International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)
Sun Valley, Idaho, USA, May 5-9, 2013, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2013)

TITLE OF THE PAPER

A. Alfonsi, C. Rabiti, D. Mandelli, J.J. Cogliati, R.A. Kinoshita
Idaho National Laboratory
2525 Fremont Avenue, Idaho Falls, ID 83415
{andrea.alfonsi, cristian.rabiti, diego.mandelli, joshua.cogliati, robert.kinoshita}@inl.gov

ABSTRACT

To be added

Key Words: Reactor Simulation, Probabilistic Risk Assessment, Dynamic PRA, Monte-Carlo, Relap-7

1. INTRODUCTION

RAVEN (**R**eactor **A**nalysis and **V**irtual control **EN**viroment) is a complex software tool that acts as the control logic driver for RELAP-7. The goal of this paper is to highlight the software structure of the code and its utilization in conjunction with the newly developed Thermo-Hydraylic code RELAP-7. RAVEN is a software framework that allows dispatching the following functionalities:

- Derive and actuate the control logic required to:
  - Simulate the plant control system;
  - Simulate the operator actions (guided procedures);
  - Perform Monte-Carlo sampling of random distributed events;
  - Perform event tree based analysis.
- Provide a Graphical User Interface (GUI) to:
  - Input a plant description to RELAP-7(components, controlled variables, controlled parameters);
  - Concurrent monitoring of Controlled Parameters;
  - Concurrent alteration of Controlled Parameters.
- Provide a post-processing data mining capability based on:
  - Dimensionality reduction;
  - Cardinality reduction.

The paper is divided in three main sections:

- RAVEN mathematical framework;
- RAVEN software structure;
- Demonstration of a Station Black Out (SBO) analysis of a Pressurized Water Reactor (PWR).

2. MATHEMATICAL FRAMEWORK

In the following paragraphs the mathematical framework is briefly described, analyzing the set of the equations needed to model the control system in a nuclear power plant.

2.1. Plant and Control System Model

The first step will be the derivation of the mathematical model representing, at a high level of abstraction, the plant and control system model. Let be $\bar{\theta}(t)$ a vector describing the plant status in the phase space, and the governing equation:

$$\frac{\partial \bar{\theta}}{\partial t} = \bar{H}(\theta(t), t) \tag{1}$$

In the above equation we have assumed the time differentiability in the phase space. This is in generally not required and it is abused here for compactness of the notation. Now an arbitrary decomposition of the phase space is performed:

$$\bar{\theta} = \begin{pmatrix} \bar{x} \\ \bar{v} \end{pmatrix} \tag{2}$$

The decomposition is made in such a way that x represent the unknowns solved by RELAP-7, while v are the variables directly controlled by the control system . The governing equation is now casted in a system of equations:

$$\begin{cases} \dfrac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}, t) \\ \dfrac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}, t) \end{cases} \tag{3}$$

As a consequence of this splitting, the contains components state variables of the phase space now that are continuous while contains all variables describing the discrete state variables that are of the system usually handled by the control system. As a next step, we realize that the function $\bar{V}(\bar{x}, \bar{v}, t)$ representing the control system, does not depend on the knowledge of the complete status of the system but on a restricted subset that we call control variables $\bar{C}$:

$$\begin{cases} \dfrac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}, t) \end{cases} \tag{4}$$

where $\bar{C}$ is a vector which having lesser dimensionality than $\bar{x}$ and therefore is more convenient to work with. In rest of this document the following naming will be adopted: $\bar{C}$: monitored variables $\bar{v}$: controlled variables. Note that even if it seems more appropriate, the standard naming of signals (monitored) and status (controlled) is not yet used. The reason for this choice is that, the chosen naming better mirrors the computational pattern between RAVEN and RELAP 7 and moreover the definition of signals is more tight to the definition of the control logic for each component and therefore relative rather than absolute in the overall system analysis. In fact we could have signal for a component that are status of another creating a definition that would be not unique. Another reason is that the standard naming will loose every meaning once used also for uncertainty analysis.

