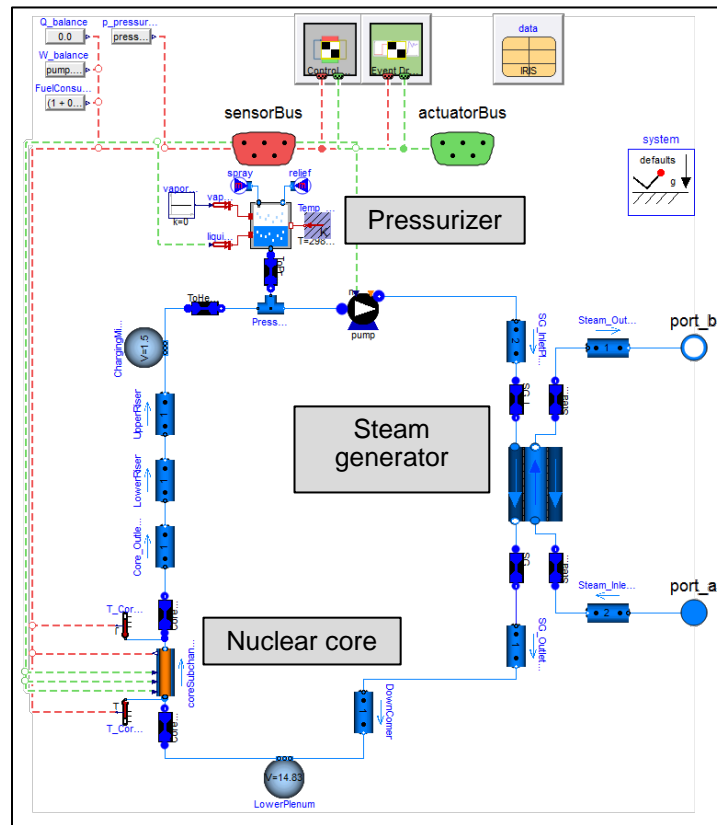


Figure 17. Templated model of the primary heat system based on the IRIS reactor.

The control system monitors the total core power, and it controls the core rod reactivity to hold the primary heat system's thermal power at the nominal state of 1,000 MWt. The control system allows the pressurizer pressure to fluctuate within approximately 0.5 MPa of the nominal condition of 15.5 MPa. The liquid heaters of the pressurizer use a simple on/off controller with hysteresis that adds heat to increase system pressure when the monitored pressure drops below 15.0 MPa.

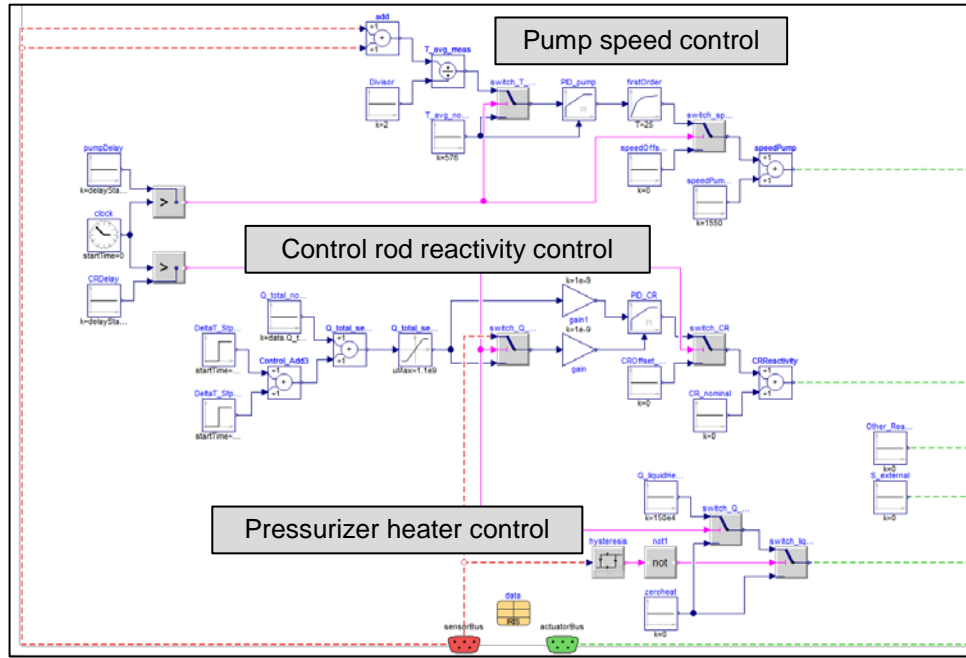


Figure 18. Control system for IRIS primary heat system based on total core power and system pressure.

