INL/RPT-22-66292
Revision 0

Status on the Development of the Infrastructure for a Flexible Modelica/RAVEN Framework for IES

March 2022

Joshua Cogliati
Paul Talbot
Konor Frick
Idaho National Laboratory

IES
Integrated Energy Systems

INL/RPT-22-66292
Revision 0

Status on the Development of the Infrastructure for a Flexible Modelica/RAVEN Framework for IES

Joshua Cogliati
Paul Talbot
Konor Frick
Idaho National Laboratory

March 2022

Idaho National Laboratory
Integrated Energy Systems
Idaho Falls, Idaho 83415

http://www.ies.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

ABSTRACT

Since the previous report, the proof-of-principle functional mock-up unit generation has been made more robust and merged into the Risk Analysis and Virtual Environment (RAVEN). Now, RAVEN has added the ability to generate functional mock-up units from ExternalModels and reduced-order models. This ability has been tested on multiple operating systems and regression tests before being added to RAVEN.

CONTENTS

1. INTRODUCTION....5
   1.1 FORCE Ecosystem....5
   1.2 FMU and FMI....5
2. FORCE Work....5
   2.1 Serialization....6
   2.2 FMU and FMI Work....6
3. Future Work....6
   3.1 Model Exchange....7
   3.2 Grey Model....7
4. REFERENCES....7

TABLES

Table 1: Added IOStep Input and Output (copied from RAVEN User Manual)....6

ACRONYMS

FMI	functional mock-up interface
FMU	functional mock-up unit
FORCE	Framework for Optimization of ResourCes and Economics
IES	Integrated Energy Systems
INL	Idaho National Laboratory
RAVEN	Risk Analysis and Virtual Environment
ROM	reduced-order model

Status on the Development of the Infrastructure for a Flexible Modelica/RAVEN Framework for IES

1. INTRODUCTION

Idaho National Laboratory (INL) has been developing models and analysis and optimization tools for the Department of Energy Office of Nuclear Energy as part of the Integrated Energy Systems program. As part of this effort, it is useful to enhance the ability to exchange and transfer both reduced-order and existing Python models between different simulation environments. This report is on the recent progress and future directions of the efforts related to functional mock-up units (FMUs), functional mock-up interfaces (FMIs), and the serialization of models.

1.1 FORCE Ecosystem

INL's Framework for Optimization of ResourCes and Economics (FORCE) ecosystem is used to solve system- and grid-level optimization problems. RAVEN[1] is part of FORCE and is designed to perform parametric and probabilistic analyses based on the response of complex system codes. As part of FORCE, INL has been developing a library of high-fidelity process models in the Modelica modeling language.

1.2 FMU and FMI

The FMI standard[6] describes an open format for exporting and importing simulation models using a common data exchange nomenclature. In other words, the FMI standard allows the user to retain the same model while selecting the tools best suited for each type of analysis.

In order to be executed, an FMI is always "shipped" with an FMU. An FMU is the executable that implements the FMI. During the exportation of an FMU, an FMU archive is generated from a systems model, whereas during an FMU import, a systems model is generated from an FMU archive.

2. FORCE Work

The model serialization and use of FMUs and FMIs in FORCE have made substantial progress in the year since the previous report.[2] Both new serialization features and an FMU exporter have been added to RAVEN.

2.1 Serialization

The new serialization features [3] added to the RAVEN code has added the ability to export external models as pickled files and input pickled external models. This adds a new way for RAVEN to run existing models. The added input and output from the serialization and the FMU export is summarized in Table 1. The serialization adds the ability for RAVEN external models to differentiate between input and output, instead of just specifying variables, which is required for creating the interface for exporting FMU models. In addition, the Steps code inside RAVEN has been refactored into multiple files to be easier to work with. These features pave the way for the FMU export ability.

Table 1. Added IOStep input and output (copied from RAVEN User Manual).

Input	Output	Resulting Behavior
File	**ExternalModel**	Load On-Disk Serialized ExternalModel
ExternalModel	**File**	Serialize ExternalModel to Disk
ROM	**File**	If *type* is fmu, Serialize ROM to FMU
ExternalModel	**File**	If *type* is fmu, Serialize ExternalModel to FMU

2.2 FMU and FMI Work

The FMU exporter work has been made more robust and merged into RAVEN.[4] Continuing the serialization work, RAVEN now can export both external models and ROMs as FMUs. These FMUs have the FMI inside to specify the inputs and output for using the FMU for co-simulation. This export feature has been tested on Mac OS, Windows, and multiple versions of Linux.[a] In order to support these platforms, the PythonFMU package (placed in the RAVEN contrib directory) had to be extended[b] to support Mac OS (it had previously only supported Windows and Linux). A build script has been created for building PythonFMU as part of the regular RAVEN compiling. The export code works both when RAVEN is installed with Miniconda and installed with PIP.

Four regression tests have been added to the RAVEN regression tests to test both the creation and use of RAVEN-created FMUs. The first test uses PythonFMU with RAVEN to create an FMU based on an external model. The second test uses PythonFMU with RAVEN to create an FMU based on a RAVEN ROM. The last two tests run scripts without using RAVEN. Instead, one test uses fmpy to test the loading and running of an FMU created from an external model and the other uses fmpy to test the loading and running of an FMU created from a ROM.

[a] Note that it did not work on CentOS 7 with the default C++ compiler due to the compiler missing features.

[b] An issue has been filed with PythonFMU explaining RAVEN's modifications.

3. Future Work

There is further work to be done with the flexible Modelica/RAVEN framework for the Integrated Energy Systems program.

3.1 Model Exchange

Currently, the FMU exporter in RAVEN only exports co-simulation FMUs. It would be useful to also be able to export model exchange FMUs. Model exchange provides the model equations, unlike co-simulation which only exports a black box that communicates at predetermined timesteps. This allows model exchange FMUs to have the importing tool support dynamic time stepping and different solver methods. Some ROMs (such as linear regression) calculate coefficients for equations that could be used for model exchange.

3.2 Grey Model

As part of a separate project, the use of PyNumero from Pyomo to generate first and second derivatives from RAVEN-generated ROMs has been completed and is currently being merged into RAVEN.[5] As future work, integrating this work with the FMU exporter would allow more uses of the FMUs. One use for grey models is for creation of model exchange FMUs when the ROMs or external models cannot otherwise easily generate a usable equation.

4. REFERENCES

[1] Alfonsi, A., C. Rabiti, D. Mandelli, J. Cogliati, C. Wang, P. W. Talbot, D. P. Maljovec, and C. Smith. 2016. "RAVEN Theory Manual and User Guide." INL/EXT-16-38178, Idaho National Laboratory. https://doi.org/10.2172/1260312.

[2] Frick, K., A. Alfonsi, C. Rabiti, and S. Bragg-Sitton. 2021. "Development of the IES Plug-and-Play Framework." INL/EXT-21-62050, Rev 00, Idaho National Laboratory.

[3] Cogliati, J. 2021. "Cogljj/serialization exporter #1608." https://github.com/idaholab/raven/pull/1608.

[4] Alfonsi, A. 2021. "Exporter for FMI/FMU #1481." https://github.com/idaholab/raven/pull/1481.

[5] Alfonsi, A. 2021. "Alfoa/rom to pyomo #1482." https://github.com/idaholab/raven/pull/1482.

[6] FMI Development Group, 2021. "Functional Mock-up Interface for Model Exchange and Co-Simulation" https://github.com/modelica/fmi-standard/releases/download/v2.0.3/FMI-Specification-2.0.3.pdf