2.2. Operator Splitting Approach

System of equations shown in Eq. 4 is, generally speaking, fully coupled and in the past it has always been solved following an operator splitting approach. The reasons for this choice are several:

- Control system reacts with an intrinsic delay anyhow;
- The reaction of the control system might move the system between two different discrete state and therefore numerical errors will be always of first order unless the discontinuity will be treated explicitly.

RAVEN is, thus, using this approach. Therefore Eq. 4 becomes:

$$\begin{cases} \dfrac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i - 1}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i - 1}, t) \end{cases} \tag{5}$$

2.3. The auxiliary plant and component status variables.

So far it has been assumed that all information needed are contained in $\bar{x}$ and $\bar{v}$. Even if, as previously shown, these information are sufficient for the calculation of the system status in every point in time, it is not a practical and efficient way to implement the control system. In order to facilitate the implementation of the control logic, it's been implemented a system of auxiliary variables. The auxiliary variables are those that in statistical analysis are artificially added to non-Markovian system into the space phase to obtain back a Markovian behavior, so that only the previous time step information are needed to determine the future status of the system These variables can be classified into two types:

- Global status auxiliary control variables: scram signal, time at which scram event begins, hot shut down, time at which hot shut down event begins, cold shut down, etc.;
- Component status auxiliary variables like: correct operating status, time from abnormal event, etc.

Thus, the introduction of the auxiliary system into the mathematical framework leads to the following formulation of the Eq. 5:

$$\begin{cases} \dfrac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i - 1}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial \bar{a}}{\partial t} = \bar{A}(\bar{x}, \bar{C}, \bar{a}, \bar{v}_{t_i - 1}, t) \\ \dfrac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i - 1}, t) \end{cases} \tag{6}$$

3. SOFTWARE STRUCTURE

RAVEN is a C++/Python software, coded in an high modular and object-oriented way and based on two software sections:

- MOOSE(**M**ultiphysics **O**bject-**O**riented **S**imulation **E**nviroment);
- RELAP-7.

3.1. MOOSE/RELAP-7.

MOOSE is a computer simulation framework, developed at Idaho National Laboratory (INL), that simplifies the process for predicting the behavior of complex systems and developing non-linear, multiphysics simulation tools. As opposed to traditional data-flow oriented computational frameworks, MOOSE is founded on the mathematical principle of Jacobian-free Newton-Krylov (JFNK) solution methods. Utilizing the mathematical structure present in JFNK, physics are modularized into Kernels allowing for rapid production of new simulation tools. In addition, systems are solved fully coupled and fully implicit employing physics based preconditioning which allows for great flexibility even with large variance in time scales. In addition to provide the algorithms for the solution of the differential equation, MOOSE also provides all the manipulation tools for the C++ classes containing the solution vector. This framework has been used to construct and develop the Thermo-Hydraylic code RELAP-7, giving a enormous flexibility in the coupling procedure with RAVEN. RELAP-7 is the next generation nuclear reactor system safety analysis. It will become the main reactor systems simulation toolkit for RISMC (**R**isk **I**nformed **S**afety **M**argin **C**haracterization) and the next generation tool in the RELAP reactor safety/systems analysis application series (the replacement for RELAP5). The key to the success of RELAP-7 is the simultaneous advancement of physical models, numerical methods, and software design while maintaining a solid user perspective. Physical models include both PDEs (Partial Differential Equations) and ODEs (Ordinary Differential Equations) and experimental based closure models. RELAP-7 will eventually utilize well posed governing equations for multiphase flow, which can be strictly verified.RELAP-7 uses modern numerical methods, which allow implicit time integration, higher order schemes in both time and space, and strongly coupled multi-physics simulations. RELAP-7 is the solver for the plant system except for the control system. From the mathematical formulation presented so far, RELAP-7 solves $\frac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i-1}, t)$. The nuclear power plant is represented and modeled by a set of components (Pipes, Valves, Branches, etc.) and each component type corresponds to a C++ class.

3.2. RAVEN .

As briefly mentioned, RAVEN has been coded in high modular and pluggable way in order to make easier the integration of different program languages (C++, Python) and coupling with other applications based on MOOSE and not. The code is constituted by two modules:

- RAVEN/RELAP-7 interface;
- Python Control Logic.