3.2 ENERGY MANIFOLD

The energy manifold presented is a steam/water distribution manifold with no additional storage or connections (Figure 19). The energy manifold relies on controller logic to actuate distribution valves to handle large and slow power set point changes to other subsystems as specified by the demand profile. This actuation diverts hot steam from the primary heat system to the desired destination. The manifold also gathers return streams and directs the flow back to the primary heat system steam generator at the proper temperature and pressure. Mixing and splitting volumes then add thermal mass to the system, dampening transient behaviors.

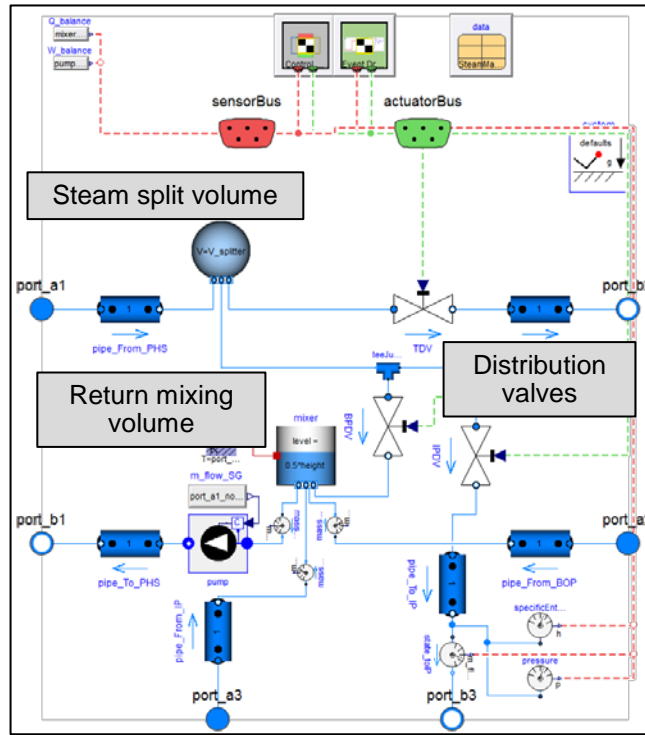


Figure 19. Energy manifold for distributing and gathering steam/water streams from subsystems (bypass valve dampens pressure responses of the system when tracking demand profiles, and the splitter and mixing volumes dampen transient thermal behavior).

The energy manifold control system (Figure 20) monitors the demand power set points of the balance of plant and actuates the turbine distribution valve to track the generated power. The bypass valve acts in a directly opposite response to the turbine distribution valve to limit system pressure swings. The position of the valve in relation to the industrial process is held at a nominal point set by the nominal operating point of the process. The industrial process then controls the flow rate it receives based on the internal pressure drop of the industrial process subsystem model.

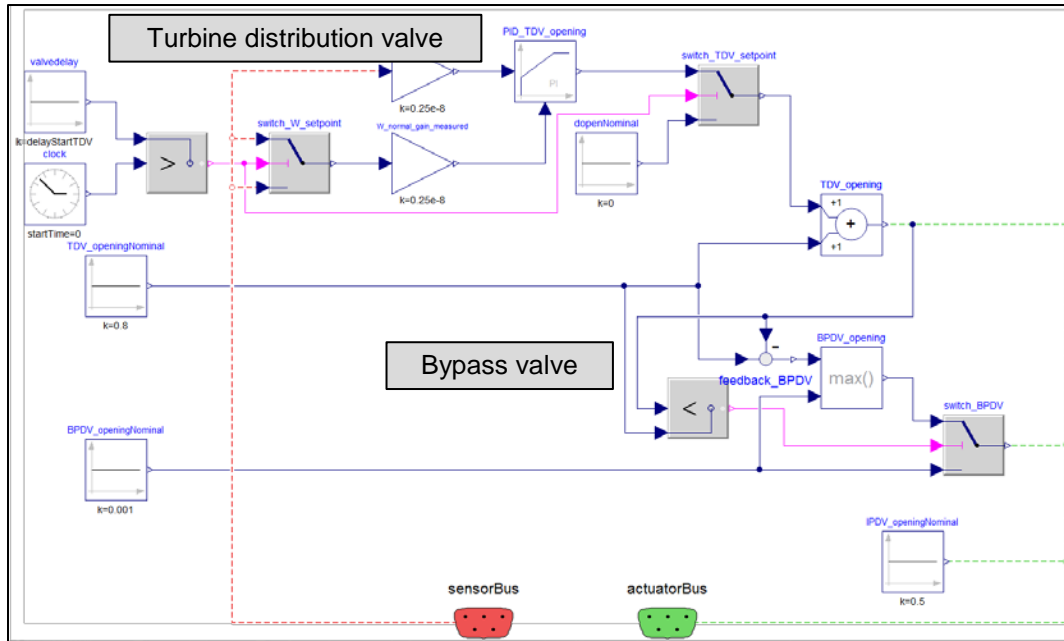


Figure 20. Energy manifold control system with the turbine distribution valve actuated to meet the demand profile set points of the balance of plant, while the bypass valve acts exactly opposite to the turbine distribution valve to dampen pressure swings within the system.

