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@inl.gov; Cristian.Rabiti@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:

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

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 $\bar{C}$:

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

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:

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

3. SOFTWARE STRUCTURE

RELAP-7 is the solver for the plant system except for the control system. From the mathematical formulation presented so far, RELAP-7 will solve $\frac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i - 1}, t)$. RELAP-7 will be based on the

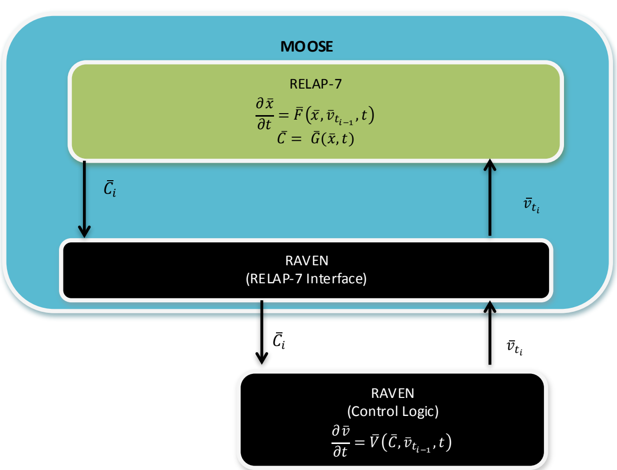

Figure 1. Control System Software Layout.

middleware software MOOSE, that, in addition to provide the algorithms for the solution of the differential equation, will also provide all the manipulation tools for the C++ classes containing the solution vector. More in detail the plant is represented by a set of components and each component type corresponds to a C++ class. At each time step RELAP-7/MOOSE will update 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 know no situations where the complexity of 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} required a numerical solution therefore for the moment RAVEN remains numerical integration free. Note that 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 3. 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 3 or ?? will be referred.

4. CONCLUSIONS

[]

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).

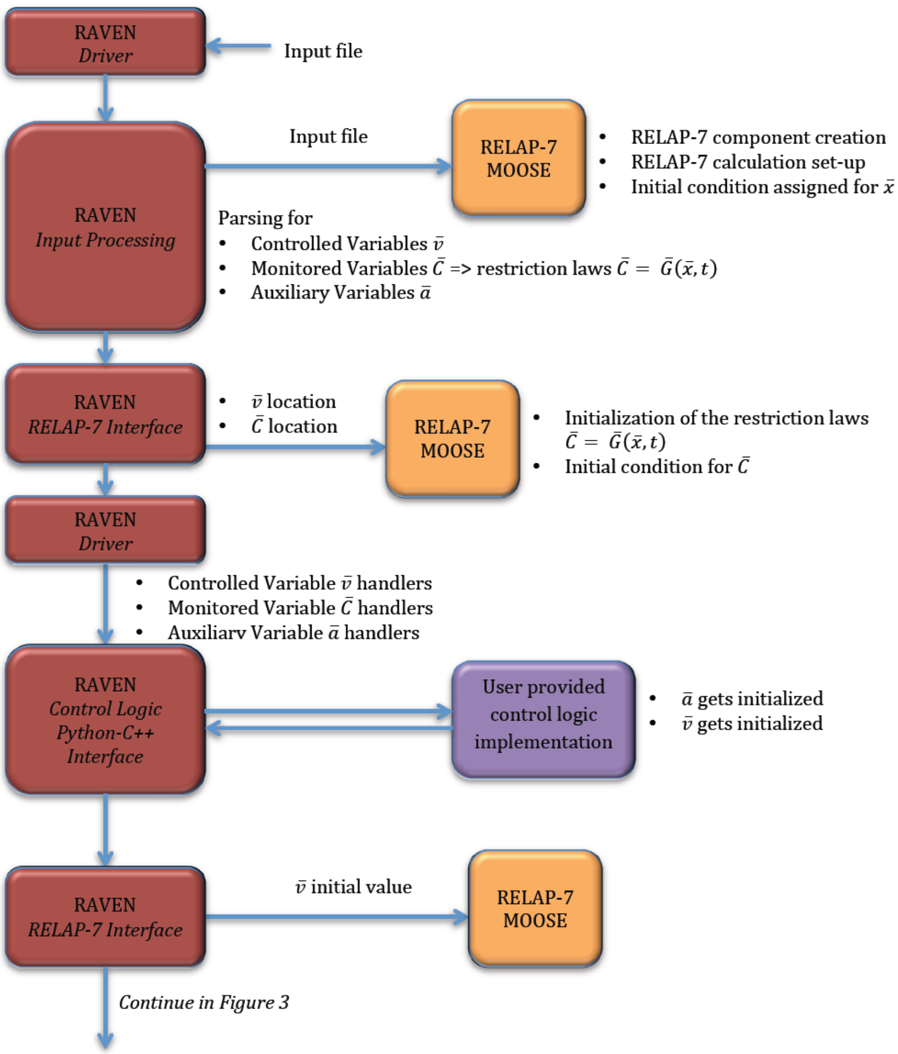

Figure 2. Control System Software Layout.

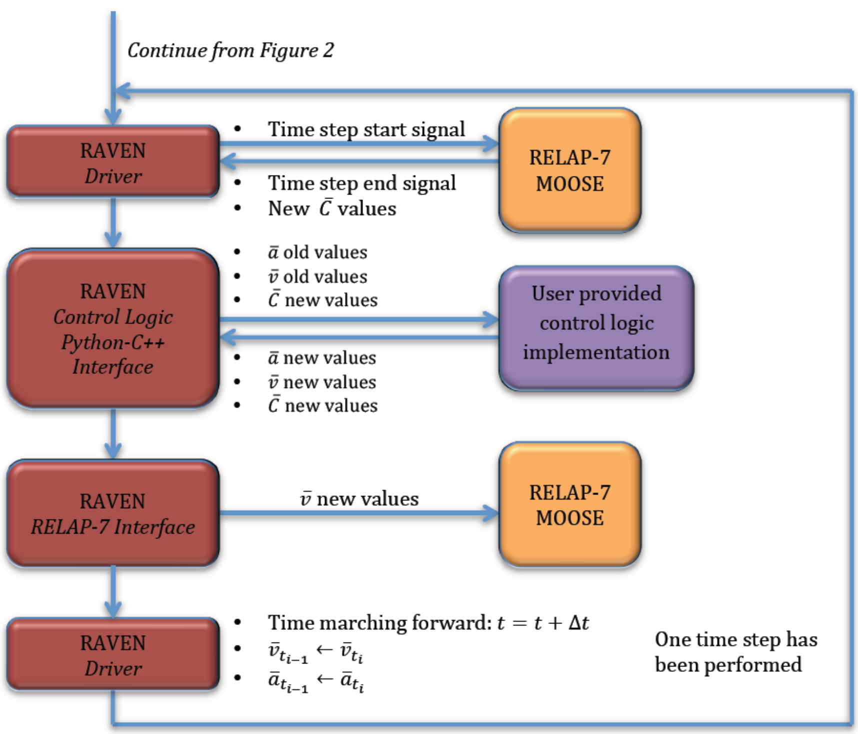

Figure 3. Control System Software Layout.