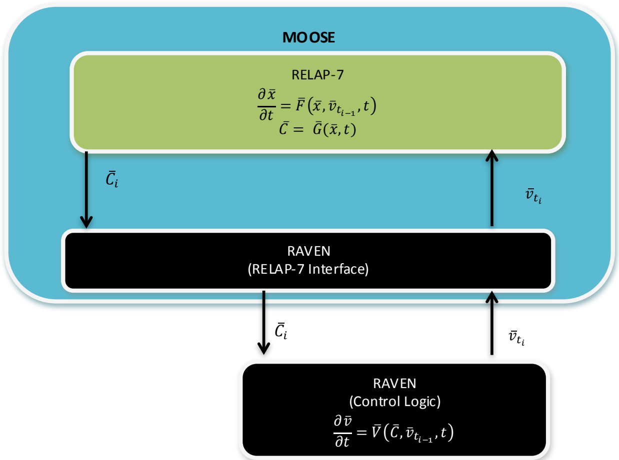

Figure 1. Control System Software Layout.

3.3. Software layout and calculation flow.

At each time step RELAP-7/MOOSE updates the information within the classes with the current solution $\bar{x}$, then RAVEN will ask MOOSE to perform the needed manipulation to construct the monitored quantities $\bar{C}$. Once $\bar{C}$ is constructed, the information is reduced to a vector of numbers understandable by the control system. The equation $\frac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i-1}, t)$ is solved and the set of control parameters for the next time step v_{t_i} is obtained. Up to now no situations required a numerical solution of the equation $\frac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i-1}, t)$, therefore for the moment RAVEN remains numerical integration free. Once the information is transferred to C, the way through which the plant solution x is computed or stored is irrelevant. The last statement highlights the capability of RAVEN to represent an easily generalizable tool. This functional scheme is represented in Figure 1. To be more specific, in reality MOOSE is made aware of the need to compute at the end of each time step the C as a consequence this is immediately available at the end of each time step. As a consequence scheme in Figure ?? is more accurate in terms of software implementation. In the following of the discussion depending of which aspect will be more relevant either scheme in Figure 1, Figure 2 or Figure 3 will be referred.

4. CONCLUSIONS

Fin!

REFERENCES

[1] B. Author(s), “Title,” *Journal Name in Italic*, **Volume in Bold**, pp. 34-89 (19xx).

[2] C. D. Author(s), “Article Title,” *Proceedings of Meeting in Italic*, Location, Dates of Meeting, Vol. n, pp. 134-156 (19xx).

[3] E. F. Author, *Book Title in Italic*, Publisher, City & Country (19xx).

[4] “Spallation Neutron Source: The next-generation neutron-scattering facility in the United States,” http://www.sns.gov/documentation/sns_brochure.pdf (2002).