3.3 SELECTION OF BASIC SUB-SYSTEM DYNAMICS

In the presented cases, the electrical grid is modeled as an infinite sink/source of electrical energy. Therefore, when the electrical producing subsystems cannot meet the demands of the electrical consuming subsystems, the overall hybrid system becomes a net consumer of electricity, so the electricity flows from the electrical grid into the system. Results from this type of model have essentially no differences in dynamic behavior of the system (e.g., flow rates, temperature and pressure swings) except for the flow of electrical energy at the electrical grid interface. Therefore, the results presented for the two cases are identical except at the grid interface. Each test case consisted of approximately 15,000 equations and simulated a two-week period in approximately 2.5 hours.

Figure 21 depicts the behavior of the pressurizer pressure in the primary heat system. The pressure slowly decreases over time due to ingress of the cooler primary loop water into the pressurizer. Eventually the pressurizer liquid cools enough that the liquid heater control turns on until the pressure rises back to the nominal 15.5 MPa. This cyclic behavior continues throughout the simulation at a frequency dependent on the dynamics of the energy manifold.

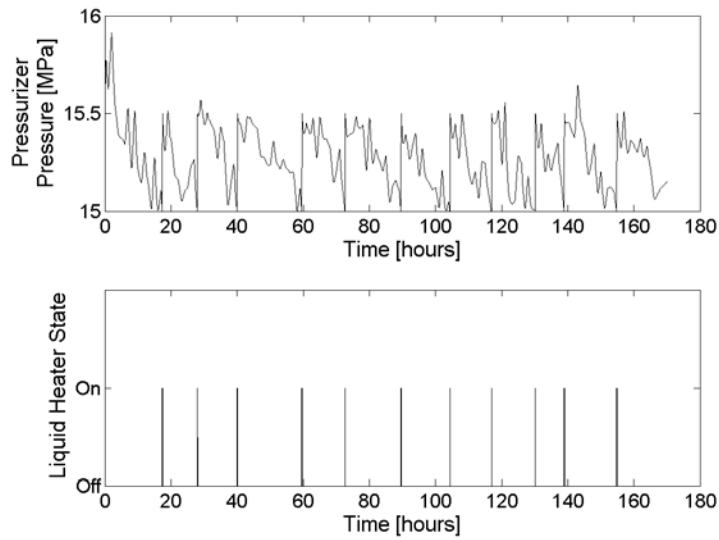


Figure 21. Pressurizer pressure in the primary heat system with oscillatory behavior due to the dynamics of the energy manifold and the liquid heater compensation raising the pressure from the minimum 15.0 MPa to the nominal 15.5 MPa.

Figure 22 demonstrates the ability of the system to be held relatively constant at the nominal thermal operating power of 1,000 MW using only reactivity control. The behavior shown is once again dependent on the behavior of the energy manifold via the dynamic boundary conditions of the steam side of the helical coil steam generator.

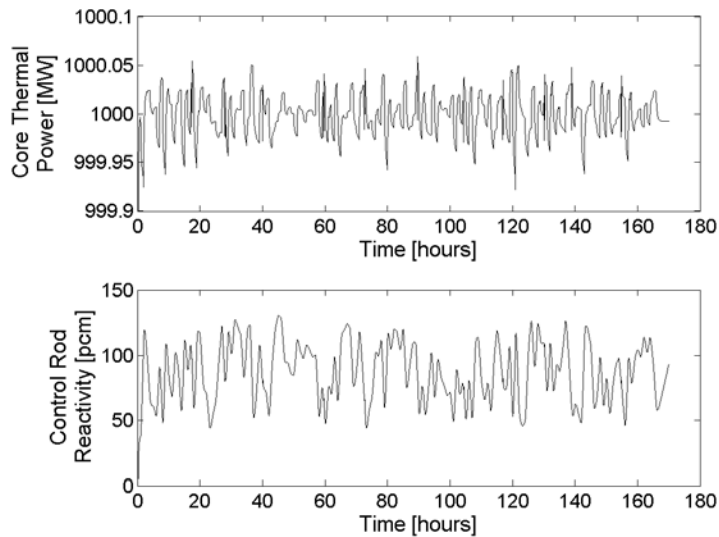


Figure 22. Core thermal power and control rod reactivity behavior necessary to keep the primary heat system operating at a nominal thermal power level of 1,000 MW.

