INL/EXT-17-43222

MOOSE IPL Extensions

(Control Logic)

Cody Permann

September 2015

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed 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. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/EXT-17-43222

MOOSE IPL Extensions

(Control Logic)

Cody Permann

September 2015

Idaho National Laboratory
Modeling and Simulation
Idaho Falls, Idaho 83415

http://www.inl.gov

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

Summary

In FY-2015, the development of MOOSE was driven by the needs of the NEAMS MOOSE-based applications, BISON, MARMOT, and RELAP-7. An emphasis was placed on the continued upkeep and improvement MOOSE in support of the product line integration goals. New unified documentation tools have been developed, several improvements to regression testing have been enforced and overall better software quality practices have been implemented. In addition the Multiapps and Transfers systems have seen significant refactoring and robustness improvements, as has the "Restart and Recover" system in support of Multiapp simulations. Finally, a completely new "Control Logic" system has been engineered to replace the prototype system currently in use in the RELAP-7 code. The development of this system continues and is expected to handle existing needs as well as support future enhancements.

CONTENTS

ACRONYMS

CASL – Consortium for the Advanced Simulation of LWRs

CL – Control Logic

DOE – Department of Energy

FY – Fiscal Year

MOOSE – Multiphysics Object Oriented Simulation Environment

NEAMS – Nuclear Engineering Advanced Modeling and Simulation

NQA-1 – Nuclear Quality Assurance Level 1

RELAP – Reactor Excursion and Leak Analysis Program

MOOSE IPL Extensions

1. Control Logic System

Currently MOOSE supports a few methods for varying parameters during a simulation. The most common use is to create a function dependent on space and time to vary a particular parameter over the course of the simulation. Additionally MOOSE allows for user to build more complex functions containing coupled parameters that allow individual parameters to change based on the state of other variables or parameters within the simulation. This system however becomes very cumbersome and requires applications to modify their parameter systems to simulate control systems or sensitivity analysis. Further, without a support system in the framework, each application is responsible for defining when and how these parameters may be changed and to what extent. This leads to difficult support issues and maintenance for each application making use of these ad-hoc methods. To overcome these difficulties, a new “Control Logic” (CL) system has been designed to provide a uniform API that will enable all developers and users to control various parameters in a consistent yet flexible manner.

The design of a flexible control logic system required a significant redesign of the way MOOSE handles all user-defined input parameters. New centralized input parameter storage was created and all ownership of parameters was seamlessly transferred to the framework instead of residing throughout the various objects created by user simulations. This task was successfully implemented and deployed to all users in July. After that the control logic system was deployed in early September, which laid out the framework for which all control logic objects would be built. A developer would be allowed to create user-defined objects that could control any number of designated parameters throughout the system during various parts of the simulation execution. The key feature of the system is the design of how parameters are advertised to the Control Logic systems as candidates for control. In the new Control Logic system, parameters may be tagged. Parameters may tagged individually, in arbitrary groups, and logically as given in the input file hierarchy. Using these tags, control logic objects may change values of complete sets of parameters throughout the simulation at designated times. Figure 1. Shows the new Control Logic syntax linking a parsed function to a coefficient in another MOOSE object. A Postprocessor was also setup to track the coefficient as changed by the CL system to indicate successful function.

Future developments in the control logic system may see additional notifications to the user as parameters change, range checking for parameters that may only be valid for defined ranges and scripting support. Existing control logic systems currently in use in MOOSE-based applications will be replaced during FY16.

```
[Functions]
  [./func_coef]
    type = ParsedFunction
    value = '2*t + 0.1'
  [../]
[]

[Controls]
  [./func_control]
    type = RealFunctionControl
    parameter = 'coef'
    function = 'func_coef'
    execute_on = 'initial timestep_begin'
  [../]
[]
```

```
+----------------+----------------+
| time           | coef           |
+----------------+----------------+
|   0.000000e+00 |   0.000000e+00 |
|   1.000000e-01 |   3.000000e-01 |
|   2.000000e-01 |   5.000000e-01 |
|   3.000000e-01 |   7.000000e-01 |
|   4.000000e-01 |   9.000000e-01 |
|   5.000000e-01 |   1.100000e+00 |
|   6.000000e-01 |   1.300000e+00 |
|   7.000000e-01 |   1.500000e+00 |
|   8.000000e-01 |   1.700000e+00 |
|   9.000000e-01 |   1.900000e+00 |
|   1.000000e+00 |   2.100000e+00 |
+----------------+----------------+
```

Figure 1. Control Logic syntax and resulting output of parameter control