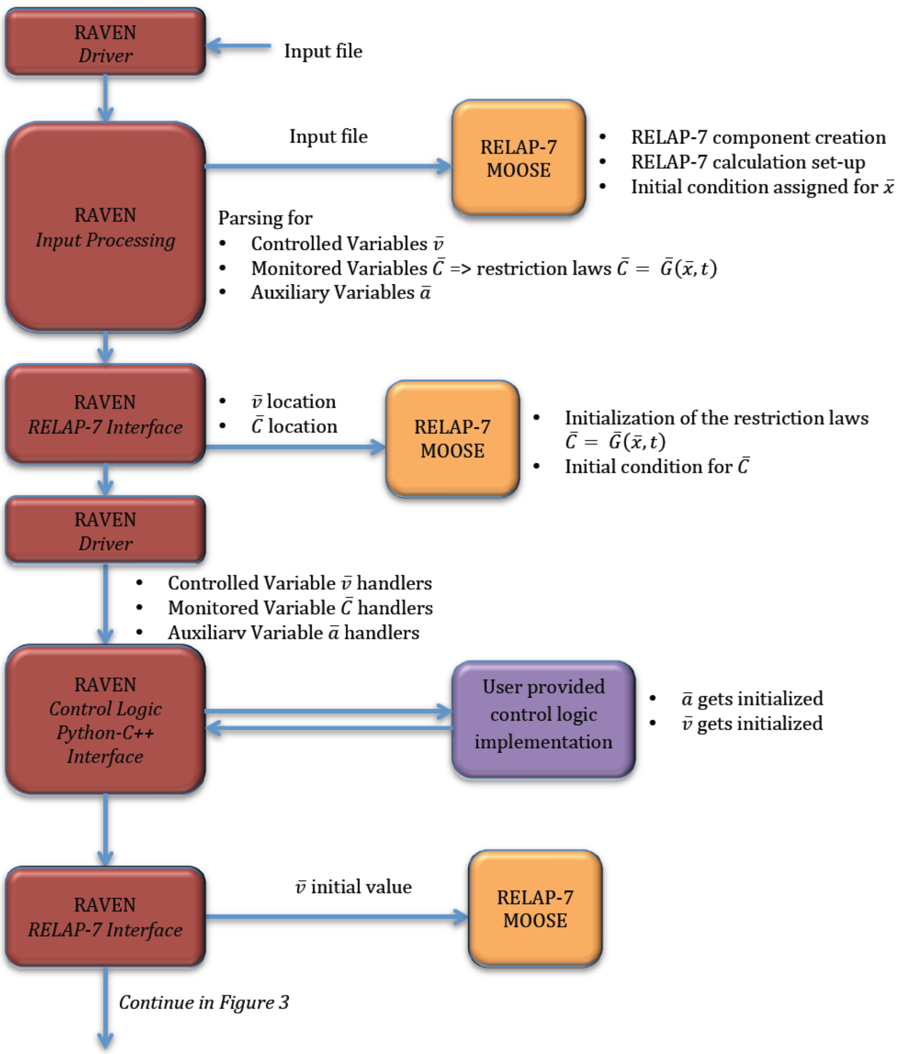

Figure 2. Control System Software Layout.

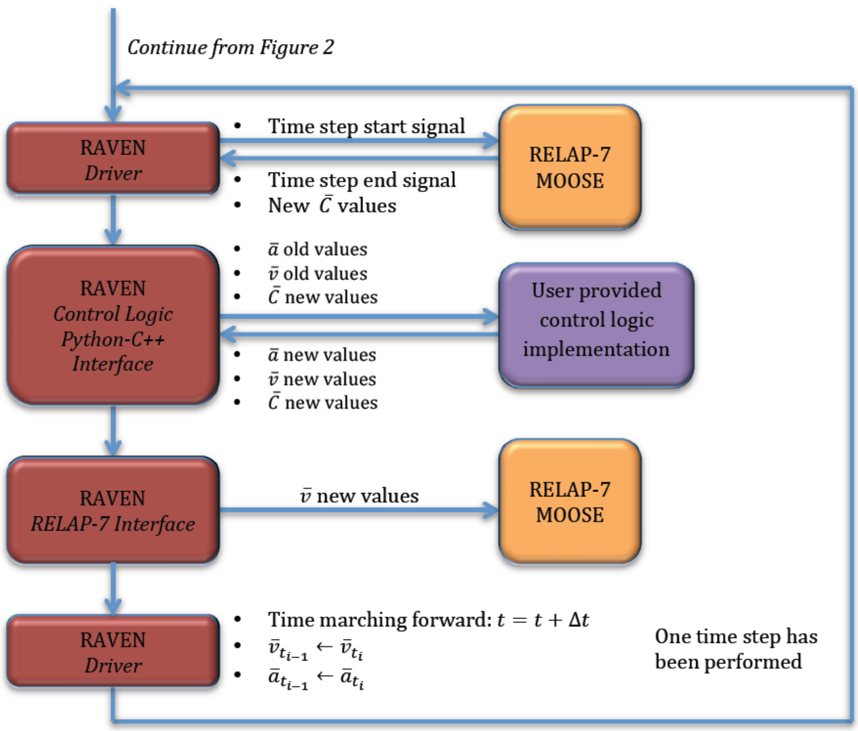

Figure 3. Control System Software Layout.