Figure 23 demonstrates the actuation of the valves in the energy manifold required to match the aggressive demand profile set points. The valves act opposite of one another to limit the impacts on other subsystems (e.g., significant pressure deviations from nominal). Physical limitations of the system (e.g., transport delay, pressure limits) prevent an exact following of the set point. However, in addition to physical limitations, the control system parameters also can have a significant impact on the ability of the system to precisely load follow. These types of considerations can lead to the need for alternative control

schemes for the same physical model. Although not demonstrated in this example, the replaceable implementation of control systems allows for this type of investigation.

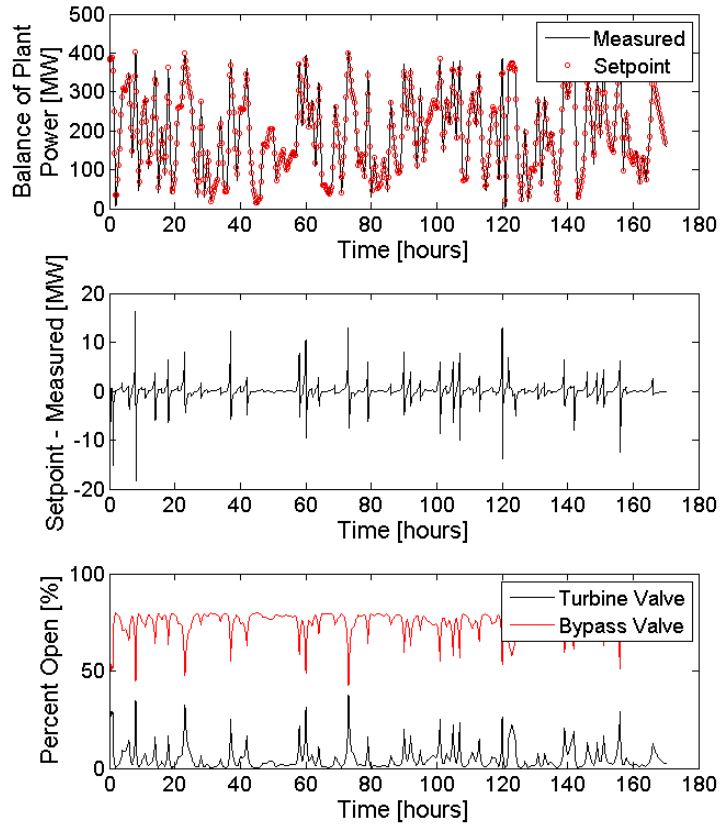


Figure 23. Balance of plant generated power compared to the set point provided by the supervisory control system, in which the dynamic behavior of the turbine and bypass valves work together to meet the specified power and dampen impacts on the connected subsystems.

Figure 24 presents the overall ability of each case to meet the electricity demand profile. The top figure presents a snapshot of approximately one day of the overall electric power provided to the grid. The lower plot provides the deviation from the setpoint power demand signal for each case. The additional subsystems for energy storage and secondary energy supply clearly contribute to significant differences in the interaction of the overall hybrid system with the electrical grid. In these example cases, the lack of energy storage and a secondary energy supply diminish the ability of the hybrid system to provide the power demanded by the grid (positive differential peaks). However, fewer periods of over-production (negative differential peaks) occur. This example demonstrates the dynamic ability of the system, as well as its capabilities to create models in a template format to improve successful model implementation, numerous variables and considerations can alter the model's behaviors.

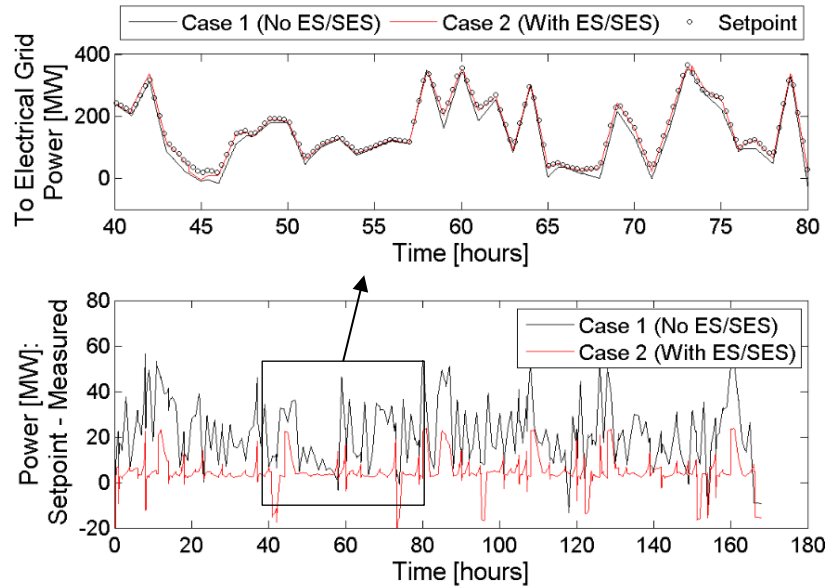


Figure 24. Result of power delivered to the grid compared to the set point demanded; periods of over and under production are evident, and the shift in differential power between the two cases demonstrates impact that variations of system architecture and subsystems can have on HES performance.

4. CONCLUSIONS AND FUTURE WORK

Investigation of HESs requires a modeling environment that permits multi-physics model development and enables use of structured formats to allow contributions from many modelers in a standard approach. For this purpose, a templated modeling approach to facilitate creation of complex HESs was implemented in the Modelica programming language. The templated approach allows the modeler to focus their efforts on individual models by providing the necessary infrastructure for variable communication and model requirements using signal busses and applying Modelica's extends-clause feature. Basing modeling on templates helps facilitate creation of models by a diverse users group and provides a common working environment.

For demonstration purposes, two case studies of HESs and a step-by-step guide (Appendix A) to implementing the template approach are presented. The principles presented are generally applicable to Modelica language; this study may be useful in creation of alternate approaches to modeling, including alternative templated approaches.

Additional subsystem models will be created using the templated approach to evaluate a range of HESs. For example, alternative industrial systems and energy storage devices may improve various factors (i.e., economic or physical performance) which in turn will provide information to investors and policy makers to help inform decisions on future energy systems.

5. ACKNOWLEDGMENTS

This project was funded by the US Department of Energy's Office of Nuclear Energy under the Office of Advanced Reactor Deployment. On behalf of all authors, the corresponding author states that there is no conflict of interest.

6. REFERENCES

- Bragg-Sitton SM, Boardman R, Ruth M, et al (2014) Rethinking the future grid: integrated nuclear renewable energy systems. In: Proceedings of the North Carolina State University 9th Nuclear Plants Current Issues Symposium: Moving Forward. p 14.
- Cetiner SM, Greenwood MS, Harrison TJ, et al (2016) Nuclear Hybrid Energy Systems FY16 Modeling Efforts at ORNL. Oak Ridge National Laboratory, Oak Ridge, TN (United States).
- Dassault Systemes Dymola Systems Engineering. <https://www.3ds.com/products-services/catia/products/dymola/>. Accessed 16 Aug 2017.
- Greenwood MS, Cetiner SM, Fugate DW (2017) Nuclear Hybrid Energy System Model Stability Testing. Oak Ridge National Laboratory, Oak Ridge, TN (United States).
- Harrison TJ, Greenwood MS (2016) Nuclear Hybrid Energy Systems Initial Integrated Case Study Development and Analysis. Oak Ridge National Laboratory, Oak Ridge, TN (United States).
- James R, Hesler S, Bistline J (2015) Fossil fleet transition with fuel changes and large scale variable renewable integration. Electric Power Research Institute, Palo Alto, CA (United States).
- Modelica Association Modelica and the Modelica Association. <https://www.modelica.org/>. Accessed 16 Aug 2017.
- Modelica Association (2017) Modelica Language Specification Version 3.4.
- Storrick GD, Petrovic B, Oriani L (2007) Computer Models for IRIS Control System Transient Analysis. Westinghouse.

APPENDIX A. EXAMPLE GUIDE TO USE THE TEMPLATE

APPENDIX A. EXAMPLE GUIDE TO USE OF THE TEMPLATE

This appendix provides the reader with a step-by-step procedure illustrating the process a modeler can use to take a template to create a new subsystem category and specific implementation and then incorporate those into an overall system model. This procedure example references the directory shown in Figure 5 and will create the *PrimaryHeatSystem* category and the specific implementation called *IRIS*. The focus is on use of the template, so details of physical models and control systems are not discussed.

Modelica-type nomenclature is used in the procedure. Four layers exist in Modelica for creating models: icon, diagram, information, and text layer. The *icon layer* is the top level graphical interface shown when a model is used. The *diagram layer* refers to the component-specific graphical interface where models to be used are dragged and dropped when creating an example. The *information layer* is the space where the modeler adds documentation and notes regarding a model. The *text layer* is where the Modelica code is scripted. Finally, a *package* is the term for the folders of the directory structure. For example, in Figure 5, “Templates,” “Examples,” and “PrimaryHeatSystem” are all packages.

Create the “PrimaryHeatSystem” subsystem category:

1. Duplicate/copy the *SubSystem_Category* package into the *NHES.Systems* package and name it *PrimaryHeatSystem*.
2. Under the newly created *PrimaryHeatSystem* package, rename *SubSystem_Specific* to *IRIS*.
3. Replace the icon of the *PrimaryHeatSystem* with a new icon that fits the category.
4. Navigate to *PrimaryHeatSystem.BaseClasses*, and perform the following actions on the specified model:
 - a. *Partial_SubSystem*
 - i. Go to text layer and scroll down to *annotation* at the bottom. Change the default name—*changeMe*—to *PHS*. This name will be used throughout the subsystem category.
 - ii. On the icon layer, change the black text to Primary Heat System.
 - iii. Add three *RealExpression* models for *Q_balance*, *W_balance*, and *FuelConsumption* with a default value of 0 on the diagram view. Connect them to the signal sensor bus once they have been added to the sensor/output signal bus.
 - b. *Record_SubSystem*, *Record_SubSystem*, *Partial_ControlSystem*, and *Partial_EventDriver*
 - i. No changes needed
 - c. *SignalSubBus_ActuatorInput*
 - i. Add IRIS sub-bus by adding the line given in the Figure A.1.

```
expandable connector SignalSubBus_ActuatorInput
    extends NHES.Systems.Interfaces.SignalSubBus_ActuatorInput;
    NHES.Systems.PrimaryHeatSystem.IRIS.BaseClasses.SignalSubBus_ActuatorInput IRIS;
end SignalSubBus_ActuatorInput;
```

Figure A.1. Primary heat system package actuator/input signal bus.

- d. *SignalSubBus_SensorOutput*
 - i. Add IRIS sub-bus and signal variables by adding the line given in the Figure A.2.

```

expandable connector SignalSubBus_SensorOutput

extends NHES.Systems.Interfaces.SignalSubBus_SensorOutput;

SI.Power Q_balance
  "Heat loss (negative)/gain (positive)
  not accounted for in connections
  (e.g., energy vented to atmosphere)";
SI.Power W_balance
  "Electricity loss (negative)/gain (positive)
  not accounted for in connections
  (e.g., heating/cooling, pumps, etc.)";
SI.MassFlowRate m_flow_fuelConsumption
  "Approximate nuclear fuel consumption rate";

NHES.Systems.PrimaryHeatSystem.IRIS.BaseClasses.SignalSubBus_SensorOutput IRIS;
a
end SignalSubBus_SensorOutput;

```

Figure A.2. Primary heat system package sensor/output signal bus.

Create the IRIS-specific subsystem:

1. Navigate to *PrimaryHeatSystem.IRIS.BaseClasses*, and perform the following actions on the specified model:
 - a. *PlaceholderModels.SubSystem_PlaceHolder*
 - i. Go to text layer and scroll down to the *annotation* at the bottom. Change the default name—*changeMe*—to *PHS*.
 - b. *Partial_SubSystem* and *Partial_SubSystem_A*
 - i. Go to text layer and scroll down to the *annotation* at the bottom. Change the default name—*changeMe*—to *PHS*.
 - c. *Partial_SubSystem_A*
 - i. Go to text layer and scroll down to the *annotation* at the bottom. Change the default name—*changeMe*—to *PHS*.
 - ii. Add two fluid connections on the right edge of the diagram layer, and then modify the diagram layer as shown in Figure A.3.

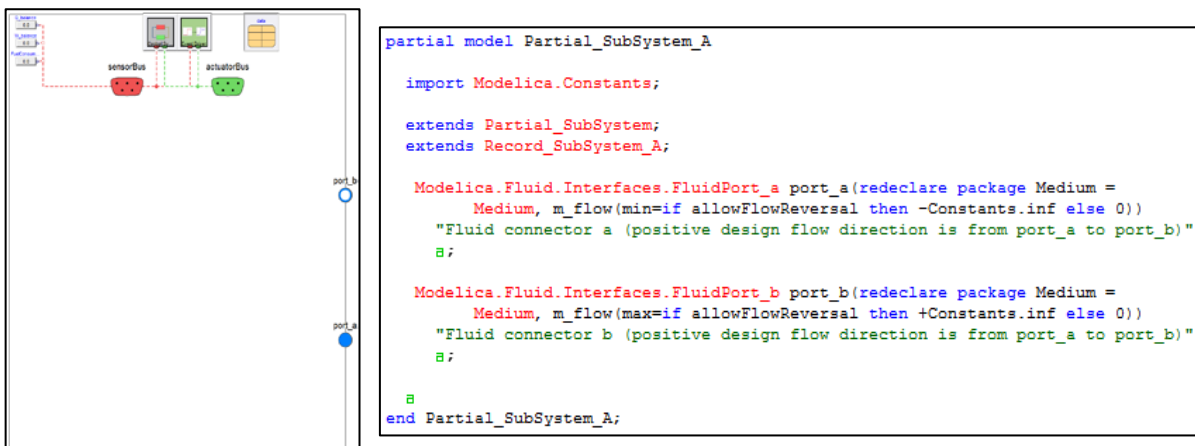


Figure A.3. IRIS *Partial_SubSystem_A* connections.

- d. *Record_SubSystem*, *Record_SubSystem*, *Partial_ControlSystem*, and *Partial_EventDriver*
 - i. No changes needed

e. *Record_SubSystem_A*

- i. Add nominal and start variables for added fluid connections as shown in Figure A.4. Variables that do not have a default assignment will be given one in a later step.

```
partial record Record_SubSystem_A
  extends Record_SubSystem;

  replaceable package Medium = Modelica.Media.Water.StandardWater
    constrainedby Modelica.Media.Interfaces.PartialMedium
    "Medium at fluid ports" a;

  /* Nominal Conditions */
  parameter NHES.Systems.BaseClasses.Record_fluidPorts port_a_nominal(
    redeclare package Medium = Medium, h=Medium.specificEnthalpy(
      Medium.setState_pT(port_a_nominal.p, port_a_nominal.T))) "port_a"
    a;
  parameter NHES.Systems.BaseClasses.Record_fluidPorts port_b_nominal(
    redeclare package Medium = Medium,
    h=Medium.specificEnthalpy(Medium.setState_pT(port_b_nominal.p,
      port_b_nominal.T)),
    m_flow=-port_a_nominal.m_flow) "port_b"
    a;

  /* Initialization */
  parameter NHES.Systems.BaseClasses.Record_fluidPorts port_a_start(
    redeclare package Medium = Medium,
    p=port_a_nominal.p,
    T=port_a_nominal.T,
    h=Medium.specificEnthalpy(Medium.setState_pT(port_a_start.p, port_a_start.T)),
    m_flow=port_a_nominal.m_flow) "port_a"
    a;
  parameter NHES.Systems.BaseClasses.Record_fluidPorts port_b_start(
    redeclare package Medium = Medium,
    p=port_b_nominal.p,
    T=port_b_nominal.T,
    h=Medium.specificEnthalpy(Medium.setState_pT(port_b_start.p, port_b_start.T)),
    m_flow=-port_a_start.m_flow) "port_b"
    a;

  /* Assumptions */
  parameter Boolean allowFlowReversal=true
    "true to allow flow reversal, false restricts to design direction (port_a -> port_b)"
    a;
a
end Record_SubSystem_A;
```

Figure A.4. IRIS *Record_SubSystem_A* nominal and start variables for fluid connections.

f. *SignalSubBus_ActuatorInput*

- i. Add variables that will be connected to the signal bus as shown Figure A.5.

```
expandable connector SignalSubBus_ActuatorInput
  extends NHES.Systems.Interfaces.SignalSubBus_ActuatorInput;

  Real reactivity_ControlRod "Control Rod Drive Mechanism Reactivity";
  Real reactivity_Other "Additional Uncategorized Reactivity";
  SI.Power Q_S_External "External source of power to the Reactor";
  SI.Conversions.NonSIunits.AngularVelocity_rpm speedPump "Rotational speed of coolant pumps";
  SI.Power Q_flow_liquidHeater "Pressurizer heater in the liquid phase";

a
end SignalSubBus_ActuatorInput;
```

Figure A.5. IRIS actuator/input signal sub-bus.

g. *SignalSubBus_SensorOutput*

- i. Add variables that will be connected to the signal bus as shown Figure A.6.

```

expandable connector SignalSubBus_SensorOutput

extends NHES.Systems.Interfaces.SignalSubBus_SensorOutput;

SI.Power Q_total "Total thermal Output of Reactor";
SI.Temperature T_Core_Inlet "Core inlet temperature";
SI.Temperature T_Core_Outlet "Core outlet temperature";
SI.Pressure p_pressurizer "Pressurizer pressure";

a
end SignalSubBus_SensorOutput;

```

Figure A.6. IRIS sensor/output signal sub-bus.

Use the newly created subsystem category and specific implementation

1. Rename *SubSystem_Dummy* to *IRIS*, ensure that the model extends from *Partial_SubSystem_A*, and create the physical model. On the Icon layer, add a distinguishing label and image.

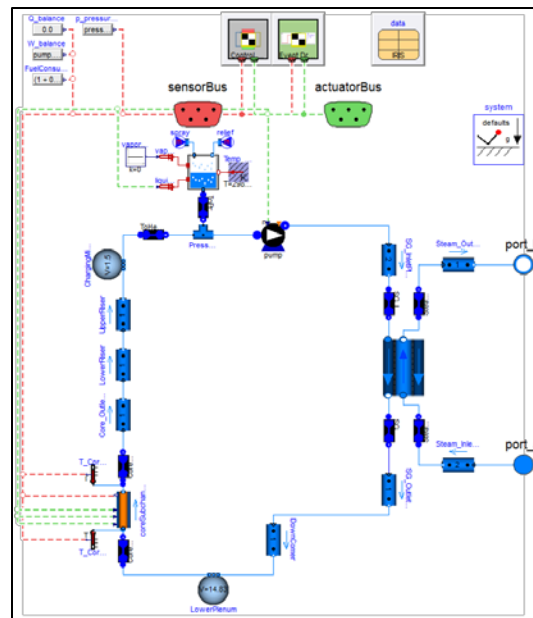


Figure A.7. Specific subsystem physical model IRIS.
Rename *CS_Dummy* to *CS_IRIS*, and create the control system.

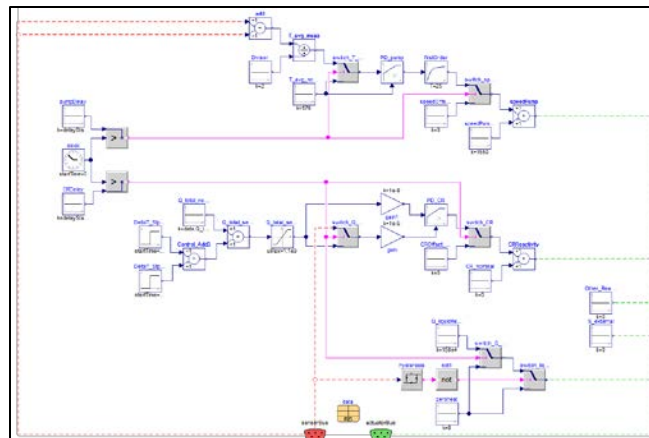


Figure A.8. Specific subsystem IRIS control system.

2. Add the newly created model to an example by right clicking the location where it belongs (i.e., box 1 from Figure 4), and connect the fluid connectors.

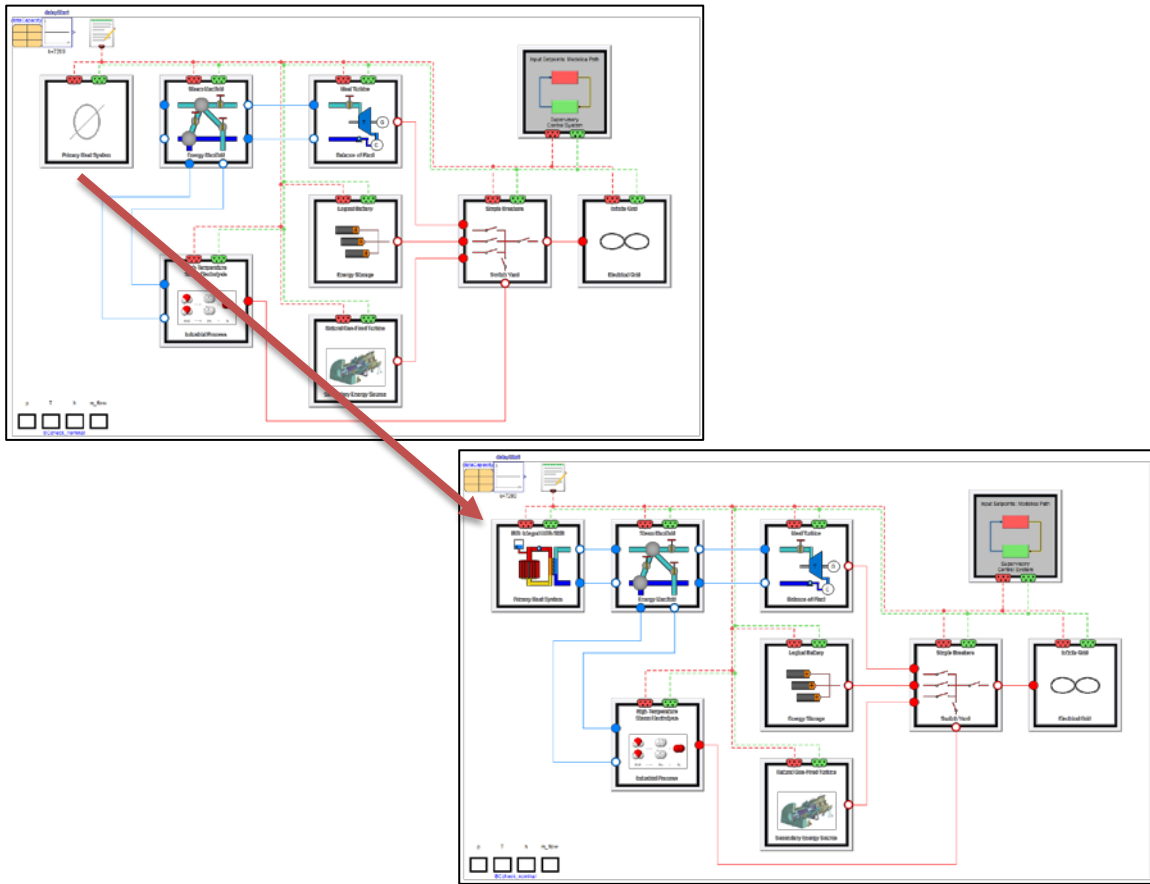


Figure A.9. Add new model by right clicking the base model and choosing Change Class>All Matching Classes>IRIS.

